UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# EXTRAÇÃO PERSONALIZADA E INCREMENTAL DE DADOS EM AMBIENTES DE BI TEMPO REAL

Daniel Barbosa Martins

**Orientador**
Fernanda Araujo Baião
**Co-orientador**
Maria Cláudia Reis Cavalcanti

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2009

EXTRAÇÃO PERSONALIZADA E INCREMENTAL DE DADOS EM AMBIENTES DE BI TEMPO REAL

Daniel Barbosa Martins

DISSERTAÇÃO APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE MESTRE PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

_____
Fernanda Araujo Baião, D.Sc. – UNIRIO

_____
Maria Cláudia Reis Cavalcanti, D.Sc. – IME

_____
Maria Luiza Machado Campos, D.Sc. – UFRJ

_____
Sean Wolfgand Matsui Siqueira, D.Sc. – UNIRIO

_____
Jonice de Oliveira Sampaio, D.Sc – UFRJ

RIO DE JANEIRO, RJ – BRASIL
SETEMBRO DE 2009

Martins, Daniel Barbosa.
M386 Extração personalizada e incremental de dados em ambientes de BI tempo real / Daniel Barbosa Martins, 2009.
ix, 112f.

Orientador: Fernanda Araujo Baião.
Co-orientador: Maria Cláudia Reis Cavalcanti.
Dissertação (Mestrado em Informática) – Universidade Federal do Estado do Rio de Janeiro, Rio de Janeiro, 2009.

1. Business intelligence. 2. Qualidade de dados. 3. Processo de extração incremental de dados. 4. Sistemas distribuídos. I. Baião, Fernanda Araujo. II. Cavalcanti, Maria Cláudia Reis III. Universidade Federal do Estado do Rio de Janeiro (2003-). Centro de Ciências Exatas e Tecnologia. Curso de Mestrado em Informática. III. Título.

CDD – 005.7

# DEDICATÓRIA

Dedico este trabalho

À minha mãe,
À minha irmã,
À minha esposa,
Ao meu filho

# AGRADECIMENTOS

Às Professoras Fernanda Araújo Baião e Maria Cláudia Reis Cavalcanti, pela ótima orientação, sempre me ajudando quando necessário para conclusão deste trabalho.

À minha esposa Conceição, pelo seu amor e compreensão que foram fundamentais nestes dois anos e meio de mestrado.

Aos Professores Astério Kiyoshi Tanaka e Sean Wolfgand Matsui Siqueira pelas sugestões e críticas durante os seminários de acompanhamento discente.

A UNIRIO, mais especificamente ao Programa de Pós-Graduação em Informática e seus professores, pelos anos de ensino.

Ao amigo e gerente Nelson Borges Alves Neto pelo apoio concedido.

A todos os colegas e amigos que me acompanharam nessa caminhada até o mestrado.

Agradeço também a todas as pessoas que contribuíram direta e indiretamente para a realização deste trabalho.

Aos professores convidados a integrar a banca pela disponibilidade e atenção.

Meus amigos do mestrado um abraço especial a todos vocês que sempre estiveram juntos nessa grande etapa de nossa vida.

Martins, Daniel Barbosa. **Extração Personalizada e Incremental de Dados em Ambientes de BI Tempo Real**. UNIRIO, 2009. 112 páginas. Dissertação de Mestrado. Departamento de Informática Aplicada, UNIRIO.

# RESUMO

Tendências atuais em ambientes de Inteligência de Negócios (*Business Intelligence – BI*) incluem a demanda pela disponibilização de informações de imediato no *Data Warehouse* (DW), reduzindo o tempo entre a ocorrência de um evento no ambiente transacional e o momento quando uma decisão é tomada no ambiente informacional, com o propósito de melhorar o desempenho dos processos de tomada de decisão. Nestes ambientes, a existência de usuários diferentes, com visões particulares sobre a qualidade dos dados, acarreta em necessidades distintas quanto à atualização dos dados no DW. Tipicamente para atender a todos os usuários, o processo de carga do DW (processo ETL) é realizado para todo o grande volume de dados (completo), na menor freqüência requerida, impactando fortemente no tempo para a atualização do DW. Este trabalho busca uma maior adequação do processo ETL às tendências atuais de BI. Para reduzir o tempo de atualização do DW, contribuindo com o aumento da qualidade de dados, e ao mesmo tempo viabilizar a co-existência de usuários com diferentes requisitos de qualidade de dados, este trabalho propõe a arquitetura Brahma, que especifica e executa dinamicamente instâncias do processo de extração incremental de dados, baseando-se nas necessidades de grupos de usuários com visões semelhantes sobre a qualidade dos dados. A arquitetura proposta foi implementada em um protótipo e avaliada num cenário baseando-se no esquema do Benchmark TPC-H. Os resultados iniciais desta avaliação mostraram que as instâncias ETL executadas segundo a arquitetura Brahma apresentam ganho de desempenho significativo, diretamente proporcional à quantidade de dados irrelevantes eliminados em cada instância ETL executada, além de manter a qualidade de dados do DW em um nível mais elevado e constante.

**Palavras-chave**: Business Intelligence, Qualidade de Dados, Sistemas Distribuídos, Processo de Extração Incremental de Dados.

# ABSTRACT

Trends in Business Intelligence environments require information immediately available in the Data Warehouse, reducing the time interval from the moment of an event in the transactional environment to the moment a decision is taken in the informational environment. This is called BI 2.0, or real-time BI, and its main goal is to improve decision-making processes. However, one of the difficulties to implement real-time BI is that ETL process execution is highly time-consuming, because it is carried out considering the whole (full) set of source data, in a pre-determined low frequency. In these environments, the existence of different users, with particular views on the data quality, leads to different needs regarding DW update frequency. This work proposes the Brahma architecture that dynamically specifies and executes ETL process instances to provide incremental extraction of data based on the needs of groups of users with similar views on the data quality. The proposed architecture was implemented in a prototype and evaluated with the TPC-H Benchmark. The experimental results showed that Brahma provided significant performance gain (proportional to the amount of irrelevant data removed in each ETL instance), while maintained the DW data quality in a higher and constant level.

**Keywords**: Business Intelligence, Data Quality, Distributed Systems, Process to Data Incremental Extraction.

# SUMÁRIO

1. Introdução ..... 1
1.1. Motivação ..... 1
1.2. Objetivo ..... 3
1.3. Contribuições ..... 4
1.4. Estrutura ..... 4
2. Business Intelligence ..... 6
2.1. Visão Geral ..... 6
2.2. Data Warehouse ..... 6
2.2.1. Modelo Dimensional ..... 8
2.2.2. Granularidade ..... 10
2.3. Extração, Transformação e Carga ..... 11
2.3.1. Extração Completa e Extração Incremental ..... 12
2.3.2. Transformação ..... 13
2.3.3. Carga Completa e Carga Incremental ..... 13
2.4. BI Tempo Real ..... 14
3. Qualidade de Dados ..... 16
3.1. Avaliação da Qualidade de Dados em Banco de Dados ..... 17
3.2. Qualidade de Dados em Data Warehouse ..... 18
3.2.1. Fatores de Qualidade ..... 19
3.3. Modelo de Qualidade de Dados Proposto por SIMMHAN ..... 22
3.3.1. Elementos do Modelo de Qualidade ..... 22
3.3.2. Medição dos Fatores de Qualidade ..... 27
3.3.3. Técnicas de Medição dos Fatores de Qualidade ..... 28
3.4. Considerações ..... 30
4. Arquitetura BRAHMA ..... 32
4.1. visão Geral da Arquitetura ..... 32
4.2. Detalhamento da Solução ..... 34
4.2.1. Modelo de Qualidade e Fatores de Qualidade ..... 34

4.2.2. Funcionamento da Arquitetura ..... 35
4.3. Considerações ..... 49
5. Implementação e Cenário de Avaliação da Arquitetura BRAHMA ..... 50
5.1. Implementação da Arquitetura BRAHMA ..... 50
5.1.1. Módulo de Perfis ..... 50
5.1.2. Módulo de Expressões de Extração ..... 52
5.2. Cenário definido para Avaliação da Proposta ..... 53
5.2.1. TPC-H ..... 53
5.2.2. Geração dos Dados do Esquema Fonte ..... 57
5.2.3. Definição de Perfis para Avaliação da Arquitetura ..... 57
5.2.4. Especificação do Processo ETL ..... 60
5.3. Implementação do ambiente de avaliação ..... 62
6. Avaliação da Arquitetura BRAHMA ..... 64
6.1. Procedimentos ..... 64
6.2. Análise dos Resultados ..... 65
6.2.1. Avaliação da Métrica Tempo de Execução ..... 65
6.2.2. Análise da Latência de Dados ..... 68
7. Conclusão ..... 72
7.1. Contribuições ..... 73
7.2. Limitações da Proposta ..... 74
7.3. Trabalhos Futuros ..... 74
Referências ..... 76
ANEXO I – Script de Criação de Tabelas do Módulo de Perfis ..... 81
ANEXO II – Script de Criação de Tabelas do TPC-H ..... 84
ANEXO III –Script de Criação do Esquema Dimensional ..... 86
ANEXO IV – Carga Inicial das Dimensões ..... 88
ANEXO V – Visões para Carga na Tabela Fato ..... 89
ANEXO VI – Carga na Tabela Fato ..... 94
ANEXO VII – Arquivo de LOG (registros / tempo de execução) ..... 95
ANEXO VIII – Arquivo de LOG (pontuação perfis) ..... 96
ANEXO IX – Scripts para Pontuação dos Perfis ..... 99
ANEXO X – Criação de Arquivos de LOG ..... 103

# *1. Introdução*

O presente trabalho de dissertação de mestrado enfoca a extração incremental de dados em um ambiente de Inteligência de Negócio (*Business Intelligence – BI*) para suporte à qualidade de dados. Neste capítulo, temos uma introdução a motivação do trabalho, seus objetivos e contribuições. Além de descrever a estrutura do mesmo.

## *1.1. Motivação*

Nas organizações atuais, um problema clássico é o grande volume de dados existente, que dificulta a gerência, manipulação e extração de informações, em especial no processo de tomada de decisões gerenciais, na medida em que a busca e recuperação dos dados torna-se mais complexa. Os dados necessários para tomadas de decisões estratégicas estão, freqüentemente, escondidos em milhares de tabelas e arquivos não documentados, ligados por relacionamentos e correlações transacionais, replicados e inconsistentes, estruturados de forma inadequada para os tomadores de decisão. Segundo BARBIERI (2001), as técnicas de *Business Intelligence* (BI) têm por objetivo, neste contexto, a definição de regras e métodos para a formatação adequada destes dados, visando transformá-los em depósitos estruturados de informações (denominados *Data Warehouses – DW*), independentemente de sua origem.

Nos ambientes de BI, a qualidade de dados é um aspecto essencial que precisa ser tratado, na medida em que mais e mais decisões estratégicas dependem do DW (OZSU e VALDURIEZ, 2001). Para AMARAL (2003), é fundamental adotar uma estratégia para garantir a qualidade de dados neste ambiente. Uma vez que a qualidade dos dados afeta o resultado das análises, é interessante que se possa medir o grau de qualidade destes dados, e que o valor obtido seja considerado durante o processo de tomada de decisão. Critérios típicos para avaliação da qualidade de dados em ambientes de DW são a interpretabilidade, a utilidade, a acessibilidade e a verossimilhança (JARKE e VASSILIOU, 1997, JARKE *et al.*, 2001), especificados na proposta do *Data Warehouse Quality* (DWQ). Para calcular o grau de qualidade dos dados de um DW é preciso definir como aplicar estes critérios. Em seu trabalho, SIMMHAN (2007) define um modelo para qualidade de dados independente de domínio (não focado em DW), permitindo a configuração de critérios de qualidade específicos aos usuários do ambiente.

Tendências em BI têm considerado disponibilizar informações de imediato, o que faz parte da definição do chamado BI 2.0 (STODDER, 2007), também denominado *Real-Time BI* (DAYAL *et al.*, 2009). O propósito dos ambientes de BI 2.0 é melhorar o desempenho dos processos de tomada de decisão, reduzindo o tempo entre a ocorrência de um evento no ambiente transacional e o momento quando uma decisão é tomada no ambiente informacional. De fato, as implementações de ferramentas segundo esta proposta já existem e apontam um grande diferencial para o mundo corporativo (IBM, 2008), (LOUIS, 2008), (ERICSON, 2009). Neste contexto, onde é esperado que os dados no DW sejam os mais recentes possíveis, o fator de qualidade que é evidenciado com relação a este aspecto é a temporalidade (*timeliness*), que engloba vários subfatores, dentre os quais a "atualidade do DW". Definimos temporalidade em um ambiente de BI, portanto, como o intervalo de tempo entre a ocorrência de um evento no ambiente operacional e a disponibilização dos dados gerados por este evento no ambiente informacional (DW).

São diferentes os perfis de usuários em um ambiente de BI, pois cada usuário pode ter uma visão particular sobre qualidade de dados, ou sobre quais critérios são mais prioritários para determinar a alta qualidade de um item de dado. Tipicamente, nestes ambientes, existe um perfil de usuário que necessita de informações mais detalhadas (de nível tático) com menor temporalidade (isto é, maior freqüência de atualização), e um perfil de usuário que precisa de informações menos detalhadas (de nível estratégico) com uma temporalidade maior (menor freqüência). Outro fator que caracteriza um perfil de usuário é o seu objetivo ao utilizar o DW, que de forma geral pode ser para fins de tomada de decisões ou para fins de auditoria. Usuários de nível tático ou que exercem atividades de auditoria normalmente requerem um nível mais fino de granularidade para os dados do DW, e isto leva a um inevitável e significativo aumento da quantidade de dados necessários para serem levados ao DW através de processos de ETL (Extração, Transformação e Carga – *Load*).

A existência de diferentes perfis de usuário requer que o ambiente de BI seja capaz de atender a diferentes requisitos de qualidade de dados, considerando todos os seus usuários. No entanto, tipicamente, estes ambientes não levam em conta estes diferentes perfis e oferecem um único processo ETL para a atualização dos dados do DW. Nos cenários de integração de dados também há uma preocupação cada vez maior com aspectos de personalização, isto é, a adequação a diferentes tipos de usuários. De acordo com ZIEGLER *et al.* (2008), a integração de dados, baseada em um esquema global único, pode interferir seriamente nas necessidades individuais dos

usuários, impactando, de forma negativa, na qualidade de dados devido à baixa adequabilidade de uso dos dados integrados. Uma vez que um ambiente de BI pode ser considerado uma abordagem para integração de dados, através de um repositório único de dados (ZIEGLER e DITTRICH, 2007), é importante que se proponham mecanismos para tratar o problema de personalização em tais ambientes, no sentido de viabilizar a co-existência de diferentes perfis de usuário, minimizando a perda da qualidade. Em particular, o presente trabalho busca a redução da temporalidade em ambientes de BI 2.0 através do aumento do desempenho de cada execução do processo de ETL, levando-se em conta a existência de diferentes perfis de usuário.

## *1.2. Objetivo*

O objetivo deste trabalho é defender a hipótese de que a redução do volume de dados a cada execução do processo de ETL melhora o seu desempenho, possibilitando atender os diferentes requisitos de temporalidade dos usuários, e conseqüentemente aumenta a qualidade dos dados do DW em ambientes de BI 2.0.

Técnicas de distribuição e paralelismo têm sido empregadas com sucesso para tratar problemas de desempenho em ambientes que manipulam grandes volumes de dados. O trabalho de VASSILIADIS *et al.* (2007), por exemplo, aponta para o uso de particionamento de dados e paralelismo intra-processo como estratégia para melhoria de desempenho em cada execução do processo de ETL. O paralelismo intra-processo significa o uso de paralelismo durante a execução de uma única instância de processo ETL.

Este trabalho propõe transformar uma instância de processo de ETL em várias instâncias distintas, buscando melhor desempenho na execução de cada instância. Neste sentido, a proposta desta dissertação é definir grupos de usuários com necessidades de qualidade de dados semelhantes entre si, e configurar instâncias distintas de processos de ETL que satisfaçam os requisitos de cada grupo. Isto é feito através da definição de perfis de usuário para cada grupo com base no modelo de qualidade de dados de SIMMHAN (2007), instanciado com os critérios definidos pelo DWQ, e da configuração de um processo ETL para cada grupo de forma a reduzir o volume de dados desnecessários a serem extraídos para aquele perfil, através da extração incremental de dados. O princípio da redução de dados irrelevantes é bastante conhecido na literatura sobre Sistemas de Bancos de Dados Distribuídos, como forma de aumentar o desempenho do processamento de consultas executadas

sobre fragmentos da base de dados (OZSU e VALDURIEZ, 2001). Esta eliminação de parte dos dados para cada grupo de usuários contribui para o aumento do desempenho da instância de ETL.

## *1.3. Contribuições*

As principais contribuições desta dissertação são a especificação, implementação e avaliação de uma arquitetura, denominada Brahma, para definição e execução dinâmica de processos de extração personalizada e incremental de dados em ambientes de BI tempo real. Além disso, contribuições específicas incluem a instanciação do modelo de qualidade de dados de SIMMHAN (2007) com os critérios definidos pelo DWQ, a implementação da arquitetura e a avaliação desta em um cenário do Benchmark TPC-H.

## *1.4. Estrutura*

Esta dissertação, além deste capítulo introdutório, possui mais seis capítulos conforme se segue.

No capítulo 2 são descritos os principais conceitos envolvidos nos ambientes de *Business Intelligence*, a definição, sua importância como diferencial competitivo para as organizações, suas características e tendências.

O capítulo 3 discorre brevemente sobre a qualidade de dados e sua importância para usuários do banco de dados, com maior enfoque no *Data Warehouse*. Este capítulo apresenta algumas propostas para os fatores de qualidade de dados, dentre elas a proposta pelo DWQ, que determina fatores de qualidade especializados para *data warehouses*, e o modelo de qualidade dos dados proposto por SIMMHAN (2007), que permite avaliar a qualidade dos dados armazenados no DW.

A solução proposta nesta dissertação é apresentada no capítulo 4, a arquitetura BRAHMA, utilizando uma instanciação necessária dos fatores de qualidade do DWQ ao modelo de qualidade para a avaliação da qualidade dos dados como proposto por SIMMHAN (2007), e enfatizando a construção das expressões de extração que viabilizam a configuração de instâncias de processos de extração incremental de dados.

Já o capítulo 5 relata como é implementada e validada a proposta, com criação de ambiente de avaliação sobre uma base de dados gerada pelo benchmark TPC-H. O capítulo apresenta ainda a especificação das aplicações desenvolvidas e as ferramentas de apoio utilizadas para prover suporte à arquitetura.

Os resultados obtidos através de testes da arquitetura são analisados no capítulo 6.

Esta dissertação é finalizada no capítulo 7 com as conclusões, que apresentam as considerações finais, ressaltando as contribuições e limitações da proposta e perspectivas futuras.

# 2. Business Intelligence

Neste capítulo, discorre-se brevemente sobre os principais conceitos envolvidos nos ambientes de *Business Intelligence*, a definição, sua importância para as organizações, suas características e tendências.

*Business Intelligence* (BI) possibilita entender toda a complexidade de geração, troca e uso de informações importantes para os negócios da organização, permitindo a previsão das conseqüências das decisões nos negócios.

## 2.1. Visão Geral

*Business Intelligence* (Inteligência em Negócios), um termo genérico para ativos de informação de todos os tipos utilizados por organizações ao efetuar uma decisão de negócio (KIMBALL e MERZ, 2000).

O conceito de BI deve ser entendido como o processo de desenvolvimento de estruturas especiais de armazenamento de informações, como *Data Warehouse* (DW), Data Marts (DM) e ODS (Operational Data Store), com o objetivo de se construir uma base de recursos informacionais, capaz de sustentar a camada de inteligência da empresa. O conceito de BI contempla também o conjunto de aplicações e ferramentas de apoio ao processo de Extração, Transformação e Carga (*Extract, Transform, Load - ETL*), fundamental para a transformação do recurso de dados transacional (existente no ambiente operacional da organização) em informacional. (BARBIERI, 2001).

## 2.2. Data Warehouse

O *Data Warehouse* (DW) é um armazém de dados. Pode ser definido como um banco de dados, voltado a sistemas de apoio à decisão, armazenados em estruturas lógicas dimensionais, segundo a proposta de KIMBALL (2002). Guarda grande quantidade de dados para se ter uma visão mais ampla das informações relacionadas à organização.

INMON (1997) define um *Data Warehouse* como “uma coleção de dados orientados por assunto, integrados, não voláteis, variável em relação ao tempo, de apoio às decisões gerenciais”.

Em termos gerais, um *Data Warehouse* é um grande banco de dados que armazena informações integradas a partir de bancos de dados operacionais e fontes externas de uma organização (REZENDE, 1999).

O conceito de DW surgiu da necessidade de integrar dados corporativos espalhados em diferentes máquinas e sistemas operacionais, para torná-los acessíveis a todos os usuários de níveis decisórios. Outro fator que contribuiu para o estabelecimento desse conceito foi a evolução da tecnologia da informação, particularmente os sistemas de apoio à decisão (DSS). O DW surge como uma solução às necessidades de informações para atividades de planejamento empresarial e controle de usuários de nível gerencial (tipicamente usuários de nível estratégico ou tático dentro de uma organização), como tomada de decisões e auditoria. No suporte à decisão, dados históricos, sumarizados e consolidados são mais importantes do que registros detalhados e individuais; enquanto que em atividades de auditoria a granularidade requerida para os itens de dados é mais fina.

Considere o seguinte cenário que ilustra a existência de necessidades bem distintas no uso de um DW. Suponha dois usuários típicos de um ambiente de BI, que acessam um mesmo DW para apoio às suas atividades em uma empresa de varejo. O usuário GRV é um gerente regional de vendas e o usuário GGV é um gerente geral de vendas, de âmbito nacional. O usuário GRV tem a preocupação de acessar as informações mais atualizadas sobre as vendas de produtos de um determinado local com maior detalhamento, e portanto atribui maior prioridade aos critérios de temporalidade e completeza para a avaliação da qualidade dos dados do DW. Já o usuário GGV acessa as informações consolidadas de vendas por semestre em cada região, e por isso atribui maior prioridade ao critério credibilidade. A existência de diferentes usuários com necessidades distintas de dados requer que o ambiente contenha os dados na sua forma mais completa, para atender aos requisitos de qualidade de dados de todos os usuários.

As características fundamentais de um *data warehouse* são:

- **Orientado por assunto:** Os dados são organizados por assuntos de interesse da organização, contendo apenas informações relevantes ao processo de tomada de decisões. Um DW difere de um banco de dados operacional no sentido de que este último, em sua maioria, é orientado por produto e ajustado para lidar com transações diárias que atualizem o banco de dados. A orientação por assunto

provida pelo DW proporciona uma visão mais abrangente da organização, permitindo determinar além de seu desempenho, as razões que o justificam;

- **Integrado:** O processo de introdução dos dados no *data warehouse* deve ser feito de forma consistente e independente da aplicação de origem, eliminando as inconsistências que possam existir oriundas dessas aplicações. Conflitos de nomenclaturas e discrepâncias entre unidades de medida devem ser resolvidos. A integração sofre forte influência da orientação por assuntos. O fato dos dados terem um alto nível de integração é considerado por INMON (1997) como a mais importante dentre todas as características presentes em um DW;

- **Não volátil:** Os dados de um *data warehouse* são normalmente carregados e acessados em grandes quantidades. A atualização dos dados geralmente não ocorre no ambiente de *data warehouse*. Dados sumarizados obsoletos são descartados e as alterações são registradas como dados novos. Em oposição, no ambiente operacional, os dados operacionais são acessados e tratados, um registro por vez, podendo sofrer atualizações. Tais fatos permitem que o *data warehouse* possa ser ajustado, quase que exclusivamente, para acesso a dados;

- **Variante no tempo:** Um *data warehouse* mantém dados históricos. O tempo é uma dimensão importante para um *data warehouse*. A estrutura de chave do *data warehouse* sempre contém algum elemento de tempo, enquanto que a estrutura de chave dos dados operacionais pode conter, ou não, elementos de tempo. O horizonte de tempo válido para o *data warehouse* é significativamente maior do que o dos sistemas operacionais. Um horizonte de tempo de dias ou meses é comum para os sistemas operacionais, enquanto que em um *data warehouse* esse horizonte passa a ser de anos ou décadas.

### 2.2.1. Modelo Dimensional

No *Data Warehouse*, os dados devem permitir consultas considerando diferentes aspectos de negócios. A representação que explicita as diferentes dimensões sobre as quais se deseja analisar os fatos relevantes ao negócio, de forma independente uma da outra, permitindo flexibilidade na definição de análises cruzadas, é uma forma de conseguir atingir esse objetivo.

A técnica utilizada na modelagem de dados que considera a característica acima é conhecida como modelagem dimensional, sendo esta utilizada na estruturação dos dados para consulta. As bases de dados são mais fáceis de navegar, de consultar e possuem menos tabelas e chaves.

A modelagem dimensional consiste na utilização de técnicas que permitam o armazenamento dos dados, em um formato distinto ao totalmente normalizado, com o objetivo de aumentar a capacidade de compreensão da informação pelo usuário, melhorar o desempenho nas consultas e flexibilizar as adaptações decorrentes de mudanças. Ainda, a modelagem dimensional permite aos usuários uma percepção dos dados através de perspectivas distintas e mais próximas de seu entendimento, dentre elas o tempo e o espaço (BARBIERI, 2001; KIMBALL e ROSS, 2002). Nessa abordagem, os dados são mapeados em uma estrutura com mais de uma dimensão para modelar os dados. Os componentes dos negócios (pontos de vista) da organização são representados nas dimensões dessa estrutura. Na interseção das dimensões têm-se as medidas ou fatos, geralmente valores numéricos.

De acordo com ELSMARI e NAVATHE (2005), dois esquemas multidimensionais comuns em ambientes de *data warehouse* são: o estrela (*Star Schema*) e o floco de neve (Snowflake Schema).

O esquema estrela, descrito por KIMBALL e ROSS (2002), consiste em uma tabela fato com uma única tabela para cada dimensão, enquanto que o esquema floco de neve é uma variação do esquema estrela, onde as tabelas dimensionais são organizadas em uma hierarquia por meio de normalização dessas tabelas. No esquema estrela, as hierarquias são representadas por atributos nas dimensões. As figuras 2.1 e 2.2 ilustram a estrutura desses esquemas.

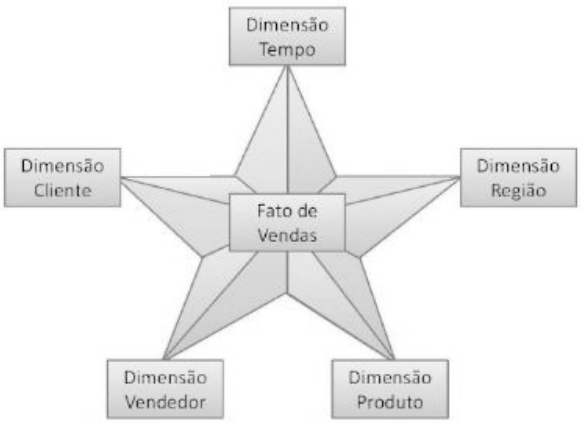


**Figura 2.1** - Esquema estrela ou *star schema* (MACHADO, 2008)

Segundo KIMBALL e ROSS (2002), as principais vantagens do esquema estrela são: possui uma arquitetura simples, facilitando o entendimento e o uso do DW, assim como, sua extensibilidade; tem estrutura simétrica, possibilitando melhor desempenho pela redução no número de junções necessárias nas consultas;


**Figura 2.2** - Esquema floco de neve ou *snowflake schema* (MACHADO, 2008)

O esquema floco de neve possui como principal vantagem a redução de redundâncias, porém a economia no armazenamento não justifica as desvantagens associadas à complexidade da estrutura necessária.

A escolha do modelo floco de neve, em detrimento ao modelo estrela, deve ser realizada com extremo conservadorismo, pelo fato do primeiro diminuir a facilidade de uso e o desempenho, quando comparado ao segundo (KIMBALL e ROSS, 2002).

### 2.2.2. Granularidade

INMON (1997) define granularidade como o nível de detalhe ou sumarização contido nas unidades de dados de um data warehouse. Quanto maior o detalhamento, menor a granularidade. Quanto menor o detalhamento, maior a granularidade.

A granularidade é o mais importante aspecto do projeto de um DW, devendo ser definida em seguida à seleção do processo de negócio a ser modelado (INMON, 1997; KIMBALL e ROSS, 2002). Tal importância reside no fato de que a granularidade determina o volume da base de dados que será necessário para o armazenamento dos dados no DW e a variedade de consultas que poderão ser atendidas.

O volume de dados armazenado no DW será diretamente proporcional ao nível de detalhe (granularidade) definido para estes dados, devendo ser considerado de forma a não comprometer o tempo na obtenção da informação. Em ambientes de BI 2.0 este aspecto é particularmente crítico, como veremos mais adiante. Estratégias como o estabelecimento de níveis de agregação para os dados contribuem para minimizar os impactos em relação ao critério tempo na obtenção da informação.

Na medida em que o nível de granularidade aumenta, a variedade de consultas que podem ser atendidas diminui, sendo que em uma granularidade mínima as consultas mais 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 dos usuários, tendo como limitação os recursos disponíveis (HOKAMA *et al*, 2004).

## *2.3. Extração, Transformação e Carga*

De modo geral, o processo de ETL, extrai dados de diversas origens distintas e heterogêneas, seguido da transformação de dados e limpeza de dados, e finalmente são carregados no DW onde ficam disponíveis às aplicações de BI (JÖRG e DESSLOCH, 2009). Contudo, essa não é uma tarefa simples. Muitos desses dados não estão adequados para serem utilizados no DW. Geralmente eles não estão padronizados, tornando difícil relacioná-los, sendo necessário que ocorram transformações e integrações desses dados.

Em um processo de tomada de decisões, ferramentas de consulta analítica, também chamadas ferramentas OLAP, são utilizadas na obtenção de informações relevantes para a organização. Estas informações são obtidas através de consultas sobre os dados armazenados em um *Data Warehouse*.

Para que os dados possam estar disponíveis, são utilizadas rotinas de ETL (*Extract Transform Load*) que trazem as informações de ambientes externos, fazendo tratamentos, agregação ou sumarizações, permitindo que o DW tenha dados íntegros e históricos. A figura 2.3 contextualiza o processo de ETL em um ambiente de DW.

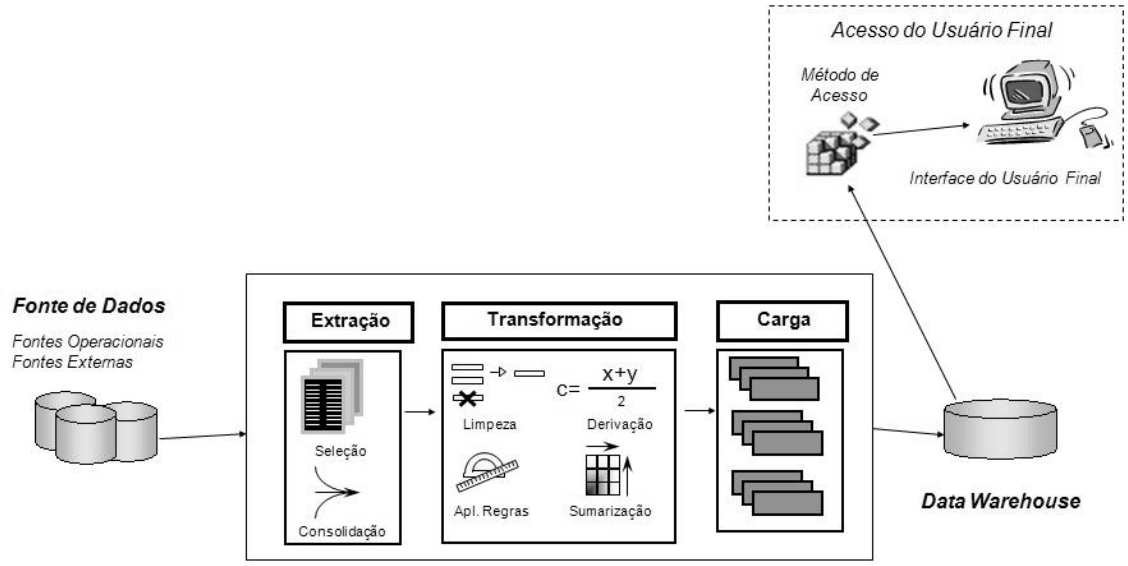


**Figura 2.3** - Contextualização do processo de ETL no ambiente de DW (adaptado de COSTA e ANCIÃES, 2001)

O processo de ETL fornece aos usuários do DW dados estruturados, integrados, consistentes e de qualidade, de maneira a prover suporte aos processos de tomada de decisões para uma organização. Quando estes dados não são corretamente manipulados no processo de ETL, a qualidade das análises é prejudicada. Como conseqüência da baixa qualidade, decisões equivocadas provavelmente serão tomadas, podendo afetar diretamente os negócios da organização.

### 2.3.1. Extração Completa e Extração Incremental

A primeira fase de um processo ETL é a extração de dados dos sistemas de origem, que é responsável pela identificação, seleção e obtenção dos dados de interesse para o DW. Estes dados podem ser provenientes de diversas fontes, que se valem de diferentes SGBDs (Oracle, SQL Server, MySql, DB2, ou PostgreSQL) ou até mesmo correspondem a um simples texto, assim como podem encontrar-se nos mais variados formatos. É nesta fase que os dados são convertidos para um mesmo formato para a entrada no processamento da transformação. Para que uma extração ocorra, é necessário que um programa percorra um arquivo ou banco de dados, utilizando alguns critérios de seleção e, encontrando dados que atendam aos critérios, transportá-los para outro arquivo ou banco de dados. Um fator de influência sobre a complexidade na etapa de extração é a variedade de plataformas e tecnologias

utilizadas pelos sistemas de origem. Neste caso, poderá ser necessária a implementação de mecanismos de extração diferenciados.

Os dados podem ser extraídos por completo do sistema de origem, ou realizar a extração incremental. Na extração incremental, os dados destinados à extração serão obtidos a partir das alterações ocorridas no sistema de origem desde a última extração, reduzindo a carga sobre os processos de armazenamento de dados em DW com grandes volumes de dados.

### 2.3.2. Transformação

A fase de transformação ocorre logo após a fase de extração dos dados. O objetivo desta fase é manipular os dados provenientes dos sistemas de origem, obtidos através do processo de extração de dados, de forma a torná-los úteis para os usuários do DW e de relevância para o negócio. Estes dados manipulados serão posteriormente armazenados no DW através do processo de carga. Aplicam-se uma série de regras ou funções aos dados extraídos para derivar os dados a serem carregados. Neste momento é colocado um padrão no resultado. Por exemplo, um simples cadastro de clientes, onde o cadastro dos clientes foi preenchido no campo do sexo de diversas maneiras (M ou F), (Masculino ou Feminino). Sendo assim, um padrão deverá ser assumido. Se for assumido o padrão (M ou F), os campos preenchidos como (Masculino ou Feminino) deverão ser transformados em M ou F. Esse trabalho de transformação pode ser utilizado em diversos casos como o telefone, CEP, CPF, Data e outros. Durante esta fase, diversos tipos de transformações podem ocorrer: Integração, Limpeza e Validação, Derivação, Sumarização, Atualizações do Histórico e Ordenação.

### 2.3.3. Carga Completa e Carga Incremental

A fase de carga é a última do processo de ETL e consiste na carga dos dados no Data Warehouse. Nesta fase, os dados transformados sofrem uma última preparação, para então serem armazenados no DW e disponibilizados para as consultas dos usuários.

Durante a carga inicial, os dados extraídos são colocados no DW. A cada alteração da fonte de dados ao longo do tempo, o DW fica obsoleto e, portanto, precisa ser

atualizado. Atualização do DW é realizada com uma periodicidade, que em ambientes tradicionais é tipicamente diária, semanal ou mensal,.

Uma abordagem para atualização do DW é a recarga completa (JÖRG e DESSLOCH, 2009). A idéia é simplesmente re-executar o trabalho de carga inicial, recolher os dados obtidos, e compará-lo ao conteúdo do *Data Warehouse*. Desta forma, as alterações necessárias para atualização do DW podem ser recuperadas. Recarga completa é obviamente ineficiente, considerando que na maioria das vezes apenas uma pequena fração dos dados de origem são alterados durante o processo de ETL.

Outra abordagem propaga para o DW apenas as alterações, e é conhecida como carga incremental (JÖRG e DESSLOCH, 2009). Em geral, a carga incremental pode ser considerada mais eficiente do que recarga completa.

## *2.4. BI Tempo Real*

Tendências atuais em BI têm considerado atualizar informações no DW de imediato, o que faz parte da definição do chamado BI 2.0 ou em tempo real (STODDER, 2007), (DAYAL *et al.*, 2009). O propósito dos ambientes de BI 2.0 é melhorar o desempenho dos processos de tomada de decisão, reduzindo o tempo entre a ocorrência de um evento no ambiente operacional e o momento quando uma decisão é tomada no ambiente informacional (DW).

Neste contexto, onde é esperado que os dados no DW sejam os mais recentes possíveis, o fator de qualidade dos dados que é evidenciado é a temporalidade (*timeliness*). Segundo a hierarquia de fatores de qualidade de dados proposta pelo DWQ (JARKE e VASSILIOU, 1997), a temporalidade se decompõe em três subfatores: atualidade das fontes, atualidade do DW e volatilidade. No escopo deste trabalho a temporalidade foi reduzida ao subfator “atualidade do DW”, que em um ambiente de BI é definido como o intervalo de tempo entre a ocorrência de um evento no ambiente operacional e a disponibilização dos dados gerados por este evento no ambiente informacional, para atividades analíticas típicas de um ambiente de DW, como, por exemplo, tomadas de decisões e auditoria. Isto é, uma temporalidade maior significa um intervalo de tempo maior para a disponibilização do item de dado no DW, ou seja, uma freqüência menor de atualização deste. No entanto, deve-se levar em consideração que o fator de temporalidade pode influenciar outros fatores de qualidade, como a utilidade dos dados. Isto é, quanto maior a temporalidade em um

ambiente de BI para um determinado item de dado, menor a utilidade dele para usuários que tenham como foco o acesso aos dados mais atualizados.

A abordagem de BI em tempo real utiliza a modelagem multidimensional de DW tradicional. No entanto, para que exista disponibilidade imediata dos dados, um novo conjunto de aplicações necessita estar disponível, de forma a manter os dados atualizados no DW com muito mais rapidez, quando comparado às típicas atualizações noturnas, realizadas por processos de ETL convencionais. Novas aplicações estão utilizando interfaces de acesso das ferramentas OLAP às fontes de dados operacionais, abordagens de extração e carga incremental de dados (JÖRG e DESSLOCH, 2009), ou a utilização de técnicas e algoritmos para aumento do desempenho da execução do ETL, como no trabalho de VASSILIADIS *et al.* (2007) que aponta para o uso de particionamento de dados e paralelismo intra-processo como estratégia para melhoria de desempenho em cada execução do processo de ETL.

O aspecto do desempenho é importante em ambientes de tempo real, mas deve ser levada em consideração outra característica deste ambiente onde os usuários do ambiente analítico estão sempre se renovando, e suas decisões dentro das organizações também mudam de acordo com as tendências do mercado e suas diferentes necessidades. É preciso um processo ETL que seja adaptável e configurável, sendo um caminho a investigar as características de personalização do processo de ETL.

## *3. Qualidade de Dados*

Qualidade de dados é um tema cada vez mais em evidência nas organizações, principalmente porque tem grande impacto nas iniciativas de negócios mais estratégicos. É importante salientar que qualidade de dados não é apenas uma questão da área de TI, mas é também uma questão corporativa, visto que seu impacto atinge toda a organização.

Existe também interesse da comunidade científica cada vez maior relacionado ao tema da qualidade de dados, que busca discutir os desafios da área, de forma a melhorar a compreensão da maneira correta para trabalhar os problemas nesta área, as principais preocupações encontradas no domínio e as tendências futuras (SAQID *et al.*, 2008*).*

Segundo AMARAL (2003), a informação é um recurso organizacional crítico, exigindo a adoção de uma estratégia que garanta a qualidade dos dados. A importância da qualidade de dados fica ainda mais evidente em sistemas gerenciais, como os sistemas de apoio à tomada de decisões, onde decisões equivocadas têm seus impactos potencializados devido à abrangência, ao elevado grau de irreversibilidade e ao maior alcance temporal das decisões envolvidas.

O custo da má qualidade de dados pode ser exemplificado pelo caso de uma companhia de seguros com cerca de dois milhões de pedidos por mês, onde cada pedido contém 377 elementos de dados. Suponha que haja uma taxa de erros de 0,001, os dados dos pedidos conterão mais de 754.000 erros por mês, totalizando mais de 9,04 milhões de erros por ano. Se essa companhia considera que 10% dos elementos de dados são críticos para suas decisões e processos de negócios, será necessário corrigir quase um milhão de erros por ano. Se essa empresa estima custo de $10 por erro, o risco a que ela está exposta, em função da má qualidade de dados, é de $10 milhões por ano (ECKERSON, 2002).

Apesar disso, na maioria dos ambientes corporativos que implantam iniciativas de BI, decisões são tomadas baseadas em informações resultantes de consultas analíticas, sem que haja um conhecimento prévio do nível de qualidade dos dados envolvidos, aumentando o risco de resultados inesperados.

O controle de qualidade de dados é uma questão que envolve todas as etapas do fluxo da informação, desde sua criação até seu uso final. Os dados são provenientes de diversas origens, e os usuários sentem necessidade de conhecer a origem dos dados, saber seu significado e quão atuais eles são, de modo a garantir um nível de qualidade de dados que deixe o usuário confiante com as informações obtidas. As informações de um banco de dados são tão boas quanto os dados que residem nele (SAVLA, 2005).

Para AMARAL (2003), a qualidade de dados é um conceito complexo porque possui significados diversos para diferentes pessoas. Analisando-se estudos já realizados e propostas para definição de critérios para a qualidade de dados, podem-se observar diversas definições para expressar este conceito, não havendo um consenso em relação a um conjunto de critérios que pudesse ser sempre utilizado para definir a qualidade. Uma razão para isso é o caráter essencialmente subjetivo da qualidade de dados, cuja avaliação pode variar de acordo com a função do observador, do contexto e dos objetivos da avaliação.

A avaliação da qualidade dos dados depende da perspectiva e da expectativa do usuário sobre os dados. Assim, é importante definir um modelo de qualidade de dados que permita a avaliação quantitativa e qualitativa dos dados, de forma personalizada.

Um modelo de qualidade de dados é necessário, portanto, para definir os elementos e os procedimentos através dos quais a qualidade pode ser avaliada.

## *3.1. Avaliação da Qualidade de Dados em Banco de Dados*

A qualidade dos dados em um banco de dados pode ser avaliada através de duas perspectivas distintas: a quantitativa, ou objetiva, e a qualitativa, ou subjetiva.

A avaliação quantitativa da qualidade é realizada com a utilização de indicadores objetivos sobre os dados armazenados no banco de dados para medição da qualidade, sendo sua medição dependente apenas do dado a ser medido. São observados, por exemplo, valores de domínio, presença de valor, entre outros itens.

Na avaliação qualitativa, indicadores subjetivos são aplicados aos dados armazenados, com o objetivo de capturar a percepção e a expectativa do usuário. A medição qualitativa depende tanto do ponto de vista do usuário quanto do dado a ser

medido. Os indicadores subjetivos são definidos pelos usuários, assim como as características de qualidade desejáveis.

## *3.2. Qualidade de Dados em Data Warehouse*

O Data Warehouse constitui uma importante ferramenta de apoio ao processo de tomada de decisões. Mas, para que se possa tirar vantagem dos recursos do DW de forma satisfatória, é preciso que as informações nele armazenadas sejam confiáveis, ou que, pelo menos o grau de confiabilidade das mesmas possa ser considerado durante o processo de tomada de decisão (OLIVEIRA, 2002).

Em uma base de dados, a qualidade dos dados é um requisito fundamental para uma correta interação com o cliente, bem como para tomadas de decisão baseadas em soluções como *Data Warehouse, Data Mart, Data Mining* etc. A inexistência de dados duplicados em base de dados, o conhecimento do número exato de clientes, a obtenção de uma visão única de cliente ou a segmentação de clientes estão intimamente ligados à qualidade dos dados que os representam (NOVABASE, 2005).

Em seu trabalho, AMARAL (2003) apresenta diversas abordagens que possuem o propósito de implementar a qualidade de dados em *Data Warehouse*. A tabela 3.1 apresenta uma comparação entre essas abordagens.

**Tabela 3.1** - Comparação entre as abordagens para a qualidade de dados no DW.
Fonte: (AMARAL, 2003)

| Autor | Escopo | Nível dos Objetos | Etapas |
|---|---|---|---|
| **David Marco** | Informações sobre a qualidade no nível de linha, nas instâncias de dados. | Instâncias dos dados | ETL |
| **Jeff Rothenberg** | Informações sobre a qualidade no nível de banco de dados, elementos de dados e instâncias de dados. | Modelos, Instâncias dos dados | Todo o ciclo de vida do DW |
| **Inmon** | Não detalha | Não detalha | OLTP,ETL,DW |
| **Kimball** | Informações sobre a qualidade no menor nível de granularidade, ligado à instância do dado, na tabela de fatos | Instâncias dos dados | ETL |
| **Loshin** | Validação das instâncias dos dados frente a regras de qualidade. | Instâncias dos dados | ETL |
| **Larry English** | Metadados sobre a qualidade da definição, do conteúdo e da apresentação dos dados. | Modelos, Instâncias dos dados | DB origem<br>DB referenciado<br>DW |
| **DWQ (Jarke e Vassiliou) e Vassiliadis** | Define um meta-modelo, onde objetivos de qualidade podem ser associados a objetos do DW. | Meta-modelos, Modelos, Instâncias dos dados | Todo o ciclo de vida do DW |

### 3.2.1. Fatores de Qualidade

As diversas propostas definem conjuntos de fatores de qualidade. Entretanto, não existe um consenso em relação ao conjunto de fatores de qualidade de dados mais adequado para avaliar a qualidade de dados em um DW.

Na literatura sobre o assunto, pode-se constatar a ausência de uniformidade em relação à nomenclatura utilizada, e, a presença de diferentes definições para um mesmo fator de qualidade (AMARAL, 2003).

De forma geral, acurácia, completeza, temporalidade, consistência, confiabilidade e interpretabilidade são fatores de qualidade compartilhados pela maioria das propostas. Entretanto, outras características, relacionadas ao observador, ao contexto e à temporalidade, podem influenciar na percepção da qualidade. De acordo com AMARAL e CAMPOS (2004), uma informação pode ser considerada de qualidade pelo administrador do DW por estar preenchida (completeza) e pertencer ao domínio (acurácia sintática), mas pode não ser considerada de qualidade pelo usuário final por se tratar de um valor estimado (precisão).

WANG e STRONG (1996) realizaram um estudo empírico com o objetivo de determinar um conjunto de fatores de qualidade que fosse de importância para os consumidores de dados. Nesta abordagem, quinze fatores de qualidade foram selecionados como os mais relevantes, sendo agrupados em quatro categorias de qualidade: Intrínseca, Contextual, Representacional e Acessibilidade. A Tabela 3.2 correlaciona os fatores de qualidade às suas respectivas categorias de qualidade.

**Tabela 3.2 -** Fatores de qualidade para consumidores de dados.
Fonte: (WANG e STRONG, 1996).

| Categorias de Qualidade | Fatores de Qualidade |
|---|---|
| **Intrínseca** | Credibilidade, Precisão, Objetividade, Reputação |
| **Contextual** | Valor Agregado, Relevância, Utilidade,<br>Atualidade, Completeza, Volume de Dados |
| **Representacional** | Interpretabilidade, Facilidade de Entendimento,<br>Consistência Representacional, Representação Concisa |
| **Acessibilidade** | Acessibilidade, Segurança de Acesso |

A categoria Intrínseca considera que o dado possui qualidade em si mesmo. Na Contextual, a qualidade do dado deve ser considerada dentro do contexto da tarefa. A categoria Representacional enfatiza que o dado deve ser representado de forma concisa e consistente. A acessibilidade avalia a facilidade do acesso e sua segurança.

Diversas abordagens definiram a sua hierarquia de fatores de qualidade, como uma especialização desses fatores, sendo este trabalho bastante referenciado na literatura. A proposta do projeto DWQ utiliza a estrutura hierárquica definida por WANG e STRONG (1996), (STRONG *et al., 1997*), especializando os fatores de qualidade de forma a contextualizá-los ao ambiente de Data Warehouse.

O *Data Warehouse Quality* (DWQ) é um projeto cooperativo do programa *ESPRIT European Communities*, cujo objetivo é o estabelecimento de bases para a qualidade do Data Warehouse. O projeto tem como proposta a união de modelos semânticos da arquitetura do DW para explicitar modelos de qualidade de dados. (JARKE e VASSILIOU, 1997).

A abordagem do Projeto DWQ considera fatores de qualidade de dados específicos para ambientes de DW. Esta abordagem define um modelo para a qualidade de dados, onde os objetivos de qualidade, relacionados aos fatores de qualidade, podem ser associados a objetos do DW. A Figura 3.1 apresenta a hierarquia de fatores de qualidade de dados proposta pelo DWQ.

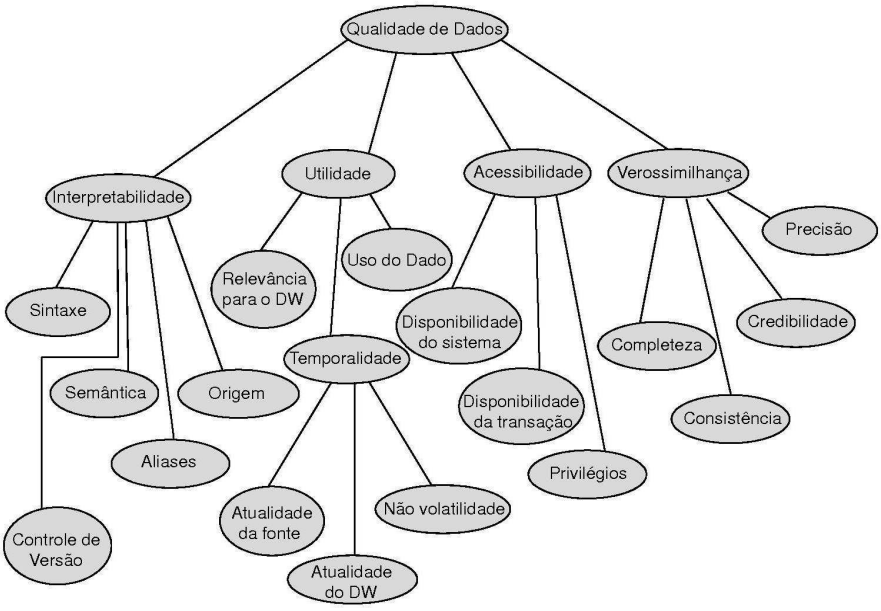


**Figura 3.1 -** Hierarquia de fatores de qualidade de dados (adaptado de JARKE e VASSILIOU, 1997).

Os fatores de qualidade propostos pelo projeto DWQ podem ser classificados nas categorias de qualidade estabelecidas para os consumidores de informação, definidas por WANG e STRONG (1996). A Tabela 3.3 apresenta a correlação entre fatores de qualidade do DWQ e categorias de qualidade para consumidores da informação.

**Tabela 3.3 -** Classificação dos fatores de qualidade do DWQ (adaptado de LEE *et al*., 2001)

| Categorias de Qualidade | Fatores de Qualidade |
|---|---|
| **Intrínseca** | Verossimilhança, Precisão, Credibilidade, Consistência, Completeza |
| **Contextual** | Utilidade, Relevância para o DW, Uso do Dado, Temporalidade, Atualidade da Fonte, Atualidade do DW, Não Volatilidade. |
| **Representacional** | Interpretabilidade, Sintaxe, Semântica, Origem, Aliases, Controle de Versão. |
| **Acessibilidade** | Acessibilidade, Disponibilidade do Sistema, Disponibilidade da Transação, Privilégios. |

As principais vantagens associadas ao uso da abordagem do DWQ explicitadas por JEUSFELD *et al*. (1998) são apresentadas em seguida:

- Qualquer tipo de objeto mensurável pode ser representado no meta-modelo do DW;
- Metas de qualidade podem ser formuladas a partir das perspectivas de um extenso conjunto de tomadores de decisão. Cada tomador de decisão pode avaliar a qualidade do DW a partir de sua perspectiva, validando as consultas de qualidade ligadas às suas metas de qualidade;
- Consultas sobre a qualidade são consultas executáveis no meta-modelo. As respostas delas são o termômetro para um tomador de decisão decidir se a qualidade é apropriada ou não. Consultas sobre a qualidade podem ser inseridas, estendidas, modificadas ou removidas;
- Medidas de qualidade são explicitamente armazenadas no meta-modelo. Materializando-se sucessões de medidas de qualidade do mesmo tipo no meta-modelo, pode-se observar tendências, através de consultas sobre a qualidade apropriadas.

## *3.3. Modelo de Qualidade de Dados Proposto por SIMMHAN*

Para que uma avaliação da qualidade dos dados possa ser realizada, de forma a contemplar os aspectos quantitativos e qualitativos dos dados, faz-se necessário definir o modelo de qualidade de dados que será utilizado como referência na avaliação da qualidade dos dados. Um modelo de qualidade define os elementos e os procedimentos, através dos quais a qualidade será avaliada.

Algumas propostas de modelos para a avaliação da qualidade dos dados podem ser encontradas na literatura sobre o assunto (JARKE e VASSILIOU, 1997; SIMMHAN, 2007; VASSILIADIS, 2000). No contexto deste trabalho, a abordagem escolhida para o modelo de qualidade de dados foi proposta por SIMMHAN (2007), por esta ser capaz de medir quantitativamente a qualidade de dados em seus aspectos objetivo e subjetivo. Ainda, o modelo de qualidade escolhido permite aos usuários expressar requisitos de qualidade para suas aplicações através de perfis de qualidade, e pode ser facilmente aplicado no domínio de DW.

O modelo de qualidade proposto por SIMMHAN (2007) permite aos usuários expressar requisitos de qualidade para suas aplicações através de perfis de qualidade. Estes perfis são compostos por restrições de qualidade individuais, definidas sobre os fatores de qualidade. Para atingir esse objetivo, o modelo é representado usando uma linguagem funcional declarativa. O modelo é avaliado por um sistema único de métricas de qualidade, dentro de um único índice de qualidade numérico, podendo ser utilizado para comparar uniformemente a qualidade dos dados.

Inicialmente, este modelo foi idealizado para domínios de aplicação científica, entretanto, segundo SIMMHAN (2007), o modelo de qualidade não está restrito somente ao campo científico, podendo segundo o autor ser estendido para qualquer outro domínio. A adaptação do modelo a outros domínios deve ser realizada através da definição de métricas de qualidade para as novas classes de qualidade, conforme exigido por esses domínios.

### 3.3.1. Elementos do Modelo de Qualidade

No modelo de qualidade de dados de SIMMHAN são utilizados os conceitos de fator de qualidade, restrições de qualidade, índice de qualidade, métrica de qualidade e perfil de qualidade.

Os *Fatores de Qualidade* são atributos que contribuem para a percepção dos dados pelo usuário. Eles são normalmente resumos de alto nível dos indicadores que são compreendidos pelo usuário final, aparecendo nas instâncias de metadados como nomes de atributos e com um valor associado.

Antes de um atributo poder ser usado dentro de uma métrica de qualidade, o seu valor numérico, ou índice de qualidade, deve ser quantificado.

As *Restrições de Qualidade* são as necessidades de qualidade dos usuários, sendo definidas com base nos atributos de metadados presentes no modelo. As restrições de qualidade podem ser de dois tipos: restrição filtrada (*filtered*) e restrição não filtrada (*unfiltered*). O filtro, na restrição filtrada, atua como uma pré-condição que deve ser satisfeita para que a restrição seja avaliada e contribua para o índice do perfil. As restrições não filtradas não têm pré-condições e são sempre avaliadas, assegurando que a métrica sempre terá um índice atribuído a ela.

O *Índice de Qualidade* é um intervalo de medição padronizado de uma escala numérica, com um limite inferior e um limite superior sobre o valor do índice de qualidade. Uma escala de 15 pontos, com valores no intervalo entre -7 e +7, é utilizada como referência para os valores dos índices. O limite inferior -7 representa o menor nível de qualidade possível. O valor 0 (zero) é o ponto médio da escala e representa uma qualidade neutra, além de ser também o índice padrão. O limite superior +7 representa o maior nível de qualidade. Esta faixa selecionada de índices é configurável e não afeta o modelo de qualidade.

Uma *Métrica de Qualidade* é uma função de agregação que opera sobre os atributos de metadados para produzir um índice de qualidade numérico. As métricas de qualidade agem sobre um conjunto de fatores de qualidade relacionados, com base nas restrições de qualidade especificadas pelo usuário, agregando os índices de qualidade atribuídos a cada um dos fatores de qualidade, dentro de um índice de qualidade para a métrica, chamado *Índice da Métrica*. Os Índices das Métricas são então combinados para formar o índice de qualidade do perfil.

Um *Perfil de Qualidade* é uma coleção de restrições de qualidade definidas para um usuário de uma aplicação. O perfil de qualidade age como uma função que calcula o

resultado da qualidade através de uma ou mais métricas de qualidade fornecidas como argumentos.

Um perfil contém uma ou mais métricas. Métricas contêm nenhuma ou muitas restrições filtradas e uma restrição não filtrada, assim como um peso relativo para a métrica. As restrições não filtradas definem uma ou mais restrições de atributo, e um peso relativo para cada coleção de restrições. As restrições filtradas são similares, mas especificam adicionalmente um filtro como uma precondição para avaliação da restrição. Todas as restrições de atributos são definidas sobre atributos tipados com um peso relativo atribuído à restrição. As restrições de atributos são compostas de uma restrição condicional opcional e uma restrição funcional. A Figura 3.2 apresenta a Backus-Naur Form (BNF) do modelo de qualidade proposto por SIMMHAN.

```
                <profile>  ::=  profile(<metric>{,<metric>})
                 <metric>  ::=  metric({<filteredConstraint>,}
                                <unfilteredContraint>, <weight>)
      <filteredContraint>  ::=  filteredContraint(<filter>,
                                <contraint>{,<constraint>},
                                <weight>)
    <unfilteredContraint>  ::=  unfilteredContraint(<filter>,
                                <contraint>{,<constraint>},
                                <weight>)
             <contraint>  ::=  attributeContraint(<attributeScheme>
                                [,<conditionalContraint>],
                                <funcionalConstraint>,<weight>)
```

**Figura 3.2 -** BNF do modelo de qualidade proposto por SIMMHAN

As restrições de qualidade e o perfil são expressos através da linguagem QCL (*Quality Constraint Language*). Para cada uma das classes de atributos definida é gerado um índice de qualidade, que é então agregado dentro de um índice geral de qualidade.

Para um melhor entendimento, vamos exemplificar, seja um esquema simplificado, para as classes de dados produzidos, utilizando os atributos dos tipos String (S), String Array (#S), numérico (N), e Booleano (B), conforme o que segue:

```
{ ID (S),  author (#S),  publisher (S),  creationTime (N),
  copyrighted (B),  keywords (#S)  }
```

Algumas instâncias μ foram criadas conforme o esquema de dados simplificado, e são mostradas na Figura 3.3. As instâncias estão ordenadas pelo nome do atributo e acompanham seus valores associados que podem estar ausentes.

```
μ1 = {  <ID, urn:data:instance-0001>,
        <author, <Rosa, Hari, Dinesh>>,
        <publisher,Acme>,
        <creationTime, 1159851600>,
        <copyrighted, false>,
        <keywords, <rain, radar, NetCDF>> }
μ2 = {  <ID, urn:data:instance-0002>,
        <author, <Raajesh, Dioni>,
        <publisher,Acme>,
        <creationTime, 1162335600>,
        <copyrighted, false>,
        <keywords, <temperature, radiosonde, GRIB>> }
μ3 = {<ID, urn:data:instance-0003>,
        <author, <Lisa, Sri, Shyam, Giri, Ramyaai>,
        <publisher,Deep Store>,
        <creationTime, 1106823600>,
        <keywords, <temperature, thermometer, GRIB>> }
μ4 = {  <ID, urn:data:instance-0004>,
        <author, <Alek>>,
        <creationTime, 934736400>,
        <copyrighted, true>,
        <keywords, <wind, rawindsonde, ASCII>> }
```

**Figura 3.3** – Exemplo de instâncias de metadados conforme o esquema de dados simplificado

A partir do esquema simplificado, podemos mostrar um exemplo de perfil de qualidade de dados definido para o mesmo, ilustrado na figura 3.4.

```
profile (                                                    start profile . . .
  metric (                                                   start metric . . .
    filteredConstraint(                                  start filteredConstraint . . .
        notFilter ( compareFilter ( <publisher, S>,  = Acme ) ),
          attributeConstraint ( <author, #S>
            compareCondition (author, #S>, ⊇, Lisa , +7 ),
            compareCondition (author, #S>, ⊇,Hari , −7 ),
            defaultValueFunction ( +0 ),
          50                                                 attributeConstraint weight
        ),
  ),                                                          . . . end filteredConstraint
        unfilteredConstraint (                           start unfilteredConstraint . . .
          attributeConstraint ( <creationTime, N>
             compareCondition ( <creationTime, N>, >, 1160000000 , +7 ),
             rangeCondition ( <creationTime, N>, 1150000000 , 1160000000 , +5 ),
             linearScaleFunction ( <creationTime,N> , 0.00000001 , −5 ),
           75                                                attributeConstraint weight
          ),
      100                                                        metric weight
```

```
    )                                          . . . end unfilteredConstraint
  )                                            . . . end metric
)                                              . . . end profile
```

**Figura 3.4** - Exemplo de perfil de qualidade de dados definido sobre um esquema de dados simplificado (SIMMHAN,2007).

No exemplo, valores dos atributos são comparados ao valor de *publisher*, “Acme”. Como na restrição de qualidade está *notFilter* o sistema verifica e avalia as restrições de atributo que não atendem à condição do filtro de comparação, ou seja, avalia todos que forem diferentes de “Acme”. Nas restrições de atributo, é feita a verificação se a mesma contém *author* (⊋)“Lisa” ou “Hari”. Se encontra “Lisa”, a restrição de atributo retorna o valor +7, se encontra “Hari” retorna o valor -7, e a pontuação padrão é definida com valor 0 (zero) atribuída quando nenhuma das restrições de atributo é satisfeita.

Ainda temos uma restrição de atributo definido no atributo *creationTime* e não filtrada (*Unfiltered*) a avaliação é realizada para todas as instâncias de metadados. Primeiramente, as verificações são de comparação se o valor encontrado é maior que um valor em milisegundos (*Comparecondition*), ou se o valor encontrado está entre dois valores (*Rangecondition*), retornando +7 e +5, respectivamente. Caso não exista nenhuma correspondência, na função de escala linear (*linearScaleFunction)* é definido um padrão para cálculo do atributo. Esta última função é definida como (fator × Valor do atributo + pontuação) operação, que neste caso é (0,00000001 × Valor do atributo - 5), e retorna como resultado a pontuação para o atributo.

O Cálculo da qualidade é realizado para a instância μ1, por exemplo, como Publisher é “Acme”, o mesmo não terá avaliação para as restrições filtradas (*Filtered*), uma vez que o *notFilter* inverte a comparação (*Comparefilter*). A restrição filtrada retorna valor padrão 0 (zero) e tem um peso relativo de 50. A restrição não filtrada (*Unfiltered*) correspondente ao *creationTime* retorna um resultado +5, com um peso de 75. A pontuação é calculada para a métrica (0 × 50 + 5 × 75) / (50 + 75) = +3,000.

Para a instância μ3, teremos a avaliação da restrição filtrada (*Filtered*) realizada, já que o publisher não é “Acme”, e o *author* contém “Lisa”, retornando o valor +7 com um peso de 50. Para a restrição não filtrada (*Unfiltered*), as condições não coincidem com o creationTime sendo utilizada função de escala linear (*linearScaleFunction*), que retorna um índice de qualidade (0.00000001 × 1106823600-5 = 6,068) com um peso

de 75. A pontuação de qualidade total para μ3 para o perfil definido é (7 x 50 + 6,068 × 75) / (50 +75) = +6,440.

### 3.3.2. Medição dos Fatores de Qualidade

A medição e a quantificação dos fatores de qualidade podem ser realizadas de diferentes maneiras, partindo-se de valores nativos até encontrar um valor numérico que fique dentro da escala do índice de qualidade.

Na Figura 3.5 são apresentadas as etapas envolvidas na medição do índice de qualidade, iniciando com a identificação dos atributos nos metadados, que são fatores de qualidade, convertendo-os em índices de qualidade através das técnicas de medição direta ou indireta, e agregando-os em um índice geral de qualidade para os dados de um perfil.

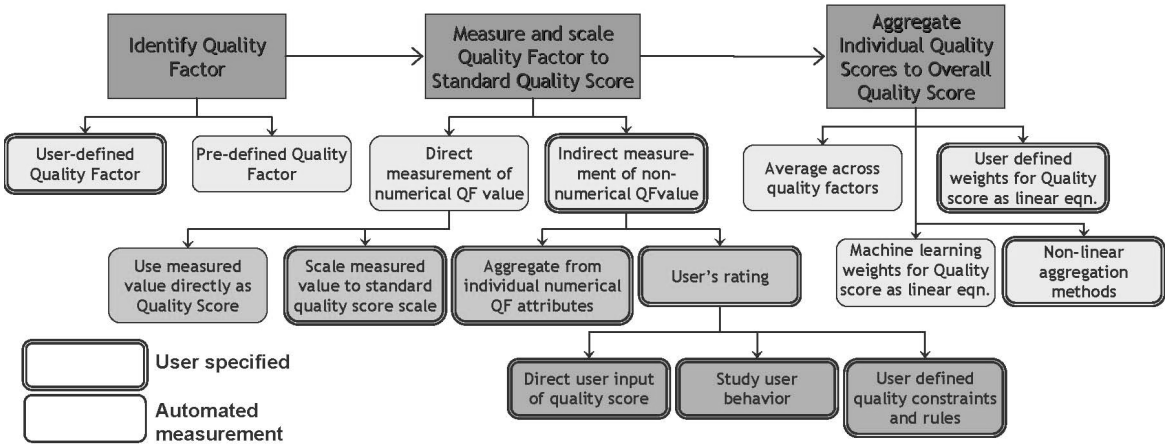


**Figura 3.5 -** Etapas na avaliação da qualidade de dados (SIMMHAN, 2007)

As etapas envolvidas na avaliação da qualidade de dados são descritas em seguida:

1. Identificar os fatores de qualidade na instância de metadados correspondente ao esquema do atributo no perfil QCL.
2. Medir o fator de qualidade através das técnicas direta ou indireta, guiado pela definição do perfil.
3. Definir a escala dos índices de qualidade para fatores individuais de acordo com os pesos relativos definidos pelo usuário.
4. Agrupar os índices de qualidade ponderados das diferentes restrições no perfil dentro de um índice geral de qualidade para o perfil.

### 3.3.3. Técnicas de Medição dos Fatores de Qualidade

Conforme definido no modelo de qualidade de dados de SIMMHAN (2007), dois tipos de técnicas de medição podem ser utilizados para converter fatores de qualidade em índices de qualidade, dependendo do tipo do fator avaliado e da forma desejada de interação com o usuário: medição direta e medição indireta.

**Medição Direta**

Nos casos em que o fator de qualidade a ser medido é numérico, sua medição pode ser realizada pela aplicação da técnica de medição direta, não sendo necessária a intervenção do usuário para tradução subjetiva. Normalmente, o valor do fator de qualidade é em si um índice de qualidade, podendo ser utilizado diretamente.

Entretanto, o valor do fator de qualidade provavelmente está em uma escala diferente da utilizada pelo índice de qualidade. Para manter o padrão da escala de índices de qualidade, precisamos utilizar uma técnica de mudança de escala, visando normalizar o fator de qualidade. Uma forma de normalizar o fator de qualidade é através de mudança para a escala linear, ajustando o valor do fator de qualidade aos intervalos mínimo e máximo do índice de qualidade.

Segundo SIMMHAN (2007), relacionamentos mais complexos entre os fatores de qualidade numéricos e as medições do índice de qualidade podem existir, mas são menos freqüentes. Neste caso, pode ser necessário definir funções de normalização específicas para os fatores de qualidade especializados.

**Medição Indireta**

Quando o fator de qualidade a ser medido é não-numérico ou é de um tipo numérico sem uma função de normalização direta, existe a necessidade de converter esse fator de qualidade para um índice de qualidade, pela aplicação da técnica de medição indireta. Fatores de qualidade, que detêm significado para o usuário e refletem sobre sua percepção da qualidade dos dados, são mensurados através da técnica de medição indireta.

De acordo com SIMMHAN (2007), existem diversas técnicas para realizar medições indiretas, com o usuário desempenhando o papel na definição da tradução subjetiva de fator para índice, de forma explícita ou implícita. A informação pode ser solicitada interativamente, derivada estudando um comportamento, ou ser especificada como restrições e regras definidas pelo usuário.

A Filtragem Colaborativa freqüentemente utiliza uma abordagem interativa, aonde usuários avaliam cada dado através de uma interface de portal. Usuários podem avaliar diferentes atributos de metadados para um dado e completar os dados propriamente ditos sobre a escala de índices de qualidade. Com um número suficiente de avaliações, esses valores podem ser extrapolados para outros dados relacionados com diferentes metadados.

A vantagem da avaliação interativa é a precisão dos índices de qualidade para fatores de qualidade e dados, devido à participação direta do usuário e da comunidade. Entretanto, a sobrecarga do usuário pode ser alta em um grande espaço de tempo com um número limitado de usuários e o número de avaliações obrigatórias para a construção efetiva das funções de normalização através da extrapolação pode ser grande. Este modelo é útil para domínios com grande participação da comunidade e cultura de rede social.

A técnica de estudar o comportamento do usuário e detectar seus padrões de utilização é uma forma menos explícita de avaliação. A gravação de dados freqüentemente acessados, ou aqueles selecionados eventualmente a partir de resultados de consultas, podem dar uma percepção dos dados preferidos pelo usuário e dos atributos de metadados comuns a esses usuários. Técnicas similares para filtragem colaborativa podem ser usadas para derivar a função de normalização para fatores de qualidade.

Embora a carga sobre o usuário seja desprezível nesse método, a avaliação está associada ao conjunto de dados como um todo. Neste caso, um pós-processamento será necessário para discernir as contribuições a partir de fatores qualidade individuais. Portanto, o número de amostras necessárias para tornar esse método tão preciso quanto o anterior pode ser significativamente mais elevado.

Outra técnica de medição indireta é estabelecer restrições e regras, definidas pelo usuário sobre fatores de qualidade, que podem ser resolvidas para índices de

qualidade. Usuários podem fornecer condições para o fator de qualidade sob os quais índices de qualidade específicos são atribuídos. Esta é uma maneira explícita de definir um mapeamento a partir de valores individuais ou faixas para o índice de qualidade para esse fator.

Uma variação simples deste método é vista em sistemas de consultas baseado em atributos que executam classificação. A consulta na forma de valor de atributo, em termos de pesquisa, pode definir uma prioridade para cada termo, para que os resultados sejam ordenados, baseados nesta prioridade. A classificação do conjunto de dados, nos resultados ordenados, é um índice de qualidade relativo do dado.

Esta abordagem dá uma avaliação exata de um fator de qualidade para uma classe inteira de fatores de qualidade e fornece aos usuários um controle de granularidade fina sobre o índice de qualidade que está sendo atribuído. No entanto, ela sofre a desvantagem de que os usuários precisam definir manualmente as restrições de qualidade.

Além das medições dirigidas ao usuário, nós podemos generalizar qualquer dessas medições para uma classe inteira de conjuntos de dados, através das técnicas de extrapolação. O aprendizado de máquina é uma técnica que utiliza um conjunto de exemplos de dados para prever um valor do fator de qualidade não preenchido ou qualquer outro fator de qualidade não mensurável.

A linguagem utilizada para expressar as regras do usuário e mapeamentos deve ser rica o suficiente para captar a percepção da qualidade pelo usuário.

## *3.4. Considerações*

Não existe consenso na literatura sobre a hierarquia de fatores de qualidade de dados mais adequada para avaliar a qualidade de dados em bancos de dados em geral. No contexto mais específico de data warehouses, no entanto, a hierarquia de fatores de qualidade proposta pelo projeto DWQ (JARKE e VASSILIOU, 1997), tem sido bastante adotada e será utilizada para os propósitos deste trabalho.

Um dos fatores presentes na hierarquia DWQ que mais impacta a qualidade dos dados, que é o foco do presente trabalho, é a atualidade do DW, que é uma especialização do fator temporalidade (*timeliness*). A temporalidade em um ambiente

de BI, para efeitos deste trabalho, é definida como o intervalo de tempo entre a ocorrência de um evento no ambiente operacional e a disponibilização dos dados gerados por este evento para atividades analíticas típicas deste ambiente, como, por exemplo, tomada de decisão e auditoria. Em outras palavras, uma temporalidade maior significa um intervalo de tempo maior para a disponibilização do item de dado no DW, ou seja, uma freqüência menor de atualização deste.

Assim, no âmbito deste trabalho, foi realizado um mapeamento da taxonomia de fatores de qualidade do DWQ aos elementos do modelo de qualidade de dados SIMMHAN (2007), de modo a instanciar o modelo ao domínio de DW. O próximo capítulo apresenta o modelo adaptado mais detalhadamente, e como é utilizado no contexto da arquitetura BRAHMA proposta. Também é utilizada a medição indireta para cálculo da qualidade dos dados, visto que os dados envolvidos têm significado para o usuário e refletem sobre sua percepção da qualidade dos dados.

A utilização do modelo de qualidade de dados de Simmhan serviu para personalização das necessidades de cada usuário do DW, identificadas em seus perfis.

# 4. Arquitetura BRAHMA

A solução proposta nesta dissertação é apresentada neste capítulo. A arquitetura BRAHMA instancia o modelo de qualidade de dados proposto por SIMMHAN (2007) com os fatores de qualidade do DWQ e agrupa perfis de usuários com necessidades similares de qualidade de dados, a fim de construir, de forma dinâmica, expressões de extração que vão servir de insumo para configuração de processos de ETL customizados para cada grupo de usuários. As expressões de extração serão compostas por filtros de dados para reduzir a quantidade de dados irrelevantes para cada grupo, levando-se em conta os requisitos de temporalidade definidos neste grupo. Os processos ETL customizados executarão uma extração personalizada e incremental de dados, atendendo de maneira prioritária aos requisitos de temporalidade do ambiente de DW.

## 4.1. visão Geral da Arquitetura

Em ambientes de BI, o processo de ETL é um conjunto de atividades pelas quais os dados operacionais são preparados para o DW, e contempla extrair os dados operacionais das fontes de dados, transformá-los, carregá-los, indexá-los, assegurar a sua qualidade, e publicá-los (KIMBALL e ROSS, 2002). A execução de cada instância de um processo ETL é tipicamente muito custosa.

Desta forma, o objetivo deste trabalho é defender a hipótese de que a redução do volume de dados em cada execução do processo de ETL melhora o seu desempenho, possibilitando atender aos diferentes requisitos de temporalidade dos usuários do DW em ambientes de BI tempo real. Para atingir este objetivo, é proposta a arquitetura BRAHMA.

A arquitetura BRAHMA tem sua importância em ambientes de BI tempo real que podem ser inviáveis utilizando o processo ETL convencional ou com alternativas de projeto ETL. Ela busca a adequação do ambiente do DW às necessidades dos usuários (conforme definido em seus perfis) levando em conta que a informação disponível para tomada de decisões seja a mais atual possível.

Para alcançar tal objetivo, a arquitetura proposta (figura 4.1) compreende um Módulo de Perfis (*Profile Module*) responsável pelo cadastro dos critérios de qualidade

considerados no ambiente, e dos perfis de qualidade definidos por cada usuário em função dos critérios disponíveis. Uma vez definidos os perfis de usuários, o módulo Gerador de Instâncias ETL (*ETL Instance Generator*) contempla diversos sub-módulos: o sub-módulo de Expressões de Extração (*Extraction Expression Module*) inclui componentes para agrupar (*Clustering*) os perfis de usuários semelhantes entre si e para definir expressões de extração (*Extractor*) que serão utilizadas para dinamicamente configurar instâncias distintas do processo de ETL que satisfaçam aos requisitos de qualidade de dados de cada grupo de perfis. Uma expressão de extração reflete o conjunto dos dados relevantes necessários para os usuários de um grupo de perfis, permitindo reduzir o volume de dados desnecessários durante a execução da instância de ETL. Os dados são extraídos das fontes de dados e carregados de forma incremental no DW, em intervalos de tempo menores do que os comumente utilizados, contribuindo não só para o aumento do desempenho do processo de ETL (de forma análoga à proposta na literatura de bancos de dados distribuídos (OZSU e VALDURIEZ, 2001), mas também para o aumento da qualidade dos dados, especialmente no que se refere ao fator temporalidade dos dados, já que os dados serão disponibilizados no DW com maior frequência. As instâncias de ETL configuradas são então agendadas (*Scheduling*) para serem executadas no momento apropriado, em função do critério de temporalidade definido para o grupo de perfis de usuários. A execução das instâncias de processo ETL é monitorada pelo sub-módulo *ETL Instance Enactment*.

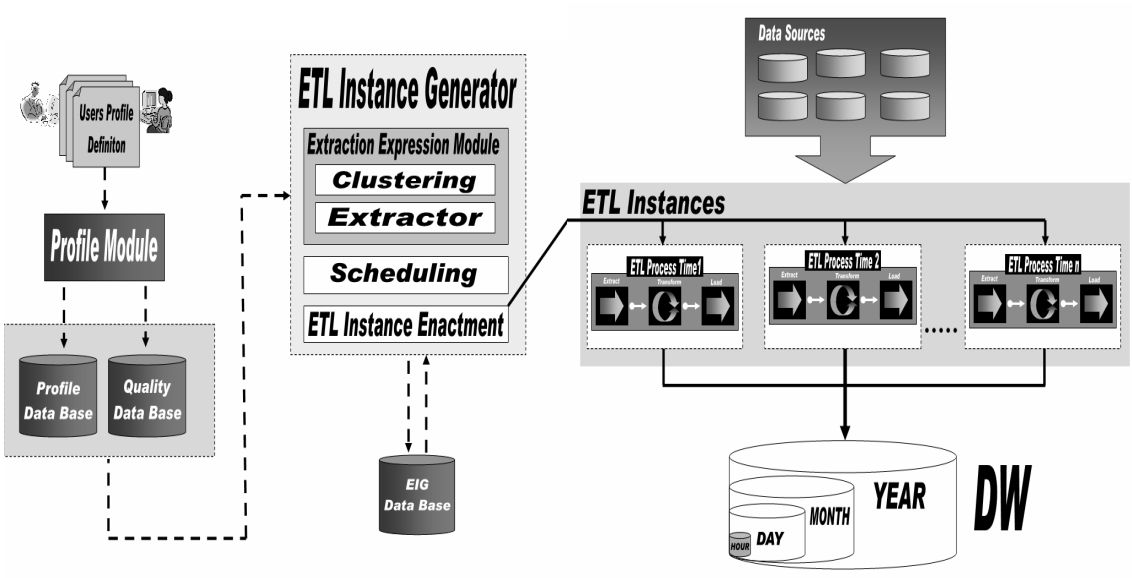


**Figura 4.1** - Arquitetura do ambiente BRAHMA.

## *4.2. Detalhamento da Solução*

Esta seção apresenta a instanciação proposta do modelo de qualidade de dados proposto por SIMMHAN (2007) para contemplar a hierarquia de fatores de qualidade do DWQ, como também detalha o funcionamento de cada módulo da arquitetura BRAHMA.

### 4.2.1. Modelo de Qualidade e Fatores de Qualidade

O modelo de qualidade de dados proposto por SIMMHAN (2007) foi selecionado para compor a solução do ambiente BRAHMA. Este modelo foi apresentado no Capítulo 3. No entanto, por ter abrangência genérica, o modelo de SIMMHAN não contempla critérios de qualidade específicos de domínio. Desta forma, esta seção apresenta a instanciação deste modelo utilizando a hierarquia de critérios de qualidade de dados em DW propostos pelo DWQ (JARKE e VASSILIOU, 1997), por representar fatores de qualidade voltados para o ambiente de BI, que é o foco deste trabalho. A Figura 4.2 apresenta o mapeamento definido para esta instanciação. Os critérios fazem referência às classes de fatores de qualidade, e os fatores são mapeados em fatores de qualidade.

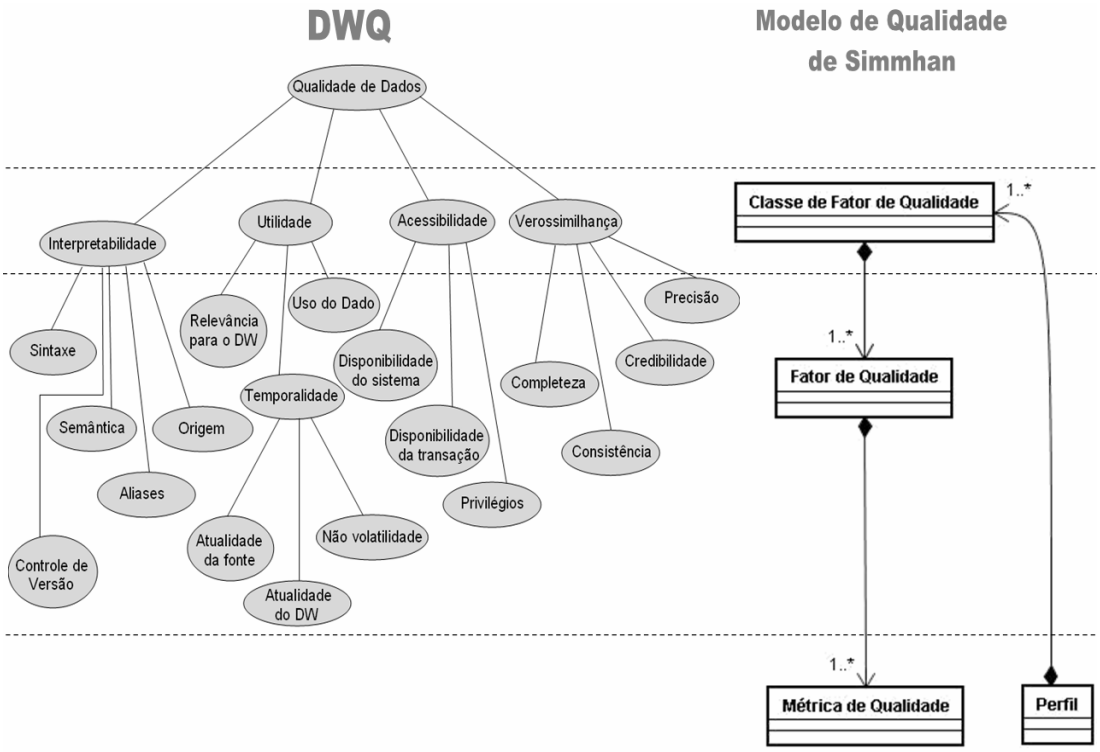


**Figura 4.2** - Adequação dos fatores de qualidade do DWQ ao modelo de qualidade de dados SIMMHAN.

### 4.2.2. Funcionamento da Arquitetura

Nesta seção, o objetivo é detalhar o funcionamento de todos os módulos que compõem a arquitetura, tendo foco nos módulos de perfis e de expressões de extração.

#### 4.2.2.1. Módulo de Perfis

O Módulo de Perfis permite aos usuários definirem seus perfis de qualidade através da indicação de suas necessidades quanto aos fatores de qualidade que julguem ser relevantes. Ainda, o Módulo de Perfis fornece um mecanismo para a avaliação da qualidade dos dados armazenados no DW, utilizando para isso o modelo de qualidade de dados proposto por SIMMHAN (2007), os fatores de qualidade do DWQ e os critérios de qualidade de dados definidos nos perfis de usuários.

O modelo conceitual do módulo de perfis foi construído de acordo com a *Backus- Naur Form* (BNF) traduzida e adaptada a partir da BNF do modelo de qualidade proposto por SIMMHAN (2007), apresentado na Figura 4.3.

**<usuario>::=**usuário({<perfil>})

**<perfil>**::=perfil(<classeFatorQualidade>{,<classeFatorQualidade>})

**<classeFatorQualidade>**::=classeFatorQualidade(<nome>,<fatorQualidade>{,<fatorQualidade>},<peso>)

**<fatorQualidade>**::=fatorQualidade(<nome>,<restricao>{,<restricao>},<peso>)

**<restrição>**::=restrição(<filtro>,<restricaoAtributo>{,<restricaoAtributo>},<peso>)

**<filtro>**::=<filtroComparacao>|<filtroIntervalo>

**<filtroComparacao>**::=filtroComparacao(<atributo>,<sinal>,<valor>)

**<filtroIntervalo>**::=filtroIntervalo(<atributo>,<valor>,<valor>)

**<restricaoAtributo>**::=restricaoAtributo(<atributo>{,<condicao>},<pontuacaoDefault>,<peso>)

**<condicao>**::=<condicaoComparacao>|<condicaoIntervalo>

**<condicaoComparacao>**::=filtroComparacao(<atributo>,<sinal>,<valor>,<pontuacao>)

**<condicaoIntervalo>**::=filtroIntervalo(<atributo>,<valor>,<valor>,<pontuacao>)

**<pontuacaoDefault>**::=pontuacaoDefault(<pontuacao>)

**Figura 4.3** - BNF do modelo de qualidade de dados utilizado

Um *Usuário* pode definir um ou muitos perfis de qualidade. Um *Perfil* é composto por pelo menos uma classe de fatores de qualidade. Cada *Classe de Fatores de Qualidade* possui um nome, um ou mais fatores de qualidade e um peso associado à

classe. Um *Fator de Qualidade* possui um nome, uma ou mais restrições, e um peso associado ao fator. Cada *Restrição* possui um filtro, uma ou mais restrições de atributo, e um peso associado à restrição.

Um *Filtro* pode ser do tipo filtro de comparação ou filtro de intervalo. Um *Filtro de Comparação* possui um atributo, um sinal e um valor. Já um *Filtro de Intervalo* possui um atributo, um valor inferior e um valor superior. Uma *Restrição de Atributo* possui um atributo, opcionalmente uma condição, uma pontuação default e um peso associado.

Uma *Condição* pode ser do tipo condição de comparação ou do tipo condição de intervalo. Uma *Condição de Comparação* possui um atributo, um sinal, um valor e uma pontuação. Uma *Condição de Intervalo* possui um atributo, um valor inferior, um valor superior e uma pontuação. Uma *Pontuação Default* é um valor padrão para uma pontuação. A Figura 4.4 apresenta o modelo conceitual do Módulo de Perfis associado a essa BNF.

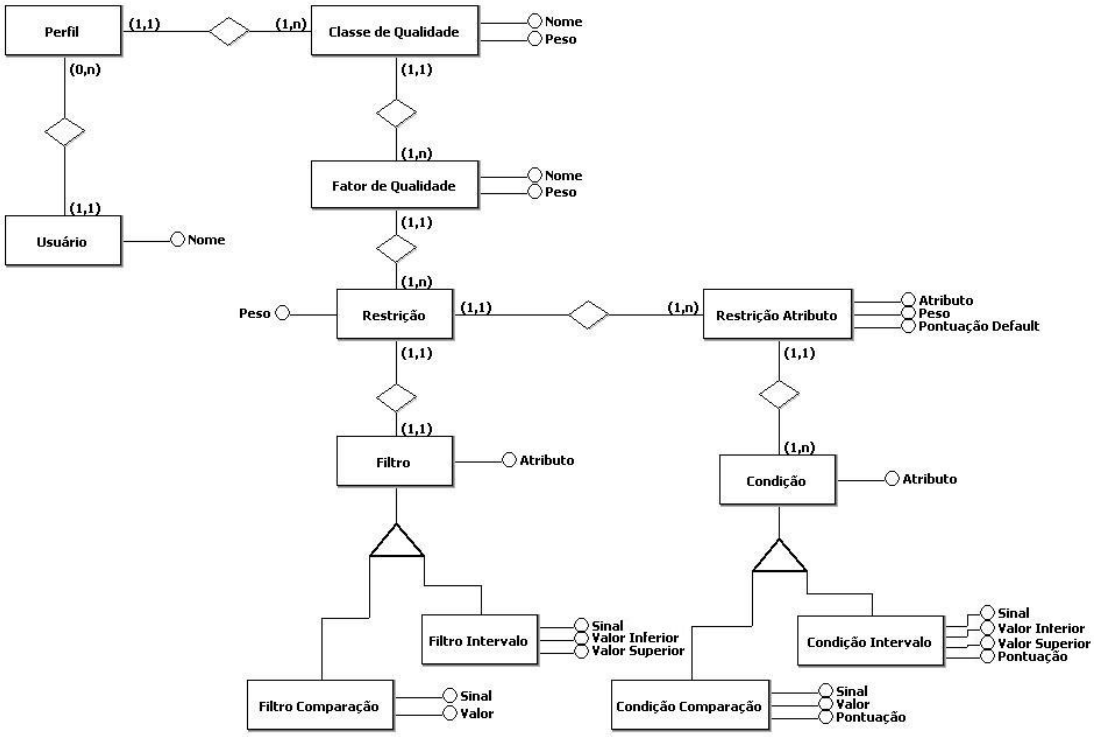


**Figura 4.4** - Modelo conceitual do Módulo de Perfis

Na arquitetura BRAHMA, os perfis dos usuários são armazenados em um repositório de dados denominado *Profile Database*, enquanto as necessidades de qualidade são armazenadas em um repositório de dados denominado *Quality Database* (estas são denominações a nível lógico, não limitando nenhuma restrição quanto ao

armazenamento dos 2 repositórios em uma mesma base de dados a nível físico). O modelo lógico do banco de dados foi projetado com o objetivo de permitir o armazenamento dos perfis de qualidade em um banco de dados relacional. A Figura 4.5 apresenta o diagrama da modelagem lógica para o banco de dados relacional do Módulo de Perfis.

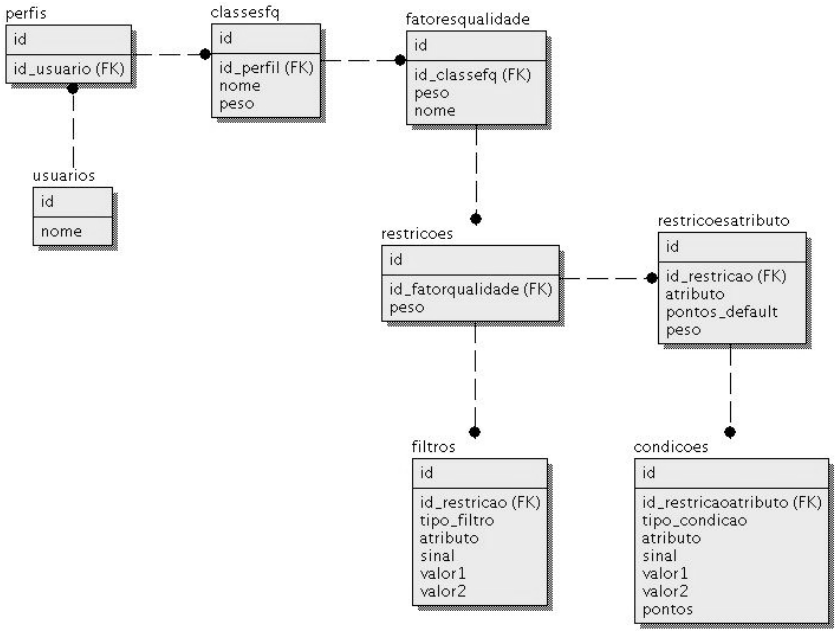


**Figura 4.5** - Modelo lógico do banco de dados relacional do Módulo de Perfis

A avaliação da qualidade dos dados no DW, disponível através do Módulo de Perfis, fornece aos usuários uma maneira mensurável de avaliação da qualidade dos dados no *Data Warehouse*, permitindo a avaliação quantitativa e qualitativa dos dados, de forma customizável, proporcionando um melhor conhecimento da qualidade dos dados envolvidos nos processos de tomada de decisões.

Utilizando os dados armazenados no módulo de perfis, pode-se realizar a avaliação da qualidade, verificando o quanto uma massa de dados está aderente aos fatores de qualidade estabelecidos pelo usuário no seu perfil.

A avaliação de qualidade assume uma pontuação na escala de -7 a +7, conforme proposto no trabalho de SIMMHAN (2007). Uma pontuação baixa, próxima de -7, indica que a massa de dados avaliada tem qualidade baixa segundo os fatores de qualidade estabelecidos pelo usuário. Vale lembrar que estes fatores representam a percepção particular de qualidade dos dados para cada usuário. Uma pontuação alta, próxima de +7, indica que a massa de dados de fato adere aos fatores de qualidade

estabelecidos pelo usuário e que, com isso, podem ser realizados os trabalhos desejados em cima destes dados de forma confiável.

A avaliação dos dados do DW segundo um perfil de qualidade ocorre de forma segmentada, também seguindo as diretrizes definidas no trabalho original de SIMMHAN (2007). Primeiro, todos os filtros do perfil, independente da restrição à qual pertençam, são utilizados para a seleção da massa de dados do DW que será avaliada. Em seguida, são iniciados os sub-processos da avaliação que tomam por base as pontuações associadas aos elementos das classes e das condições.

Desta forma, o cálculo da pontuação resultante é feito a partir da média aritmética simples das pontuações das condições de uma mesma restrição de atributo, ou seja, são somadas as pontuações e a soma é dividida pela quantidade de registros avaliados, obtém-se a pontuação de cada restrição de atributo. Na seqüência, o cálculo é feito utilizando a média ponderada sobre as pontuações das restrições de atributo. O resultado é a pontuação da restrição à qual elas pertencem (o cálculo da média ponderada leva em conta o peso da restrição de atributo). Do mesmo modo, a média ponderada da pontuação calculada sobre as pontuações das restrições representa a pontuação do fator de qualidade à qual elas pertencem. O mesmo se aplica à pontuação dos fatores de qualidade e à pontuação das classes, até se chegar à pontuação do perfil de qualidade. Por fim, se obtém a pontuação final da avaliação.

#### 4.2.2.2. Módulo ETL *Instance Generator*

O módulo ETL *Instance Generator* (EIG) é responsável pela especificação, preparação e execução das instâncias de ETL dentro da arquitetura BRAHMA, a partir das necessidades de qualidade de dados definidas nos perfis dos usuários. O seu objetivo é definir instâncias ETL, de modo que cada instância seja responsável pela atualização incremental do DW, de acordo com os perfis de usuários de um grupo.

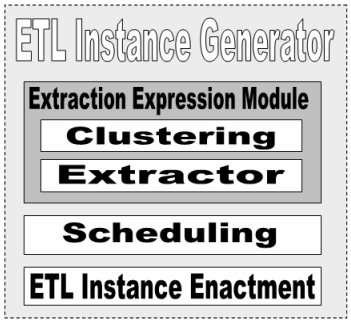


**Figura 4.6** - Componente ETL *Instance Generator*

O módulo EIG é composto por três sub-módulos, a saber: um sub-módulo de Expressões de Extração (MEE), outro para o agendamento das execuções, chamado *Scheduling* e um terceiro chamado ETL *Instance Enactment*, como mostra a figura 4.6. O MEE compreende os componentes *Clustering* e *Extractor* e, por sua importância dentro da arquitetura BRAHMA, apresenta-se como o sub-módulo principal do módulo EIG, e é detalhadamente apresentado a seguir.

**Agrupamento de Perfis de Usuários (*Clustering*)**

O MEE agrupa os perfis de qualidade de dados de acordo com a temporalidade através do *Clustering*, onde um grupo de perfis é composto por todos os perfis de mesmo nível de temporalidade.

O nível de temporalidade de um perfil é definido como a necessidade da periodicidade em que os dados devem ser extraídos para o DW, ou seja, a necessidade de atualização dos dados. Como em ambientes de BI existem usuários que necessitam de informações mais detalhadas com menor periodicidade (isto é, maior freqüência de carga) e usuários que precisam de informações menos detalhadas com uma periodicidade maior (menor freqüência de carga), diferentes níveis de temporalidade devem ser considerados, dependendo da temporalidade definida em seus perfis.

O nível de temporalidade é dado pela granularidade mínima de acordo com o tipo do dado resultante do atributo utilizado no filtro de comparação ou na condição de comparação, aquele que for menor define a granularidade mínina para o fator de qualidade de dados, descrito como *'Temporalidade'*. Os valores para os níveis de temporalidades possíveis podem ser, por exemplo: minuto, hora, dia, mês, ano, e, indicam a freqüência de carga, sendo os níveis de granularidade mais fina (minuto, hora) cada vez mais freqüentes em ambientes de BI 2.0. Um exemplo da definição do nível de temporalidade a partir de um perfil de qualidade é apresentado na seção 4.2.2.3.

**Definição das Expressões de Extração de Dados (*Extractor*)**

A definição das expressões de extração é realizada pelo componente *Extractor*. Para cada grupo de perfis de usuários (i.e., para cada nível de temporalidade), em ordem crescente de nível de temporalidade, é definida uma expressão de extração (EE) e

uma expressão de extração complementar (EEC), com base nos filtros de qualidade especificados pelos usuários do grupo.

Uma expressão de extração (EE) representa as necessidades de dados de todos os usuários do grupo, sendo a EE definida da seguinte forma:

Seja uma relação R(A1,A2,...,An) do esquema das fontes de dados do processo de ETL, onde Ai é um atributo definido sobre um domínio Di, 1 <= i <= n. Um predicado simples Pj sobre R tem a forma

**Pj : Ai θ Valor**

onde θ **Є { =, <, <>, <=, >, >=}**, e **Valor** é escolhido no domínio de **Ai (Valor Є Di)**. Para cada grupo de perfis, uma expressão de extração (EE) é definida como a disjunção dos predicados simples, sobre uma mesma relação, que são mapeados a partir dos filtros de comparação dos perfis que compõem o grupo. Os predicados simples são mapeados em predicados que referenciam o esquema das fontes de dados para compor a EE. A EE resultante vai ser então utilizada pela consulta do módulo de extração do processo ETL para alimentar o DW com tais dados.

Além da EE é gerada uma expressão de extração complementar (EEC) para cada agrupamento, que se refere aos dados que não são necessários aos usuários daquele grupo, correspondendo à negação da EE.

**EEC = not (EE)**

A EEC é utilizada para garantir a não repetição de uma extração já realizada e a totalidade dos dados no DW, ao final do processo ETL. A expressão de extração complementar não se aplica aos filtros relacionados ao fator de qualidade de dados de Temporalidade, somente considera os demais fatores, pois a EEC será utilizada como base da EE de temporalidade superior, que tem outra temporalidade definida.

Os agrupamentos são tratados pelo módulo *Extractor* em ordem crescente de nível de temporalidade, a fim de definir a EE de cada grupo. À EE de cada nível de temporalidade adiciona-se a EEC do nível de temporalidade inferior. Ao executar a instância do processo ETL para um nível de temporalidade, são atualizados no DW os dados relevantes para os usuários do grupo correspondente. O subconjunto de dados relevantes para este grupo é definido pela sua expressão de extração, configurando um processo de atualização incremental dos dados do DW, como ilustra a figura 4.7.

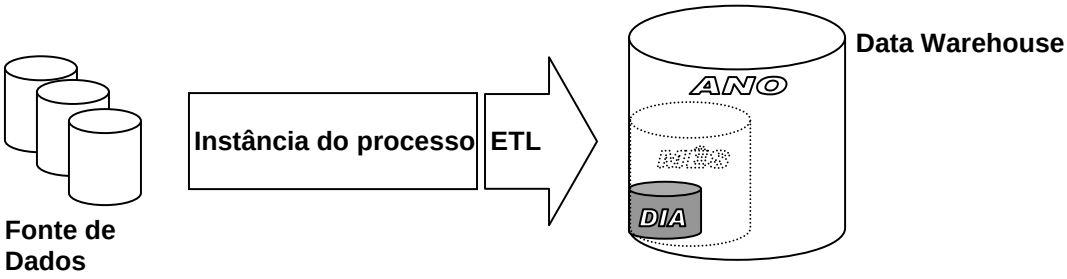


**Figura 4.7** - Atualização incremental dos dados de um DW

Por exemplo, suponha que existam na *Profile Database* perfis de usuário cadastrados de 2 níveis de temporalidade distintos, DIA e MÊS[1]. Serão formados, portanto, 2 grupos de perfis de usuários, agrupando os perfis de cada nível. Para cada grupo, será definida uma EE e uma EEC, e agendadas as execuções das instâncias ETL. A instância de ETL 1 vai ser executada diariamente, extraindo para o DW o subconjunto de dados necessários apenas para os usuários do seu grupo (EE do nível DIA). A instância de ETL 2 vai ser executada mensalmente, extraindo para o DW a união do subconjunto de dados necessários para os usuários do seu grupo (EE do nível MÊS) com o subconjunto dos dados obtidos pela EEC do nível DIA.

A definição da expressão de extração para cada grupo de perfis de usuários considera apenas a atualização incremental dos dados das tabelas de fatos no DW. As tabelas de dimensão devem ter a carga integral realizada, pois contêm as chaves primárias dos relacionamentos com as tabelas de fatos. Como as consultas no DW tipicamente envolvem agrupamentos não previstos segundo uma combinação das dimensões, e as restrições definidas sobre uma dimensão não se aplicam às demais dimensões, a carga das dimensões deve necessariamente ser total.

A carga incremental de dados apenas nas tabelas de fatos não representa, a princípio, uma limitação no aumento do desempenho a ser alcançado pela proposta, se considerarmos o potencial para aumento no desempenho das consultas a serem executadas sobre o DW. De fato, as tabelas de fatos em um DW são as que tipicamente concentram o maior volume de dados, portanto uma redução do volume de dados desnecessários nesta tabela pode trazer ganhos significativos. As tabelas de dimensão, por outro lado, normalmente possuem volume reduzido e não sofrem cargas freqüentes, portanto propõe-se que suas cargas sejam totais.

[1] Em ambientes de BI 2.0, as temporalidades encontradas terão tipicamente granularidade menor, por exemplo, “5 MINUTOS” (atualizações no DW a cada 5 minutos) ou “HORA”. As temporalidades “DIA” e “MÊS” são utilizadas ao longo deste trabalho para efeitos ilustrativos, sem perda de generalidade da proposta.

**Agendamento e Execução das Instâncias de ETL (*Scheduling e Enactment*)**

Depois de definidas as EE, o módulo *Scheduling* agenda a execução de 1 instância de ETL para cada grupo, com periodicidade de execução correspondente ao seu nível de temporalidade (por exemplo, se houver um nível de temporalidade “HORA”, a instância de ETL correspondente será agendada para executar de hora em hora). O módulo *ETL Instance Enactment* é responsável por monitorar a programação definida pelo *Scheduling* e disparar as execuções das instâncias de ETL sempre que necessário. A base de dados EIG *Database* armazena as expressões de extração, as expressões de extração complementar e as informações de agendamento dos processos de ETL.

### 4.2.2.3. Exemplo de aplicação da proposta sobre o esquema TPC-H

Para facilitar o entendimento da arquitetura BRAHMA, faz-se necessário exemplificar o seu uso, principalmente os passos implementados pelo módulo de expressões de extração. O exemplo ilustrado foi definido sobre um cenário do benchmark TPC-H (TPC, 2008).

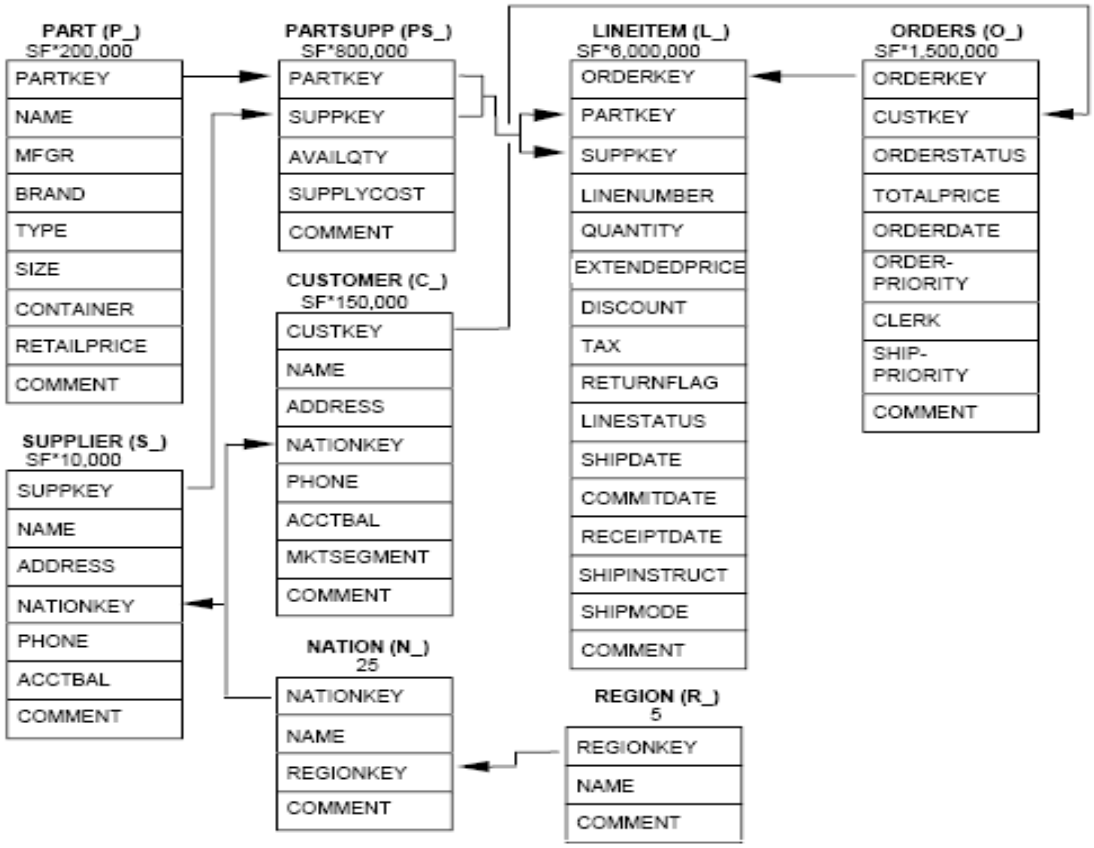


**Figura 4.8 –** Esquema TPC-H original

A Figura 4.8 apresenta o esquema TPC-H original, e, a figura 4.9 apresenta o esquema dimensional do DW utilizado no ambiente BRAHMA, que foi obtido aplicando-se as transformações propostas em (CHEN *et al.,* 2008) para transformar o esquema original TPC-H em um esquema estrela. As modificações necessárias para a transformação do esquema TPC-H no esquema dimensional são apresentadas no capítulo 5.

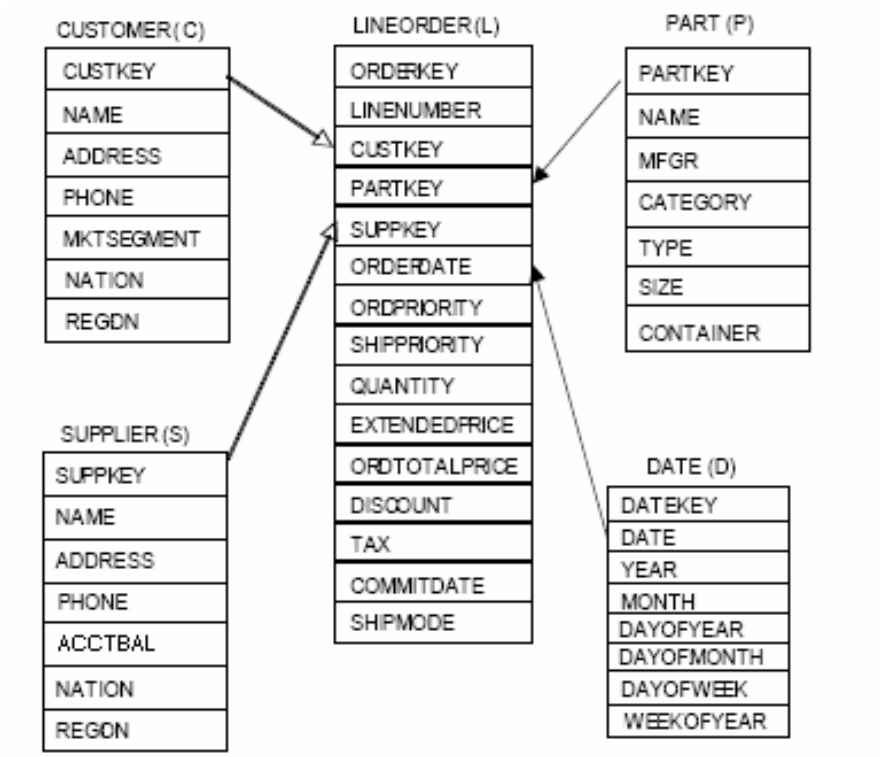


**Figura 4.9 –** Esquema dimensional adaptado de CHEN (2008).

Sejam os perfis Perfil 1, Perfil 2 e Perfil 3 definidos como nas figuras 4.10, 4.11 e 4.12, respectivamente, e que buscam representar papéis típicos em um DW (decisor de nível tático, auditor de nível tático e decisor de nível estratégico). Tais perfis podem ser cadastrados no Módulo de Perfis da implementação da arquitetura BRAHMA.

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
            filter (compareFilter ( C.NATION = 'BRAZIL' )),
                  attributeConstraint (C.MKTSEGMENT,
                        compareCondition ( C.MKTSEGMENT = 'AUTOMOBILE', +7 ),
                        compareCondition ( C.MKTSEGMENT ='HOUSEHOLD', +3 ),
                        defaultValueFunction ( +0 ), 60),
      filteredConstraint(
            filter (compareFilter ( C.NATION = 'ARGENTINA' )),
                  attributeConstraint (C.MKTSEGMENT ,
                        compareCondition ( C.MKTSEGMENT = 'HOUSEHOLD', +7 ),
```

```
                    compareCondition (  C.MKTSEGMENT ='AUTOMOBILE', +5 ),
                    defaultValueFunction ( +0 ), 40),100) ,30)

  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( YEAR(L.ORDERDATE)= YEAR(TODAY() )),
            attributeConstraint ( L.ORDERDATE,
                compareCondition ( MONTH(L.ORDERDATE) = MONTH(TODAY()),  +7 ),
                compareCondition ( MONTH(L.ORDERDATE) < MONTH(TODAY()), +2 ),
                defaultValueFunction ( +0 ), 100),100),70) )
```

**Figura 4.10** - Perfil 1 - Decisor Tático típico

```
profile(
  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( MONTH(L.ORDERDATE)=MONTH(TODAY()) )),
            attributeConstraint ( L.ORDERDATE,
                compareCondition ( L.ORDERDATE = TODAY(),    +7 ),
                compareCondition ( L.ORDERDATE = TODAY()-1,  +3 ),
                compareCondition ( L.ORDERDATE = TODAY()-2,  +1 ),
                defaultValueFunction ( -7 ), 100),100),20) )

class(name='VEROSSIMILHANÇA'
    factor (name='CREDIBILIDADE'
      filteredConstraint(
          filter (compareFilter ( C.NATION='PERU' )),
            attributeConstraint ( L.ORDERPRIORITY,
                compareCondition ( L.ORDERPRIORITY<> NULL,    +7 ),
                defaultValueFunction ( -7 ), 100),100),80) )
```

**Figura 4.11** - Perfil 2 – Auditor Tático típico

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( C.REGION = 'AMERICA' )),
              attributeConstraint (C.MKTSEGMENT,
                    compareCondition (C.MKTSEGMENT = 'BUILDING', +7 ),
                    compareCondition (C.MKTSEGMENT ='FURNITURE', +3 ),
                    defaultValueFunction ( +0 ), 50),
      filteredConstraint(
          filter (compareFilter ( C.REGION = 'EUROPE' )),
              attributeConstraint (C.MKTSEGMENT,
                    compareCondition (C.MKTSEGMENT = 'MACHINERY', +7 ),
                    compareCondition (C.MKTSEGMENT ='BUILDING', +3 ),
                    defaultValueFunction ( +0 ), 50),100),50)
  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( YEAR(L.ORDERDATE) >= YEAR(TODAY())-1 ),
            attributeConstraint ( L.ORDERDATE,
                compareCondition ( YEAR(L.ORDERDATE) = YEAR(TODAY()),    +7 ),
                defaultValueFunction ( -7 ), 100),100),50) )
```

**Figura 4.12 -** Perfil 3 – Decisor Estratégico típico

O processo de construção de expressões de extração e expressões de extração complementares é realizado como se segue.

O componente de *Clustering* utiliza as definições do fator de Temporalidade dos perfis armazenados no *Profile Database*, para agrupá-los segundo o nível de temporalidade. O nível de temporalidade de um perfil é definido pela granularidade mínima do tipo do dado do atributo utilizado no filtro de comparação ou condição de comparação do fator Temporalidade, e mapeado para um valor em uma sequência de valores como na tabela 4.1:

**Tabela 4.1** - Mapeamento para níveis de temporalidade.

| Granularidade mínima do tipo do dado | Nível de Temporalidade |
|---|---|
| Dia | 0 |
| Mês | 1 |
| Ano | 2 |

Para o Perfil 1 (figura 4.10) temos a expressão do filtro de comparação: *filter (compareFilter ( YEAR ( L.ORDERDATE ) = YEAR ( TODAY() ) ) )*, e a expressão da condição de comparação: *compareCondition ( MONTH ( L.ORDERDATE) = MONTH (TODAY()), +7),* quando utilizamos a função *YEAR()* temos algo como, '1996', referenciando a unidade de tempo ano, já quando utilizamos a função *MONTH()* estamos trabalhando com o formato AAAAMM (exemplo: '199612') , referindo-se ao mês, portanto, o nível de temporalidade do perfil é mensal, e foi definido pela menor granularidade encontrada para o fator de qualidade de dados, descrito como *'Temporalidade'*.

Analogamente, define-se o nível de temporalidade do Perfil 2 (figura 4.11) como diária e para o Perfil 3 (figura 4.12) o nível de temporalidade anual. Ao final do processo do *Clustering* cada grupo é composto por todos os perfis de mesmo nível de temporalidade.

O processo continua no módulo de expressões de extração. O componente *Extractor* inicia o processo de construção da expressão de extração e da expressão de extração complementar a cada nível de temporalidade, em ordem crescente.

O predicado da EE relativo ao fator temporalidade é atribuído pelo próprio filtro ou condição identificado como *'Temporalidade'*, ou seja, se a temporalidade é dia então teremos o predicado, como segue:

```
p1: LINEORDER.ORDERDATE = TODAY()
```

No nosso exemplo o grupo de nível 0 de temporalidade é composto pelo Perfil 2, cujo filtro de comparação para relação *CUSTOMER* é:

```
filter (compareFilter ( C.NATION = ‘PERU’ ))
```

A partir deste filtro de comparação obtém-se o seguinte predicado simples para a relação CUSTOMER:

```
p2: CUSTOMER.NATION = ’PERU’
```

Em seguida dando origem a seguinte expressão de extração:

```
CUSTOMER.NATION = ’PERU’ AND LINEORDER.ORDERDATE = TODAY()
```

Esta expressão de extração será adicionada à cláusula `where` da consulta SQL que extrai os dados operacionais das fontes:

```
where ...
    and (c_nation=‘PERU’) and (l_orderdate=today())
```

A expressão de extração complementar diária é definida como a negação da expressão de extração. No exemplo, *NOT (p2),* que nada mais é que a negação do predicado p2.

```
CUSTOMER.NATION !=‘PERU’
```

O predicado p1 que define a temporalidade só é aplicado para EE, o mesmo não se aplica para geração da EEC, visto que a utilização da EEC será no nível de temporalidade superior, que tem outra temporalidade definida.

Ao executar o processo de ETL para o nível diário de temporalidade, utilizando a primeira expressão de extração, o DW terá o conteúdo mostrado na figura 4.7, ou seja, apenas os dados referentes às necessidades diárias dos usuários.

Seguindo ao próximo nível com temporalidade mensal, teremos o predicado, como segue:

```
p3: MONTH(LINEORDER.ORDERDATE) = MONTH(TODAY())
```

Ainda no nível com temporalidade mensal, temos para o Perfil 1 os filtros de comparação para relação CUSTOMER:

```
filter (compareFilter ( C.NATION = ‘BRAZIL’ ))
filter (compareFilter ( C.NATION = ‘ARGENTINA’ ))
```

A partir destes filtros de comparação obtêm-se os seguintes predicados simples para a relação *CUSTOMER*:

```
p4: CUSTOMER.NATION = ’BRAZIL’
p5: CUSTOMER.NATION = ’ARGENTINA’
```

O que dá origem à seguinte expressão de extração:

```
CUSTOMER.NATION = ’BRAZIL’ OR CUSTOMER.NATION = ’ARGENTINA’ AND
MONTH(LINEORDER.ORDERDATE) = MONTH(TODAY())
```

Esta expressão de extração será adicionada à expressão de extração base, e, para garantir a não repetição de extrações já realizadas, deve ter adicionada a cláusula de expressão de extração complementar do nível anterior, sendo as cláusulas para EE mensal as que seguem:

```
where ...
    and (c_nation=’BRAZIL’  OR c_nation=’ARGENTINA’)
    and (MONTH(l_orderdate)= MONTH(today()))
    and NOT (c_nation=‘PERU’)
```

A expressão de extração complementar mensal é definida como a negação da expressão de extração. No caso, *NOT (p4 OR p5)*, que nada mais é que o *AND* da negação de cada predicado, ou seja, *(NOT p4) AND (NOT p5)*. O predicado p3 que define a temporalidade da extração não se aplica para geração da EEC.

A expressão de extração complementar é utilizada sempre no nível de temporalidade superior, de forma a complementar a expressão de extração deste grupo, garantindo que os dados já extraídos não serão extraídos novamente.

Para encerrar o processo chegamos ao último nível de temporalidade, o anual, teremos o predicado, como segue:

```
p6: YEAR(LINEORDER.ORDERDATE) = YEAR (TODAY())
```

Neste nível os filtro de comparação na relação *CUSTOMER* são:

```
filter (compareFilter ( C.REGION = ‘AMERICA’ ))
filter (compareFilter ( C.REGION = ‘EUROPE’ ))
```

Conforme os filtros de comparação obtêm-se os seguintes predicados simples para a relação *CUSTOMER*:

```
p7: CUSTOMER.REGION = ’AMERICA’
p8: CUSTOMER.REGION = ’EUROPE’
```

O que dá origem à seguinte expressão de extração:

```
CUSTOMER.REGION = ’AMERICA’ OR CUSTOMER.REGION = ’EUROPE’ AND
YEAR(LINEORDER.ORDERDATE) = YEAR(TODAY())
```

Esta expressão de extração será adicionada a expressão de extração base, e, para garantir a não repetição de extrações já realizada deve ter adicionada, as expressões de extração complementares dos níveis anteriores, sendo a EE como se segue:

```
where ...
    and (c_region=’AMERICA’  OR c_region=’EUROPE’)
    and (YEAR(l_orderdate)= YEAR(today()))
    and (NOT (c_nation=’BRAZIL’) AND NOT (c_nation=’ARGENTINA’))
    and NOT (c_nation=‘PERU’)
```

Porém, ainda é necessário obter a expressão de extração complementar do nível 2, definida como a negação da expressão de extração. No caso, *NOT (p7 OR p8)***,** que nada mais é que o AND da negação de cada predicado, ou seja, *(NOT p7) AND (NOT p8)*.

A expressão de extração complementar final que preencherá o DW com todos os dados é o conjunto de todas as EEC e sua periodicidade de carga deve considerar o maior nível de temporalidade, porém só devendo ser realizada após a extração do nível mais alto ter sido concluída, ou seja, no exemplo dado esta deve ser realizada após a carga anual de dados. A EEC final será conforme segue:

```
where ...
   and (NOT (c_region='AMERICA')  AND NOT (c_region='EUROPE'))
   and (NOT (c_nation='BRAZIL') AND NOT (c_nation='ARGENTINA'))
   and NOT (c_nation='PERU')
```

## *4.3. Considerações*

A arquitetura BRAHMA proposta neste capítulo compreende mecanismos para o agrupamento de usuários do DW segundo os requisitos de temporalidade dos dados definidos em seus perfis. A partir deste agrupamento, a arquitetura propõe mecanismos para definição automática de expressões de extração (EE) e de expressões de extração complementar (EEC) de dados que configuram instâncias de processos ETL para cada nível de temporalidade considerado. As instâncias são então configuradas para serem executadas com a periodicidade necessária, atualizando o DW de forma incremental com o subconjunto de dados relevantes para os usuários do DW. A extração incremental de dados, de acordo com os diferentes requisitos de temporalidade dos usuários, auxilia na redução do volume de dados a cada execução do processo de ETL, deixando o DW o mais atualizado possível, para o que realmente seus usuários necessitam a cada nível de temporalidade.

Neste capítulo foi apresentado um exemplo de uso da arquitetura BRAHMA, ilustrando a construção das EE e EEC para o contexto da base de dados do benchmark TPC-H. Com base nestas expressões é possível realizar a extração incremental e reduzir o volume de dados a cada execução.

# *5. Implementação e Cenário de Avaliação da Arquitetura BRAHMA*

A implementação da proposta é apresentada neste capítulo. Na seção 5.1 são apresentados o Módulo de Perfis e o Módulo de Expressões de Extração desenvolvidos para construção do ambiente BRAHMA, na seção 5.2 descrevemos o cenário para avaliação da arquitetura, ferramentas de apoio para geração dos dados, os perfis considerados para a avaliação e a especificação do processo ETL. A seção 5.3 apresenta as expressões de extração montadas a partir dos perfis e apresenta como são realizadas as cargas no DW.

## *5.1. Implementação da Arquitetura BRAHMA*

Para a implementação da arquitetura BRAHMA foram desenvolvidos o Módulo de Perfis implementado através de programação, e o Módulo de Expressões de Extração implementado através de scripts encontrados nos anexos.

### 5.1.1. Módulo de Perfis

O Módulo de Perfis foi concebido e especificado dentro do escopo do presente trabalho, e sua implementação está descrita com maiores detalhes em (AQUINO e JUNIOR, 2009). O objetivo do Módulo de Perfis é permitir aos usuários gerenciar perfis de qualidade, incluindo funcionalidades para cadastro, alteração, remoção e busca de perfis, além de permitir avaliar a qualidade dos dados armazenados no DW segundo os perfis existentes e carregados pelo processo de ETL através da extração dos dados operacionais das bases de dados dos sistemas transacionais.

Os perfis de qualidade a serem usados neste projeto, em particular no Módulo de Perfis, seguem fundamentalmente o mesmo o propósito dos perfis de qualidade utilizados no modelo de qualidade proposto por SIMMHAN (2007). Porém, devido ao fato de serem aplicados em um domínio totalmente diferente, a estrutura do perfil de qualidade possui algumas diferenças. O modelo de qualidade proposto em (SIMMHAN, 2007) tem o propósito de avaliar a qualidade subjetiva de dados derivados, para aplicações científicas. Tal modelo de qualidade é composto de métricas de qualidade que incluem dados de proveniência, metadados intrínsecos ao modelo em questão, informações sobre qualidade de serviço e percepção da

comunidade para estimar, numericamente, a qualidade dos dados, permitindo com isso, que cientistas selecionem os conjuntos de dados de melhor qualidade para as suas aplicações (SIMMHAN, 2007). Para o ambiente BRAHMA o domínio escolhido foi o de um ambiente de negócios com dados operacionais fornecidos pelo benchmark TPC-H (TPC, 2008).

A tela principal do módulo, mostrada na figura 5.1, permite o acesso a todas as suas funcionalidades, incluindo cadastrar usuários, classes e fatores de qualidade, perfis, e avaliação da qualidade dos dados do *Data Warehouse*.

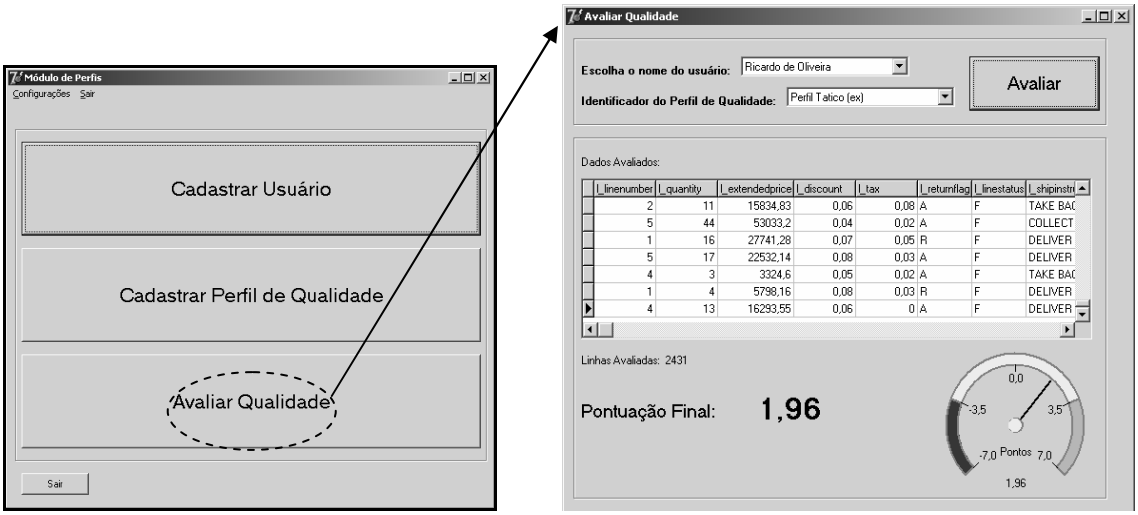


**Figura 5.1** – Tela inicial do sistema e Tela de avaliação da qualidade

CADASTRAR USUÁRIO: Ao entrar no sistema, caso o usuário ainda não tenha sido criado, o mesmo deve clicar em Cadastrar Usuário, no menu principal. Em seguida, será apresentada a tela de cadastro de usuários. Basta entrar com o nome desejado e clicar em Cadastrar. Uma mensagem de confirmação será mostrada informando que o usuário foi cadastrado com sucesso.

CADASTRAR PERFIL DE QUALIDADE: Para cadastrar um perfil de qualidade para um usuário basta selecionar a opção Cadastrar Perfil de Qualidade no menu principal. Ao selecionar esta opção será apresentada a tela de cadastro de Perfil de Qualidade. Primeiro, deve-se selecionar o nome do usuário ao qual o perfil será associado e, em seguida, deve-se clicar em Criar Perfil. As instruções são apresentadas nas telas para que o perfil de qualidade seja cadastrado corretamente. Ao término do cadastro do Perfil de Qualidade, o usuário deve clicar no botão Concluir, e o Perfil será salvo no Banco de Dados da aplicação.

AVALIAR QUALIDADE: Para avaliar a qualidade de uma massa de dados, em relação a um determinado perfil de qualidade, basta selecionar a opção Avaliar Qualidade no menu principal. Ao entrar na tela de Avaliação de Qualidade, selecionar o usuário e Perfil de Qualidade deste usuário. A escolha do Perfil de Qualidade irá determinar o subconjunto dos dados do Data Warehouse que serão apresentados, pois somente serão avaliados os dados que satisfizerem os filtros estabelecidos no Perfil de Qualidade. Após selecionar o Perfil de Qualidade desejado, o usuário deve clicar no botão Avaliar. Cada linha da massa de dados selecionada será individualmente avaliada, segundo os fatores do Perfil de Qualidade, e sua pontuação será agregada para formar a pontuação final.

O Módulo de Perfis foi implementado na linguagem de programação Delphi em sua versão 7 (DELPHI, 2009), sobre o SGBD PostgreSQL em sua versão 8.1 (POSTGRESQL, 2009). Os scripts de criação da estrutura de tabelas para suporte ao módulo de perfis estão disponíveis no anexo I.

### 5.1.2. Módulo de Expressões de Extração

As funcionalidades do Módulo de Expressões de Extração foram implementadas através de scripts PL-SQL, invocados através de uma interface gráfica ilustrada na figura 5.2. Foram implementados scripts para criar as estruturas de dados necessárias à arquitetura, agrupar os perfis de usuários, configurar instâncias de ETL, e executar as instâncias de ETL configuradas, carregando o DW. Os scripts criados estão apresentados nos anexos III, IV e V.


**Figura 5.2** – Processo de Agrupamento dos Perfis por Temporalidade (*CLUSTERING*).

Na interface gráfica da figura 5.2, o usuário pode requisitar o agrupamento dos perfis. Quando isto acontece, o script de *clustering* consulta todos os perfis existentes na *Profile Database*, determina quantos níveis de temporalidade distintos existem cadastrados, e define um grupo de perfis para cada nível contendo todos os perfis de mesmo nível. Após isso, o script busca todos os filtros cadastrados (predicados simples das *filtered constraints* dos perfis de usuário) para cada nível de

temporalidade. De posse dos perfis agrupados, o próximo passo é utilizar os filtros estabelecidos para montar a Expressão de Extração (EE) para cada nível de temporalidade e a Expressão de Extração Complementar (EEC), como ilustra a figura 5.3.

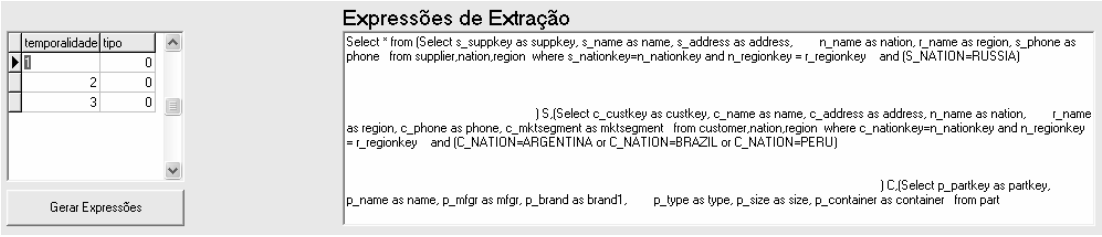


**Figura 5.3** – Processo criação de EE e EEC (*EXTRACTOR*).

## *5.2. Cenário definido para Avaliação da Proposta*

O esquema transacional, para os dados operacionais, utilizado no ambiente de avaliação da qualidade proposto neste trabalho foi estabelecido com base no esquema do benchmark TPC-H (TPC, 2008). A partir deste esquema, construímos um processo ETL de modo a transformar os dados para um esquema estrela, criando assim um cenário para avaliação da proposta. A subseção seguinte explica em mais detalhes porquê consideramos o esquema TPC-H como nossa fonte de dados. Em seguida, a seção 5.2.2 apresenta como foi gerada a massa de dados para avaliação. A seção 5.2.3 apresenta os perfis considerados para a avaliação, e por fim a seção .5.2.4 apresenta a especificação do processo ETL utilizado.

### 5.2.1. TPC-H

O TPC-H é um benchmark padrão, definido pelo *Transaction Processing Performance Council* (TPC), voltado para avaliação do desempenho de sistemas de bancos de dados em ambientes de suporte à decisões. O benchmark especifica o conjunto de dados que deve ser armazenado, um conjunto de consultas analíticas, e fornece um aplicativo de linha de comando para a geração dos dados denominado DBGEN. Entretanto, os arquivos de textos gerados pelo aplicativo são de difícil manipulação, e o esquema dos dados é fixo, dificultando assim, as adaptações necessárias para avaliação de cenários alternativos aos definidos pela especificação. Além disso, as tabelas de dimensão no esquema do TPC-H são normalizadas, caracterizando-o como um esquema floco de neve. A Figura 5.4 ilustra o esquema do TPC-H.

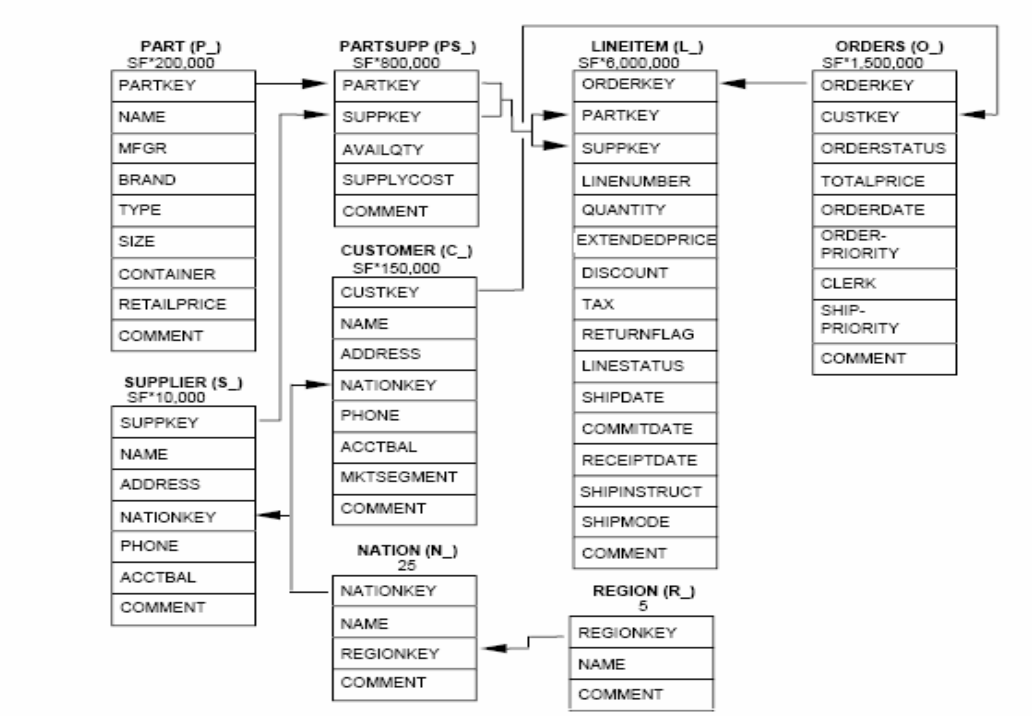


**Figura 5.4 –** Esquema TPC-H

- Os parênteses após cada nome de tabela contêm o prefixo dos nomes das colunas para essa tabela;

- As setas apontam no sentido dos relacionamentos um para muitos entre as tabelas;

- Os números abaixo de cada nome de tabela representam a cardinalidade (número de tuplas) da tabela. Alguns desses números são precedidos por SF, Scale Fator (fator de escala), que é configurado pelo usuário que gera a massa de dados, para obter o tamanho escolhido para a base de dados. A cardinalidade para a tabela LINEITEM é aproximada.

No TPC-H os dados históricos armazenados são relativos a pedidos e a vendas de uma organização. As tabelas *LINEITEM* e *PARTSUPP* são tabelas de fatos, enquanto que as tabelas *CUSTOMER, NATION, ORDERS, PART, REGION* e *SUPPLIER* são tabelas de dimensão. Por apresentar as tabelas de dimensão normalizadas, podemos verificar a presença de atributos de uma dimensão presentes em outras dimensões. Por exemplo, atributos de *SUPPLIER* encontram-se divididos nas tabelas *SUPPLIER*, *NATION* e *REGION*, enquanto que atributos de *ORDERS* encontram-se divididos nas tabelas *CUSTOMER*, *NATION* e *REGION*.

O modelo de dados do benchmark TPC-H foi proposto para a avaliação do desempenho de sistemas de bancos de dados em ambientes de suporte à tomada de decisões, portanto deveria ser adequado para avaliação de desempenho de aplicações OLAP sobre *data warehouses*. Entretanto, alguns aspectos o caracterizam como um modelo de dados transacional, sendo inadequado para uso em um *Data Warehouse*:

- Existência de relacionamento com cardinalidade muitos-para-muitos (*PARTSUPP*), característicos de esquemas relacionais transacionais, entre as dimensões *PART* e *SUPPLIER*;
- Presença de atributos descritivos (*COMMENT*) em tabela de fatos;
- Ausência de uma dimensão tempo;

Conforme apresentado no capítulo 2, o esquema estrela é o modelo multidimensional mais adequado para os ambientes de *data warehouse*. Portanto, manipulações no esquema do TPC-H são necessárias, visando transformá-lo em um esquema estrela, tornando-o mais adequado para utilização no ambiente de avaliação da qualidade dos dados no DW.

Desta forma para realizar estas transformações foi especificado um processo ETL através da ferramenta VisualTPCH desenvolvida por DOMINGUES, CIFERRI e CIFERRI (2008). Neste processo, o esquema dimensional utilizado para avaliação do ambiente BRAHMA de avaliação da qualidade foi criado a partir de adaptações, segundo o trabalho de CHEN (2008), no esquema original do TPC-H. A especificação deste processo ETL está descrita em mais detalhes na seção 5.2.4. As modificações necessárias para a transformação do esquema TPC-H no esquema dimensional utilizado neste trabalho são apresentadas em seguida:

- Descartar a tabela *PARTSUPP* do TPC-H por ser uma segunda tabela fato no esquema;
- Combinar as tabelas *REGION*, *NATION*, *SUPPLIER* na tabela *SUPPLIER*. Esta combinação tem por objetivo a desnormalização do esquema original do TPC-H. Neste processo, as colunas *R_NAME* e *N_NAME*, foram renomeadas para *S_REGION* e S_NATION, respectivamente;
- Combinar as tabelas *REGION*, *NATION*, *CUSTOMER* na tabela *CUSTOMER*. Esta combinação tem por objetivo a desnormalização do esquema original do TPC-H. Neste processo, as colunas *R_NAME* e *N_NAME*, foram renomeadas para *C_REGION* e *C_NATION*, respectivamente;

- Renomear a coluna *P_BRAND* da tabela *PART* para *P_CATEGORY*;
- Descartar a coluna *P_RETAILPRICE* da tabela *PART*, as colunas *O_ORDER*STATUS e *O_CLERK* da tabela *ORDERS*, e a coluna *COMMENT* de todas as tabelas;
- Descartar as colunas *L_RETURNFLAG*, *L_LINESTATUS*, *L_SHIPDATE*, *L_RECEIPTDATE*, *L_SHIPINSTRUCT* da tabela *LINEITEM*;
- Combinar as tabelas *ORDER*, *LINEITEM* na tabela *LINEORDER*. Esta combinação tem por objetivo a desnormalização do esquema original do TPC-H. Neste processo, as colunas *O_CUSTKEY*, *O_TOTALPRICE*, *O_ORDERDATE*, *O_ORDERPRIORITY*, *O_SHIPPRIORITY*, foram renomeadas para *L_CUSTKEY*, *L_ORDTOTAQLPRICE*, *L_ORDERDATE*, *L_ORDERPRIORITY* e *L_SHIPPRIORITY*, respectivamente;
- Criar uma nova dimensão *DATE*, povoando-a com uma massa de dados que considere todas as possibilidades de datas geradas pelo VisualTPCH. Adicionalmente foi preciso adaptar uma coluna da tabela fato (*L_ORDERDATE*), transformando-a em chave estrangeira, fazendo referência à nova dimensão *DATE*.

A Figura 5.5 apresenta o esquema dimensional considerado como “alvo” do processo ETL para avaliação do ambiente BRAHMA de avaliação da qualidade dos dados, transformado a partir do esquema original do TPC-H em um esquema estrela. Os scripts que definem o ambiente dimensional estão apresentados no anexo III.

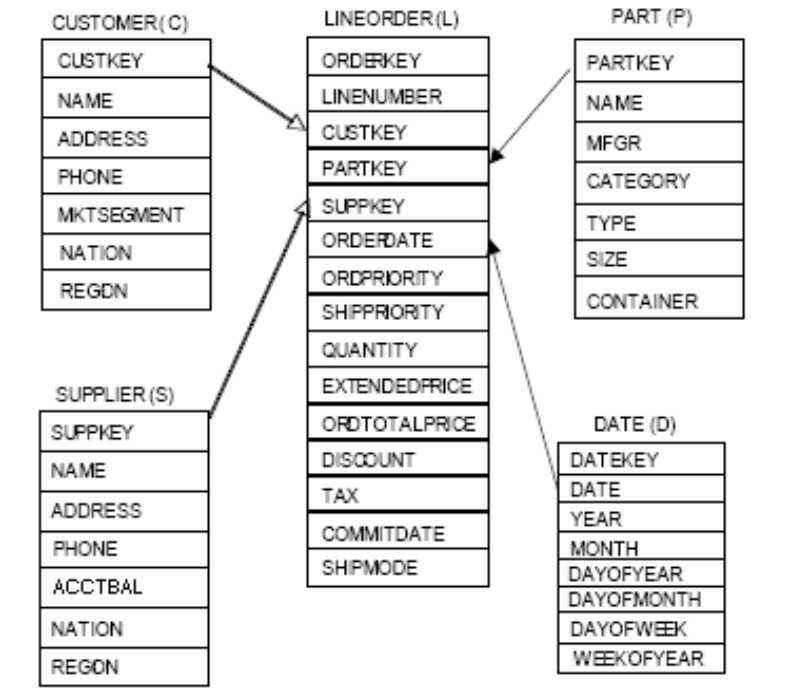


**Figura 5.5 –** Esquema dimensional adaptado de CHEN (2008).

### 5.2.2. Geração dos Dados do Esquema Fonte

A massa de dados gerada utilizando o VisualTPCH, teve como padrão 1GB de informações. As quantidades de registros nas tabelas da fonte de dados transacional da avaliação são informadas na tabela 5.1.

**Tabela 5.1** – Quantidade de registros por tabelas

| Nome da Tabela | Quantidade de Registros |
|---|---|
| *NATION* | *25* |
| *REGION* | *5* |
| *PART* | *200.000* |
| *PARTSUPP* | *800.000* |
| *SUPPLIER* | *10.000* |
| *CUSTOMER* | *150.000* |
| *ORDERS* | *1.500.000* |
| *LINEITEM* | *6.001.204* |

### 5.2.3. Definição de Perfis para Avaliação da Arquitetura

Neste ambiente da arquitetura que envolve BI existem tipicamente usuários que necessitam de informações mais detalhadas (táticos) com menor periodicidade (isto é, maior freqüência), e usuários que precisam de informações menos detalhadas (estratégicos) com uma periodicidade maior (menor freqüência).

No experimento para avaliar a arquitetura BRAHMA um grupo de 6 usuários foi cadastrado no módulo de perfis de qualidade de dados e para cada um dos usuários um perfil, sendo três perfis com temporalidade diária (temporalidade nível 0), como mostram as figuras 5.6, 5.7 e 5.8, mais dois perfis com temporalidade mensal (temporalidade nível 1), mostrados nas figuras 5.9.e 5.10, e um perfil com temporalidade anual (temporalidade nível 2), na figura 5.11.

Os perfis detalham bem as características da temporalidade onde os perfis com menor nível de temporalidade necessitam de informações mais detalhadas e com maior freqüência, o que pode ser caracterizado por decisores e auditores táticos. À medida que avançamos nas necessidades de níveis de temporalidade dos perfis as características se tornam mais estratégicas com informações menos detalhadas.

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( S.NATION='RUSSIA' )),
            attributeConstraint ( L.ORDERPRIORITY,
                compareCondition ( L.ORDERPRIORITY = '1-URGENT',    +7 ),
                compareCondition ( L.ORDERPRIORITY = '2-HIGH',      +5 ),
                compareCondition ( L.ORDERPRIORITY = '3-MEDIUM',    +3 ),
                defaultValueFunction ( +1 ), 50)
            attributeConstraint ( L.SHIPMODE,
                compareCondition ( L.SHIPMODE <> 'MAIL',    +7 ),
                defaultValueFunction ( +3 ), 50), 100), 100), 20)

class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( L.ORDERDATE = TODAY() )),
            attributeConstraint ( L.EXTENDEDPRICE,
                compareCondition ( L. EXTENDEDPRICE >= '15000',    +7 ),
                defaultValueFunction ( -7 ), 100), 100), 100), 80) )
```

**Figura 5.6** - Perfil A cadastrado pelo Usuário1

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( C.NATION = 'BRAZIL' )),
              attributeConstraint (C.MKTSEGMENT,
                  compareCondition ( C.MKTSEGMENT = 'AUTOMOBILE', +7 ),
                  compareCondition ( C.MKTSEGMENT ='HOUSEHOLD', +3 ),
                  defaultValueFunction ( +0 ), 100), 60),
      filteredConstraint(
          filter (compareFilter ( C.NATION = 'ARGENTINA' )),
              attributeConstraint (C.MKTSEGMENT ,
                  compareCondition ( C.MKTSEGMENT = 'HOUSEHOLD', +7 ),
                  compareCondition (  C.MKTSEGMENT ='AUTOMOBILE', +5 ),
                  defaultValueFunction ( +0 ), 100), 40), 100) , 30)

  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( L.ORDERDATE = TODAY() )),
            attributeConstraint ( L.COMMITDATE,
                compareCondition ( L.COMMITDATE <= L.ORDERDATE+60,    +7 ),
                defaultValueFunction ( -7 ), 100), 100), 100), 70) )
```

**Figura 5.7** - Perfil B cadastrado pelo Usuário2

```
profile(
  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( MONTH(L.ORDERDATE) = MONTH(TODAY()) )),
            attributeConstraint ( L.ORDERDATE,
                compareCondition ( L.ORDERDATE = TODAY(),     +7 ),
                compareCondition ( L.ORDERDATE = TODAY()-1,   +4 ),
                compareCondition ( L.ORDERDATE = TODAY()-2,   +1 ),
                defaultValueFunction ( +0 ), 100), 100), 100), 20)

class(name='VEROSSIMILHANÇA'
    factor (name='CREDIBILIDADE'
      filteredConstraint(
          filter (compareFilter ( C.NATION='PERU' )),
            attributeConstraint ( L.ORDERPRIORITY,
                compareCondition ( L.ORDERPRIORITY <> NULL,    +7 ),
                defaultValueFunction ( -7 ), 100), 100), 100), 80) )
```

**Figura 5.8** - Perfil C cadastrado pelo Usuário3

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( C.NATION = 'BRAZIL' )),
              attributeConstraint (C.MKTSEGMENT,
                  compareCondition ( C.MKTSEGMENT = 'AUTOMOBILE', +7 ),
                  compareCondition ( C.MKTSEGMENT ='HOUSEHOLD', +3 ),
                  defaultValueFunction ( +0 ), 100), 60),
      filteredConstraint(
          filter (compareFilter ( C.NATION = 'ARGENTINA' )),
              attributeConstraint (C.MKTSEGMENT ,
                  compareCondition ( C.MKTSEGMENT = 'HOUSEHOLD', +7 ),
                  compareCondition (  C.MKTSEGMENT ='AUTOMOBILE', +5 ),
                  defaultValueFunction ( +0 ), 100), 40), 100) , 30)

  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( MONTH (L.ORDERDATE)= MONTH (TODAY()) )),
            attributeConstraint ( L.ORDERDATE,
              compareCondition ( MONTH (L.ORDERDATE) = MONTH (L.COMMITDATE),  +7 ),
              defaultValueFunction ( -7 ), 100), 100), 100), 70) )
```

**Figura 5.9 -** Perfil D cadastrado pelo Usuário4

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( C.REGION = 'AMERICA' )),
              attributeConstraint (C.MKTSEGMENT,
                  compareCondition (C.MKTSEGMENT = 'BUILDING', +7 ),
                  compareCondition (C.MKTSEGMENT ='FURNITURE', +3 ),
                  defaultValueFunction ( +0 ), 100), 50),
      filteredConstraint(
          filter (compareFilter ( C.REGION = 'EUROPE' )),
              attributeConstraint (C.MKTSEGMENT,
                  compareCondition (C.MKTSEGMENT = 'MACHINERY', +7 ),
                  compareCondition (C.MKTSEGMENT ='BUILDING', +3 ),
                  defaultValueFunction ( +0 ), 100), 50), 100), 50)
  class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
        filter (compareFilter ( YEAR (L.ORDERDATE) = YEAR(TODAY()) )),
            attributeConstraint (L.ORDERDATE,
                compareCondition (MONTH(L.ORDERDATE) =  MONTH (TODAY(),    +6 ),
                compareCondition (MONTH(L.ORDERDATE) =  MONTH (TODAY()-1,  +3 ),
                compareCondition (MONTH(L.ORDERDATE) =  MONTH (TODAY()-2,  +0 ),
                defaultValueFunction ( -5 ), 100), 100), 100), 50) )
```

**Figura 5.10 -** Perfil E cadastrado pelo Usuário5

```
profile(
  class(name='INTERPRETABILIDADE'
    factor (name='ORIGEM'
      filteredConstraint(
          filter (compareFilter ( S.REGION='MIDDLE EAST' )),
            attributeConstraint ( L. ORDERPRIORITY,
                compareCondition ( L. ORDERPRIORITY <> '5-LOW',    +5 ),
                defaultValueFunction ( -5 ), 60)
            attributeConstraint ( L. SHIPMODE,
                compareCondition ( L. SHIPMODE <> 'RAIL',    +7 ),
                defaultValueFunction ( -7 ), 40), 100), 100), 70)

class(name='UTILIDADE'
    factor (name='TEMPORALIDADE'
      filteredConstraint(
          filter (compareFilter ( YEAR(L. ORDERDATE) = YEAR (TODAY() )),
```

```
attributeConstraint ( L.EXTENDEDPRICE,
    compareCondition ( L. EXTENDEDPRICE >= '10000',    +7 ),
    defaultValueFunction ( -3 ), 100), 100), 100), 30) )
```

**Figura 5.11** - Perfil F cadastrado pelo Usuário6

### 5.2.4. Especificação do Processo ETL

Utilizado para realizar a transformação do esquema TPC-H no esquema dimensional do DW, e para carregar os dados no ambiente transacional, foi utilizada a ferramenta VisualTPCH (Figura 5.12).

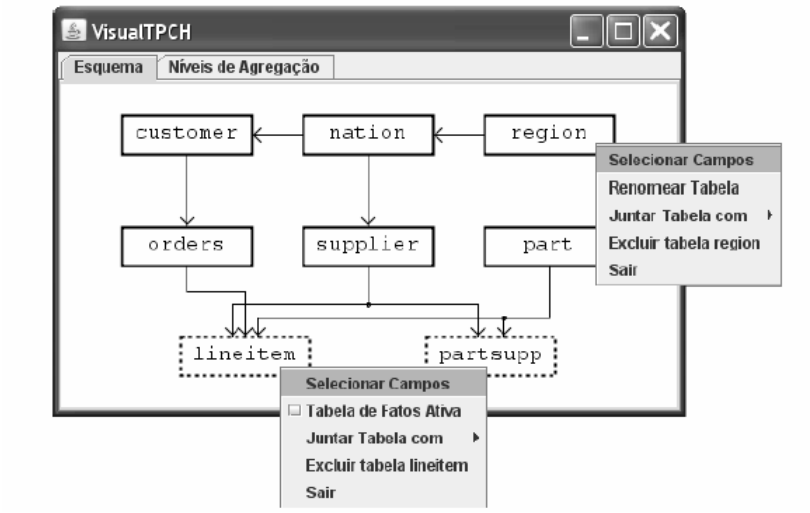


**Figura 5.12 –** Ferramenta VisualTPCH (DOMINGUES, CIFERRI e CIFERRI , 2008)

A ferramenta VisualTPCH, desenvolvida por DOMINGUES, CIFERRI e CIFERRI (2008), fornece uma interface gráfica que facilita a geração de dados sintéticos para ambientes de data warehouse com base no benchmark TPC-H. A ferramenta permite a manipulação do esquema TPC-H, a geração de níveis de agregação para os dados armazenados no DW e o armazenamento dos dados gerados em diferentes sistemas gerenciadores de banco de dados. A Figura 5.12 ilustra uma das principais funcionalidades da ferramenta: a manipulação do esquema TPC-H.

Na implementação da dimensão tempo (*DATE*) foi utilizada a ferramenta KETLLE, para a dimensão não é requerida fonte de dados.

KETTLE é um acrônimo para *Kettle Extraction, Transformation, Transportation and Loading Environment*. KETTLE é uma suíte ETL de código aberto, parte integrante do pacote de ferramentas *Pentaho Data Integration* - PDI (PENTAHO, 2009). A versão da ferramenta utilizada foi a PDI 3.1.0 GA.

A suíte é composta pelas seguintes ferramentas:

- ***Spoon*:** Ferramenta gráfica utilizada para modelar o fluxo dos dados e as transformações (*transformation*);
- ***Pan*:** Executa as transformações modeladas pelo *Spoon*;
- ***Chef*:** Utilizado para organizar as transformações em tarefas (*jobs*);
- ***Kitchen*:** Executa as tarefas criadas no *Chef*.

A implementação dos processos ETL na ferramenta envolve os conceitos de transformação e tarefa. Uma transformação é um mecanismo capaz de realizar leitura, escrita ou manipulação de dados para e a partir de diversas fontes de dados. Uma tarefa é a maneira de invocar transformações e controlar a seqüência de sua execução. A Figura 5.13 ilustra a criação de transformações para um processo de ETL da dimensão *SUPPLIER*.

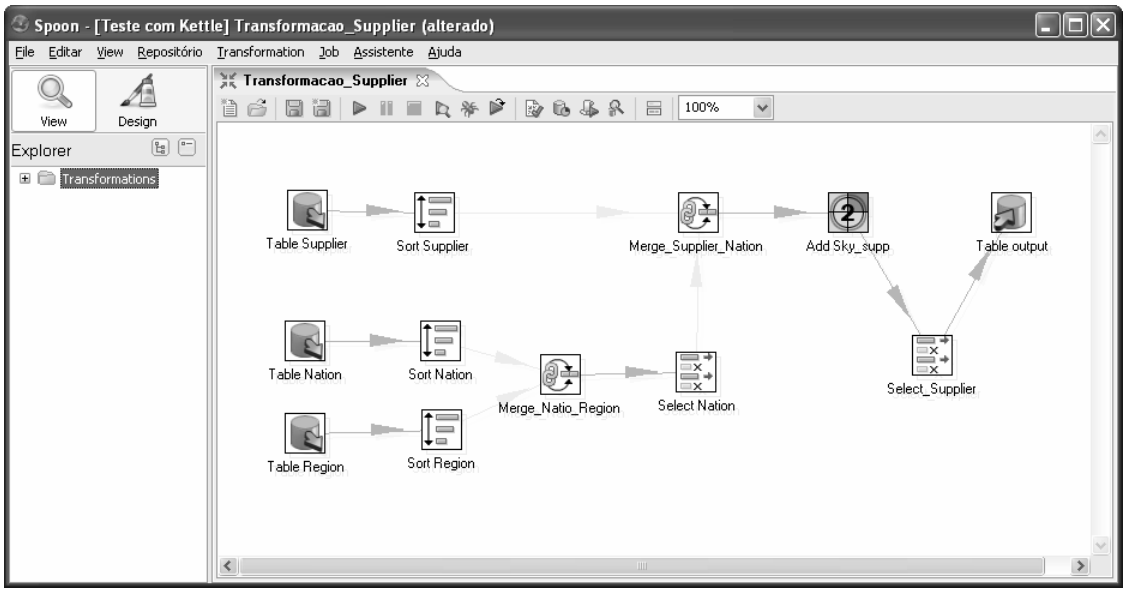


**Figura 5.13 -** Criação de Transformações no KETTLE

A carga das dimensões *SUPPLIER*, *CUSTOMER* e *PART*, a partir das tabelas existentes no esquema fonte de dados TPC-H, é realizada apenas uma vez, devido os dados neste ambiente de avaliação não sofrerem modificações e já termos o conteúdo completo das dimensões.

Observa-se na figura 5.13, como é especificada a criação da dimensão *SUPPLIER*, onde são informadas as tabelas envolvidas e a integração dos dados, sendo o resultado da transformação elaborada no KETTLE, expressa pelo Script SQL abaixo:

```
SELECT  s_suppkey, s_name, s_address, phone, s_acctbal,
        n_name as s_nation, r_name as s_region
   FROM supplier, nation, region
  WHERE s_nationkey = n_nationkey
    AND n_regionkey = r_regionkey
```

Para todas as dimensões foram geradas transformações, cujos scripts estão disponíveis no anexo IV.

## *5.3. Implementação do ambiente de avaliação*

Aqui são apresenta as expressões de extração montadas a partir dos perfis e apresenta como são realizadas as cargas no DW para o fato (*LINEORDER*).

Primeiramente são extraídos pelo Módulo de Expressões de Extração (MEE) os predicados de todos os perfis cadastrados no Módulo de Perfis, e agrupados por temporalidade, conforme apresentado na tabela 5.2.

**Tabela 5.2** – Predicados dos perfis agrupados por temporalidade

| | **TEMPORALIDADE** | | |
|---|---|---|---|
| **Perfil** | **Diária** | **Mensal** | **Anual** |
| A | S.NATION='RUSSIA' | | |
| B | C.NATION = 'BRAZIL'<br>C.NATION = 'ARGENTINA' | | |
| C | C.NATION='PERU' | | |
| D | | C.NATION = 'BRAZIL'<br>C.NATION = 'ARGENTINA' | |
| E | | C.REGION = 'AMERICA'<br>C.REGION = 'EUROPE' | |
| F | | | S.REGION='MIDDLE EAST' |

A partir dos predicados, o MEE gera as expressões de extração e expressões de extração complementares, que são adicionadas à cláusula `where` da consulta SQL que extrai os dados operacionais das fontes, apresentadas na tabela 5.3. No ambiente Brahma foram criadas visões (*VIEWS*) que conseguem refletir as necessidades de dados diárias, mensais e anuais, além da extração complementar final com os dados não requeridos por nenhum perfil (anexo V). As visões são utilizadas pelos procedimentos de carga que são executados de forma a simular a carga de dados diária no DW, e encontram-se no anexo VI.

**Tabela 5.3** – Expressões de Extração (EE) e Expressões de Extração Complementar (EEC)

| | EE | EEC |
|---|---|---|
| ***Diária*** | AND (s.s_nation = 'RUSSIA'<br>OR c.c_nation = 'ARGENTINA'<br>OR c.c_nation = 'BRAZIL'<br>OR c.c_nation = 'PERU') | AND NOT s.s_nation = 'RUSSIA'<br>AND NOT c.c_nation = 'ARGENTINA'<br>AND NOT c.c_nation = 'BRAZIL'<br>AND NOT c.c_nation = 'PERU' |
| ***Mensal*** | AND NOT s.s_nation = 'RUSSIA'<br>AND NOT c.c_nation = 'ARGENTINA'<br>AND NOT c.c_nation = 'BRAZIL'<br>AND NOT c.c_nation = 'PERU'<br>AND (c.c_region = 'EUROPE'<br>OR c.c_region = 'AMERICA') | AND NOT c.c_region = 'AMERICA' |
| ***Anual*** | AND NOT s.s_nation = 'RUSSIA'<br>AND NOT c.c_nation = 'ARGENTINA'<br>AND NOT c.c_nation = 'BRAZIL'<br>AND NOT c.c_nation = 'PERU'<br>AND NOT c.c_region = 'EUROPE'<br>AND NOT c.c_region = 'AMERICA'<br>AND s.s_region = 'MIDDLE EAST' | AND NOT s.s_region = 'MIDDLE EAST' |
| ***Final*** | AND NOT s.s_nation = 'RUSSIA'<br>AND NOT c.c_nation = 'ARGENTINA'<br>AND NOT c.c_nation = 'BRAZIL'<br>AND NOT c.c_nation = 'PERU'<br>AND NOT c.c_region = 'EUROPE'<br>AND NOT c.c_region = 'AMERICA'<br>AND NOT s.s_region = 'MIDDLE EAST' | |

# 6. Avaliação da Arquitetura BRAHMA

O presente capítulo é direcionado à execução dos testes de avaliação, executados sobre o ambiente simulado descrito no capítulo 5, para avaliação da arquitetura BRAHMA. Na primeira seção são relacionados os procedimentos executados e suas finalidades. A seção 6.2 apresenta as informações capturadas durante os eventos, e a análise detalhada das mesmas. Na última seção são apresentados os resultados do experimento no ambiente BRAHMA.

## 6.1. Procedimentos

A execução dos procedimentos para avaliação da arquitetura BRAHMA foram realizados a partir do ambiente construído no capítulo 5. As métricas avaliadas foram o tempo de execução dos processos de carga e latência de dados. A latência de dados refere-se a quão rapidamente os dados serão entregues para utilização pelos usuários finais, ou seja o quanto a minha informação está atualizada (HAINSTEN, 2001) (THO e TJOA, 2003) (LINTHICUM, 2009). Cada uma das duas métricas foi coletada em 2 cenários: o primeiro cenário (denominado “padrão”) simulou a execução das instâncias ETL em um ambiente tradicional, enquanto o segundo cenário (denominado “Brahma”) simulou a execução das instâncias ETL conforme especificado pela arquitetura Brahma.

Os resultados coletados para a métrica do tempo de execução foram registrados em uma tabela com os campos abaixo. Um trecho da tabela com os resultados coletados é ilustrado na figura 6.1. O registro completo dos resultados coletados está apresentado no anexo VII. Os resultados coletados também serviram de base para avaliação latência de dados. O número de registros trazidos em cada execução foi registrado para permitir validar a completude do DW ao final da execução de todas as cargas no ambiente Brahma, comparando com o numero de registros total existente no DW do ambiente padrão.

- *idlog* (número com seqüência de execução dos procedimentos);
- *atividade* (texto descritivo relativo ao procedimento em execução);
- *numlines* (número de registros acessados durante a execução do procedimento);
- *startdate* (data e hora do início da execução do procedimento); e

- *enddate* (data e hora do término da execução do procedimento), como ilustra a tabela 6.1.

**Tabela 6.1** – Tabela para armazenamento de quantidade de registros com tempo de execução dos procedimentos.

| idlog | Atividade | numlines | Startdate | enddate |
|---|---|---|---|---|
| *nn* | *aaaaaaaaaaaaaaaaaa* | *nnnnnn* | *yyyy-mm-dd hh:mm:ss.nnn* | *yyyy-mm-dd hh:mm:ss.nnn* |
| 1 | Carga padrão 01/1996 | 76859 | 2009-06-17 15:56:00.078 | 2009-06-17 15:56:03.453 |
| 2 | Carga padrão 02/1996 | 73197 | 2009-06-17 16:05:55.218 | 2009-06-17 16:06:00.421 |
| 3 | Carga padrão 03/1996 | 76283 | 2009-06-17 16:16:13.406 | 2009-06-17 16:16:19.109 |
| 4 | Carga padrão 04/1996 | 74941 | 2009-06-17 16:26:19.000 | 2009-06-17 16:26:25.781 |
| 5 | Carga padrão 05/1996 | 76445 | 2009-06-17 16:36:37.281 | 2009-06-17 16:36:44.218 |
| ... | ... | ... | ... | ... |
| 21 | Carga Brahma 01/1996 | 21183 | 2009-06-17 20:23:00.921 | 2009-06-17 20:23:02.281 |
| 22 | Carga Brahma 02/1996 | 19973 | 2009-06-17 20:23:02.312 | 2009-06-17 20:23:02.843 |
| 23 | Carga Brahma 03/1996 | 20617 | 2009-06-17 20:23:02.843 | 2009-06-17 20:23:03.390 |
| 24 | Carga Brahma 04/1996 | 20559 | 2009-06-17 20:23:03.390 | 2009-06-17 20:23:03.781 |
| 25 | Carga Brahma 05/1996 | 20665 | 2009-06-17 20:23:03.781 | 2009-06-17 20:23:05.437 |
| ... | ... | ... | ... | ... |

Os cenários “padrão” e “Brahma” foram simulados através da execução de scripts. No cenário padrão as extrações foram realizadas mensalmente e as cargas de dados são completas, ou seja, sem reduzir a quantidade de dados irrelevantes trazidos a cada carga. No cenário Brahma, as extrações utilizaram as temporalidades como definido nos perfis de qualidade de dados para indicar as periodicidades de extração, e seus filtros que indicaram como os dados foram carregados de modo incremental no DW. Desta forma, o ambiente Brahma simulou a execução diária, mensal e anual de instâncias do processo de extração de dados, implementando carga incremental dos dados no DW.

## *6.2. Análise dos Resultados*

Os arquivos de registro dos procedimentos executados, *Logs*, foram armazenados e encontram-se no anexo VII e anexo VIII deste trabalho.

### 6.2.1. Avaliação da Métrica Tempo de Execução

A tabela 6.2 apresenta um subconjunto dos resultados relativos ao tempo de execução que foram coletados nos 2 cenários. São comparadas as execuções dos processos de carga executadas ao final de cada mês. Como pode ser observado, as extrações

de dados do cenário padrão (tradicional) trouxeram um volume maior de registros em todas as execuções, enquanto as extrações no cenário Brahma conseguiram atingir o objetivo de reduzir o volume de dados irrelevantes trazidos para o DW, e conseqüentemente reduzindo o tempo de execução da extração em até 64%.

**Tabela 6.2** - Tabela de Comparação do Tempo de Consulta Tradicional x BRAHMA

| ANO:1996 | TRADICIONAL | | BRAHMA | | Redução Percentual de Tempo do Cenário BRAHMA |
|---|---|---|---|---|---|
| Mês | Nº. Registros | Tempo(s) | Nº. Registros | Tempo (s) | |
| Janeiro | 76859 | 634,875 | 21183 | 237,672 | -62,56% |
| Fevereiro | 73197 | 605,125 | 19973 | 230,063 | -61,98% |
| Março | 76283 | 623,844 | 20617 | 251,062 | -59,76% |
| Abril | 74941 | 629,453 | 20559 | 244,375 | -61,18% |
| Maio | 76445 | 639,5 | 20665 | 248,469 | -61,15% |
| Junho | 75727 | 608,844 | 20743 | 243,597 | -59,99% |
| Julho | 76868 | 615,516 | 20895 | 244,485 | -60,28% |
| Agosto | 79101 | 638,328 | 21192 | 226,312 | -64,55% |
| Setembro | 75401 | 608,578 | 20259 | 222,844 | -63,38% |
| Outubro | 77467 | 617,906 | 20352 | 221,656 | -64,13% |
| Novembro | 75472 | 619,047 | 20510 | 293,031 | -52,66% |
| Dezembro | 77728 | 618,562 | 21191 | 237,094 | -61,67% |

Os dados da tabela 6.2 são apresentados graficamente no gráfico 6.1. O tempo de consulta para extração utilizando a Extração Incremental é reduzido, sendo em média 61,11% menor que no cenário padrão. A redução média no tempo de execução alcançada no cenário Brahma é diretamente proporcional ao volume de dados irrelevante que foi eliminado.

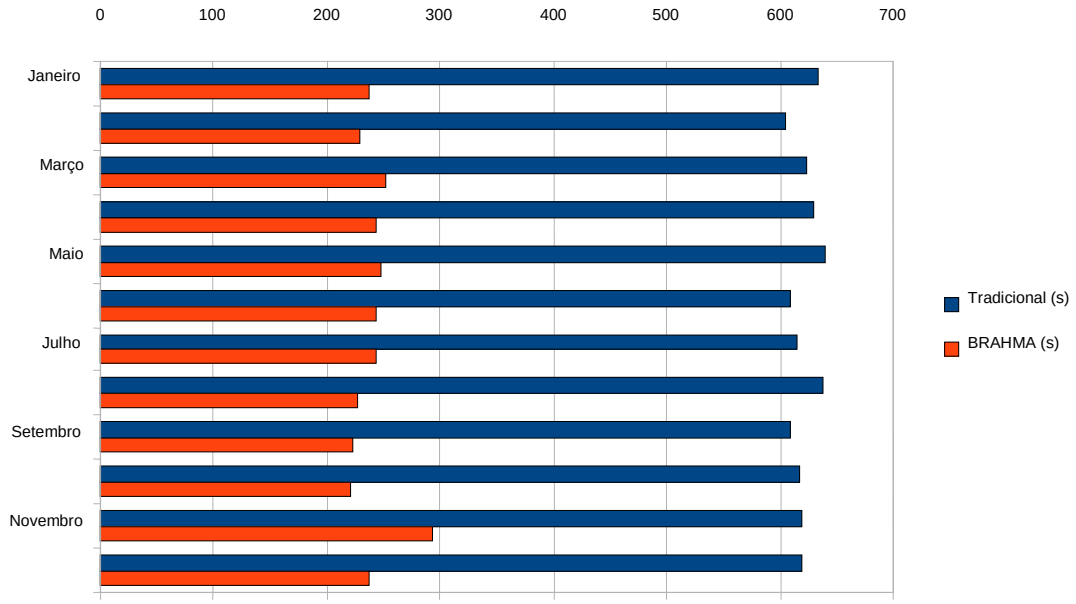


**Gráfico 6.1** – Gráfico Comparativo do Tempo de Consulta nos Cenário Tradicional e BRAHMA.

Para efeitos ilustrativos, também foram coletados os tempos de execução da etapa de carga (INSERT) no DW, apresentados na tabela 6.3, onde pode ser constatado que o ganho de desempenho obtido na etapa de carga supera, na maioria dos casos, o ganho obtido na etapa de extração, também em função da redução do volume de dados irrelevantes em cada instância executada.

**Tabela 6.3** - Tabela de Comparação do Tempo de Carga Tradicional x BRAHMA

| ANO:1996 | TRADICIONAL | | BRAHMA | | Redução Percentual Conseguida com BRAHMA |
|---|---|---|---|---|---|
| Mês | Nº. Registros | Tempo(s) | Nº. Registros | Tempo (s) | |
| Janeiro | 76859 | 3,375 | 21183 | 1,360 | -59,70% |
| Fevereiro | 73197 | 5,203 | 19973 | 0,531 | -89,79% |
| Março | 76283 | 5,703 | 20617 | 0,547 | -90,41% |
| Abril | 74941 | 6,781 | 20559 | 0,391 | -94,23% |
| Maio | 76445 | 6,937 | 20665 | 1,656 | -76,13% |
| Junho | 75727 | 8,328 | 20743 | 0,500 | -94,00% |
| Julho | 76868 | 5,750 | 20895 | 4,094 | -28,80% |
| Agosto | 79101 | 5,766 | 21192 | 1,594 | -72,36% |
| Setembro | 75401 | 5,953 | 20259 | 0,922 | -84,51% |
| Outubro | 77467 | 6,391 | 20352 | 3,766 | -41,07% |
| Novembro | 75472 | 4,563 | 20510 | 2,203 | -51,72% |
| Dezembro | 77728 | 8,454 | 21191 | 1,140 | -86,52% |

Conforme pode ser confirmado no gráfico 6.2, o tempo de carga que utiliza a Extração Incremental tem uma redução que é em média de 72,44%.

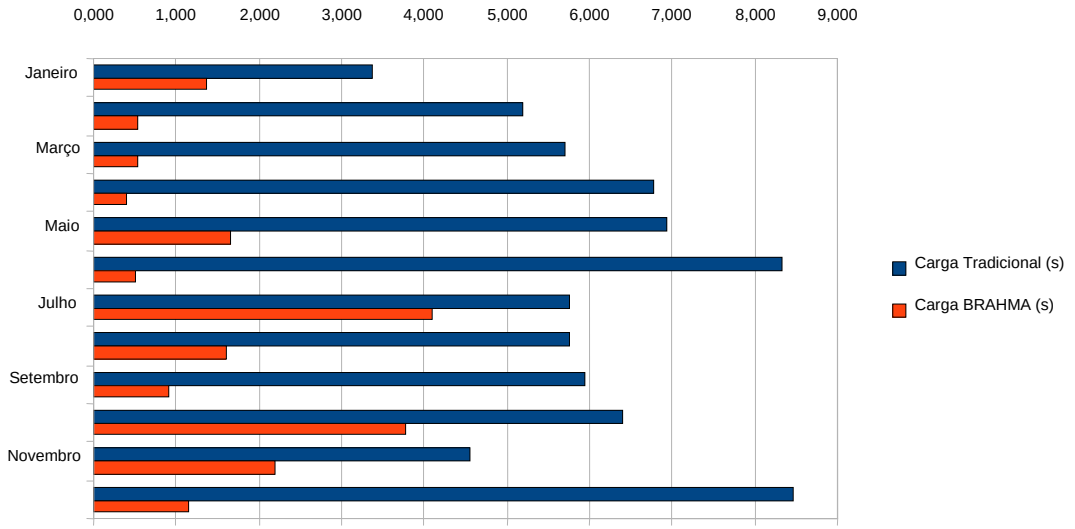


**Gráfico 6.2** – Gráfico Comparativo do Tempo de Carga nos Cenários Tradicional e BRAHMA.

Pelas análises comparativas fica evidente que a redução é proporcional à quantidade de dados irrelevantes que a arquitetura BRAHMA reduz, que é 72,9% menor que o total de dados carregado no DW pelo processo de ETL tradicional.

### 6.2.2. Análise da Latência de Dados

A avaliação da latência de dados do DW foi realizada através de procedimentos criados no PostgreSQL, (Anexo IX). Primeiramente, para cada um dos perfis diários definidos no capítulo 5 foram armazenadas a data de cadastro dos dados no banco de dados e a data em que os dados foram acessados no ambiente informacional, calculada a taxa de latência de dados, como sendo a diferença das datas dividido pelo total máximo de dias que o dado pode esperar para ser acessado.

As taxas de latência para os perfis de usuários diários nos ambientes padrão (tradicional) e Brahma, estão apresentados na tabela 6.4.

**Tabela 6.4** - Tabela de Taxa de Latência de Dados Tradicional x BRAHMA em dezembro/1996

| **Período dez/1996** | **Latência de dados no Ambiente TRADICIONAL** | | | **Latência de Dados no Ambiente BRAHMA** | | |
|---|---|---|---|---|---|---|
| **DIA** | **Perfil A** | **Perfil B** | **Perfil C** | **Perfil A** | **Perfil B** | **Perfil C** |
| 01 | 1,00 | 1,00 | 1,00 | 0,03 | 0,03 | 0,03 |
| 02 | 0,97 | 0,97 | 0,97 | 0,03 | 0,03 | 0,03 |
| 03 | 0,94 | 0,94 | 0,94 | 0,03 | 0,03 | 0,03 |
| 04 | 0,90 | 0,90 | 0,90 | 0,03 | 0,03 | 0,03 |
| 05 | 0,87 | 0,87 | 0,87 | 0,03 | 0,03 | 0,03 |
| 06 | 0,84 | 0,84 | 0,84 | 0,03 | 0,03 | 0,03 |
| 07 | 0,81 | 0,81 | 0,81 | 0,03 | 0,03 | 0,03 |
| 08 | 0,77 | 0,77 | 0,77 | 0,03 | 0,03 | 0,03 |
| 09 | 0,74 | 0,74 | 0,74 | 0,03 | 0,03 | 0,03 |
| 10 | 0,71 | 0,71 | 0,71 | 0,03 | 0,03 | 0,03 |
| 11 | 0,68 | 0,68 | 0,68 | 0,03 | 0,03 | 0,03 |
| 12 | 0,65 | 0,65 | 0,65 | 0,03 | 0,03 | 0,03 |
| 13 | 0,61 | 0,61 | 0,61 | 0,03 | 0,03 | 0,03 |
| 14 | 0,58 | 0,58 | 0,58 | 0,03 | 0,03 | 0,03 |
| 15 | 0,55 | 0,55 | 0,55 | 0,03 | 0,03 | 0,03 |
| 16 | 0,52 | 0,52 | 0,52 | 0,03 | 0,03 | 0,03 |
| 17 | 0,48 | 0,48 | 0,48 | 0,03 | 0,03 | 0,03 |
| 18 | 0,45 | 0,45 | 0,45 | 0,03 | 0,03 | 0,03 |
| 19 | 0,42 | 0,42 | 0,42 | 0,03 | 0,03 | 0,03 |
| 20 | 0,39 | 0,39 | 0,39 | 0,03 | 0,03 | 0,03 |
| 21 | 0,35 | 0,35 | 0,35 | 0,03 | 0,03 | 0,03 |

| | | | | | | |
|---|---|---|---|---|---|---|
| 22 | 0,32 | 0,32 | 0,32 | 0,03 | 0,03 | 0,03 |
| 23 | 0,29 | 0,29 | 0,29 | 0,03 | 0,03 | 0,03 |
| 24 | 0,26 | 0,26 | 0,26 | 0,03 | 0,03 | 0,03 |
| 25 | 0,23 | 0,23 | 0,23 | 0,03 | 0,03 | 0,03 |
| 26 | 0,19 | 0,19 | 0,19 | 0,03 | 0,03 | 0,03 |
| 27 | 0,16 | 0,16 | 0,16 | 0,03 | 0,03 | 0,03 |
| 28 | 0,13 | 0,13 | 0,13 | 0,03 | 0,03 | 0,03 |
| 29 | 0,10 | 0,10 | 0,10 | 0,03 | 0,03 | 0,03 |
| 30 | 0,06 | 0,06 | 0,06 | 0,03 | 0,03 | 0,03 |
| 31 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 | 0,03 |

A média da taxa de latência para os dados carregados mensalmente (Ambiente Tradicional) é de 0,52, ou seja os dados somente ficam disponíveis aos usuários após em média 15 dias aproximadamente após seu cadastro. Enquanto que para no ambiente BRAHMA a média é constante, a taxa é 0,03, todos os dados necessários, ao grupo de usuários de perfil diário, ficam disponíveis aos seus usuários após 1 dia do seu cadastro.

Os perfis mensais executados no ambiente padrão e Brahma obtiveram resultados também melhores, visto que parte dos dados já tinha sido atualizada, os resultados são apresentados na tabela 6.5.

**Tabela 6.5** - Tabela de Taxa de Latência de Dados Tradicional x BRAHMA

| **ANO 1996** | **Latência de dados no Ambiente TRADICIONAL** | | **Latência de dados no Ambiente BRAHMA** | |
|---|---|---|---|---|
| **MÊS** | **Perfil D** | **Perfil E** | **Perfil D** | **Perfil E** |
| Janeiro | 1,00 | 1,00 | 0,08 | 0,08 |
| Fevereiro | 0,92 | 0,92 | 0,08 | 0,08 |
| Março | 0,83 | 0,83 | 0,08 | 0,08 |
| Abril | 0,75 | 0,75 | 0,08 | 0,08 |
| Maio | 0,67 | 0,67 | 0,08 | 0,08 |
| Junho | 0,58 | 0,58 | 0,08 | 0,08 |
| Julho | 0,50 | 0,50 | 0,08 | 0,08 |
| Agosto | 0,42 | 0,42 | 0,08 | 0,08 |
| Setembro | 0,33 | 0,33 | 0,08 | 0,08 |
| Outubro | 0,25 | 0,25 | 0,08 | 0,08 |
| Novembro | 0,17 | 0,17 | 0,08 | 0,08 |
| Dezembro | 0,08 | 0,08 | 0,08 | 0,08 |

A latência ideal para um ambiente de tempo real é que todos os dados tenham uma latência o mais próxima de 0 (zero), como pode ser visualizado no gráfico 6.3, a medida que os dias passam e os dados são cadastrados cada vez mais próximos da execução do processo ETL, a latência é reduzida. No gráfico 6.4, temos evidenciada a mesma redução para os perfis mensais.

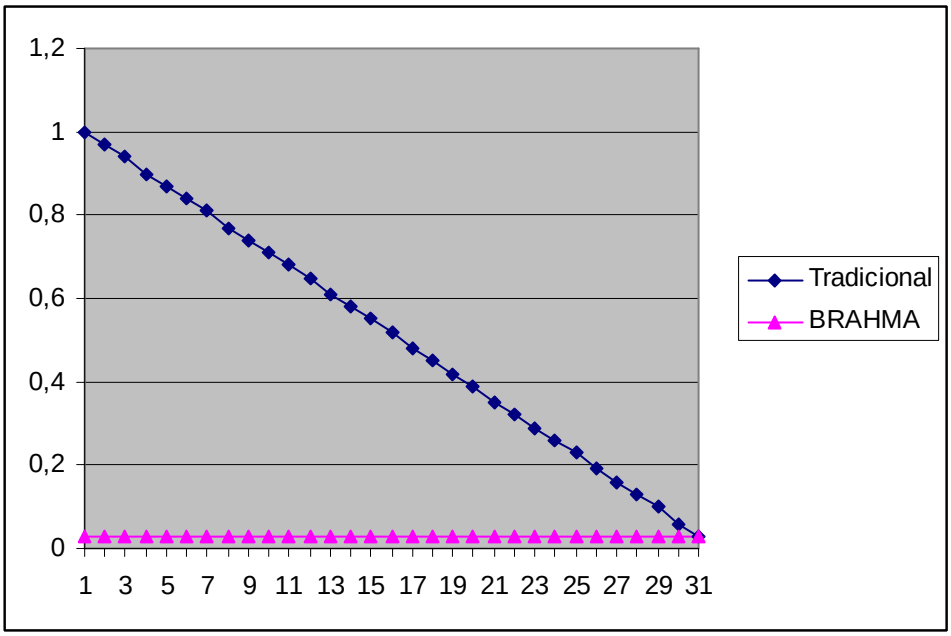


**Gráfico 6.3** – Gráfico Comparativo da Taxa de Latência nos Cenários Tradicional e BRAHMA para Perfis Diários.

A redução da latência tem impacto na atualidade dos dados no DW (fator de qualidade do DWQ) melhorando um fato de qualidade está melhhorando a qualidade de dados do DW de modo geral.

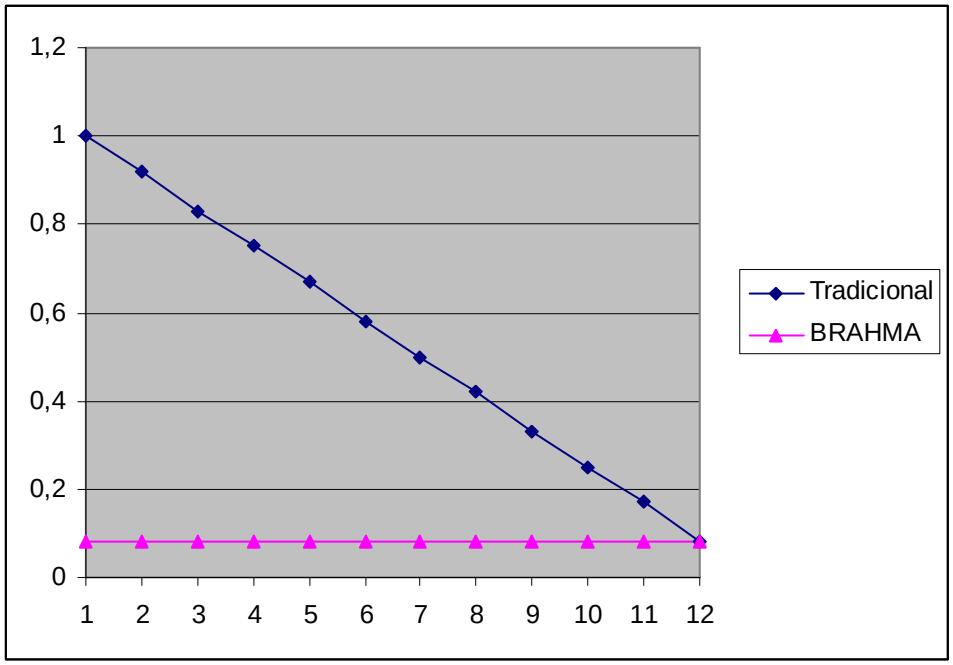


**Gráfico 6.4** – Gráfico Comparativo da Taxa de Latência nos Cenários Tradicional e BRAHMA para Perfis Mensais.

Com a utilização do ambiente Brahma é possível que usuários de todos os níveis de temporalidade tenham os dados disponíveis com uma antecedência em média 88% maior que para tomada de decisão. do que o usuário de um ambiente “padrão” (tradicional) que fica sujeito apenas à temporalidade de carga única.

# *7. Conclusão*

Na atualidade cada vez mais e mais decisões estratégicas são tomadas sobre as informações consolidadas a partir do DW, e neste ambiente é interessante que o nível de qualidade de dados seja conhecido durante a tomada de decisão. Além disso, ter informações disponíveis o mais rápido possível é buscado para dar agilidade ao processo de tomada de decisão, reduzindo o tempo entre a ocorrência de um evento no ambiente transacional e o momento quando uma decisão é tomada no ambiente informacional (BI 2.0).

Ainda, em ambientes de BI existem diferentes usuários com visões particulares sobre qualidade de dados, pode ser um usuário que necessita de informações mais detalhadas ou menos detalhadas, ou ainda, seu objetivo pode ser para fins de tomada de decisão ou para fins de auditoria. Para cada perfil de usuário, diferentes porções de dados são requeridas no DW. No entanto, tipicamente, estes ambientes não levam em conta estes diferentes perfis e oferecem um único processo ETL para a atualização dos dados do DW. Para tratar o problema de personalização em tais ambientes, no sentido de viabilizar a co-existência de diferentes perfis de usuário, minimizando a perda da qualidade. Em particular, o presente trabalho busca a redução da temporalidade requerida por ambientes de BI tempo real através da redução da latência entre a transação ocorrida na fonte operacional e sua utilização no DW tornando a tomada de decisões mais rápida.

Desta forma, o objetivo deste trabalho foi que a redução do volume de dados em cada execução do processo de ETL melhora o seu desempenho, possibilitando atender os diferentes requisitos de temporalidade dos usuários em ambientes de BI tempo real. Para atingir este objetivo, foi proposta a arquitetura BRAHMA.

A arquitetura BRAHMA reduziu o volume de dados irrelevantes tratados a cada execução do processo de ETL. Esta redução diminui o tempo de execução do processo de ETL (possibilitando atender os diferentes requisitos de temporalidade definidos pelos usuários em seus perfis) e, conseqüentemente, aumentou a atualidade dos dados do DW em ambientes de BI tempo real.

Neste capítulo da conclusão são apresentadas as considerações finais sobre o trabalho de dissertação, suas contribuições, as limitações da proposta e perspectivas futuras.

## *7.1. Contribuições*

As principais contribuições deste trabalho incluem: a especificação de uma arquitetura para a extração personalizada e incremental de dados em ambientes de BI tempo real; a implementação dos principais módulos da arquitetura; a especificação do cenário de experimentação da arquitetura proposta, que incluiu tanto a instanciação de um meta-modelo de qualidade de dados da literatura com critérios específicos para ambientes de BI sobre DWs quanto a definição de um processo ETL sobre a especificação original do benchmark TPC-H para transformá-lo em um esquema estrela; os resultados obtidos com a execução da avaliação, demonstrando a efetividade da proposta ao reduzir em 66,77% na média o tempo de execução da extração de dados, e a redução da latência de dados que possibilita aos usuários terem em um tempo menor os dados necessários a tomada de decisão em mãos.

Além das contribuições citadas, outras contribuições específicas podem ser relacionadas, como:

- A implementação de um sistema de cadastro de perfis de qualidade de dados, que facilita a entrada de dados para avaliar a qualidade de dados em um ambiente de DW, numa interface mais amigável;
- A adaptação do benchmark TPC-H (2008), como proposto por CHEN (2008), transformando-o em um modelo informacional mais adequado ao ambiente de BI e ao Data Warehouse para realização dos testes e avaliação da arquitetura BRAHMA;
- A comprovação que o meta-modelo de qualidade de dados de Simmhan (2007), pode ser adequado ao domínio de DWQ (JARKE e VASSILIOU, 2001), e que possibilitou pontuar a qualidade de dados do DW no ambiente BRAHMA;
- Os usuários com perfis de granularidade menor que podem ter a qualidade dos dados aferida, enquanto num ambiente tradicional não é possível calcular a qualidade dos dados para temporalidade inferiores da definida como padrão. E ainda, é possível realizar verificações parciais do *score* de qualidade de dados

do DW para usuários com perfis com requisitos atendidos por perifs de temporalidade inferiores.

## *7.2. Limitações da Proposta*

O ambiente BRAHMA para funcionar necessita que alguns requisitos do cenário sejam atendidos na definição dos perfis:

- Todos os perfis devem incluir o fator de qualidade de dados *'Temporalidade'*, sem esta definição não é possível identificar o nível de temporalidade do perfil, e realizar o agrupamento dos perfis;
- Os subconjuntos de dados que os usuários requerem devem ter granularidade da mais fina até a mais grossa. O ganho será maior quanto mais fina a granularidade dos subconjuntos. Se todos os perfis definirem a mesma temporalidade, não haverá ganho no desempenho dos processos de extração. Por outro lado, neste caso o desempenho do ETL na arquitetura BRAHMA vai coincidir com o do ambiente de DW tradicional;
- O cadastro de perfis requer que o usuário tenha conhecimento sobre o modelo de qualidade adotado, para que ele consiga expressar as suas necessidades de qualidade de dados de forma correta e adequada. Alternativamente, em ambientes que adotem a arquitetura BRAHMA proposta, pode-se requerer a existência de pessoas especificamente treinadas para assessorar os usuários a cadastrarem seus perfis de qualidade.

## *7.3. Trabalhos Futuros*

Ao longo da evolução da dissertação, várias direções foram identificadas para trabalhos futuros, realizando a evolução do ambiente BRAHMA, descritas a seguir.

A aplicação do ambiente BRAHMA em uma situação real, para através desse estudo de caso realizar a análise dos impactos de sua utilização para os usuários e para os negócios;

A execução de ETL com paralelismo inter-processo, conforme proposto no trabalho de Martins *et al.*(2008). No paralelismo inter-processo, diversas instâncias de processos ETL são executadas em paralelo. Cada instância de processo ETL seria independente

das demais, e acessaria um conjunto distinto de dados, potencialmente contribuindo ainda mais para o aumento de desempenho do ETL e redução da latência de dados.

O agrupamento de perfis em grupos no módulo de Clustering pode passar a considerar diversos fatores de qualidade, e não só a temporalidade. Neste caso formariam-se grupos de usuários que têm interesse em subconjuntos distintos de dados para extração, levando-se em conta outros critérios que não o nível de temporalidade. Isso poderia contribuir para definir expressões de extração mais complexas, fazendo com que cada grupo de usuário acessasse um subconjunto reduzido de dados, voltado para atender os critérios de qualidade especificados pelos usuários daquele grupo.

Existem na literatura trabalhos que utilizam paralelismo e distribuição para melhorar o desempenho de consultas sobre DW. O trabalho de Furtado (2005) propôs o uso de técnicas de paralelismo físico e virtual para obter alto desempenho em agrupamentos de banco de dados, enquanto Mattoso (2005) propôs um middleware (Pargres) para possibilitar o processamento paralelo de consultas OLAP sobre um DW em um cluster de banco de dados. A arquitetura proposta pode ser integrada ao cluster de PCs, contribuindo para o aumento do desempenho não só das consultas OLAP, mas também do processo de ETL no ambiente de BI.

A ampliação dos fatores de qualidade de dados disponíveis para especificação dos perfis. Além dos fatores de qualidade de dados carregados no DW, podem-se utilizar os metadados obtidos durante o processo de ETL para maior detalhamento dos perfis, ampliando a visão de qualidade de dados do usuário sobre os negócios.

A arquitetura proposta pode ser aplicada em outros contextos além de BI tempo real, onde a existência de processos definidos e a busca pelo aumento da qualidade de dados sejam relevantes para buscar maior transparência de quais foram os dados gerados durante a execução de um processo, e como eles foram gerados. Para isso, os critérios de qualidade devem ser estendidos para compreender também a qualidade do processo de geração dos dados.

Espera-se que o presente trabalho de dissertação sirva de inspiração para o desenvolvimento acadêmico, auxilie na ampliação dos conhecimentos na área de sistemas de informação e na constante busca pelo saber.

## *Referências*


AMARAL, Glenda Carla Moura. *AQUAWARE: Um Ambiente de Suporte à Qualidade de Dados em Data Warehouse*. Dissertação de Mestrado.(UFRJ), Rio de Janeiro, 2003.

AMARAL, G. C. M.; CAMPOS, M. L. M. *Modelando Metadados de Qualidade no Data Warehouse*. IV Simpósio de Desenvolvimento e Manutenção de Software da Marinha, 2004.

AQUINO, L. G. B.; JUNIOR, R. O. C. *Brahma: Um Ambiente para Avaliação da Qualidade dos Dados em Data Warehouses em Apoio à Business Intelligence.* 2009. Projeto Final – Escola de Informática Aplicada, Universidade Federal do Estado do Rio de Janeiro (UNIRIO), Rio de Janeiro, RJ, Brasil. Orientadores: Fernanda Araújo Baião e Daniel Barbosa Martins.

BARBIERI, Carlos. *BI – Business Intelligence, Modelagem e Tecnologia*. Axcel Books, 2001.

CHEN, X.; O'NEIL, P.; O'NEIL, E. *Adjoined Dimension Column Clustering to Improve Data Warehouse Query Performance*, ICDE08, Cancun, 2008.

COSTA, A. F., ANCIÃES, F. C.; *Aspectos de Criação e Carga de um Ambiente de Data Warehouse.* Trabalho de Conclusão de Curso (Bacharelado em Informática), Universidade Federal do Rio de Janeiro, março, 2001.

DAYAL, U.; CASTELLANOS, M; SIMITSIS, A.; WILKINSON, K.: *Data integration flows for business intelligence*. EDBT 2009: p.1-11

DELPHI. *Linguagem de Programação DELPHI*. Disponível em: http:www.borland.com/br/products/delphi/index.html. Acesso em: Janeiro, 2009.

DOMINGUES, G. R.; CIFERRI, C. D. A.; CIFERRI, R. R. *VisualTPCH: Uma Ferramenta para a Geração de Dados Sintéticos para Data Warehouse*. In: Anais da Sessão de Demos do XXIII Simpósio Brasileiro de Banco de Dados, p.31-36, 2008.

ECKERSON, W. W. *Data Quality and The Botttom Line: Achieving Business Success through a Commitment to High Quality Data*. The Data Warehousing Institute Report Series, USA. Disponível em: http://www.stuart.iit.edu/courses/im510/spring2002/dqreport.pdf. Acesso em: 24 de junho de 2007.

ELMASRI, R. E.; NAVATHE, S. *Sistemas de banco de dados*. 4ª ed. São Paulo: Addison Wesley, 2005.

ERICSON, J.. *Near Real-Time Data Warehouse at Work*. Information Management Special Reports, Fevereiro, 2009. Disponível: http://www.information-management.com/specialreports/2009_125/10014931-1.html

FURTADO, C.; LIMA, A.A.B.; PACITTI, E.; VALDURIEZ, P.; MATTOSO, M.. *Physical and virtual partitioning in OLAP database clusters. Computer Architecture and High Performance Computing*, 2005. SBAC-PAD 2005. 17th International Symposium on Volume , Issue , 24-27 Oct. 2005 Page(s): 143. - 150Digital Object Identifier 10.1109/CAHPC.2005.32.

HAINSTEN, M. *The Real-Time Data Warehousing Defined.* White Paper, Daman Consulting, 2001.

IBM. *Evaluating Real-Time Data Integration Solutions*. White Paper, Abril,2008.

INMON, William H. *Como construir o data warehouse*. 2ª Edição, Editora Campus, 1997, Rio de Janeiro.

JARKE, M.; VASSILIOU, Y. *Data Warehouse Quality: A Review of the DWQ Project*. In: Proceedings of the 1997 Conference on Information Quality. Cambridge, MA, p. 229-313, 1997.

JARKE, LENZERINI, VASSILIOU, VASSILIADIS, *Fundamentals of Data Warehouses*, 2nd ed., Springer, 2001.

JEUSFELD, M. A.; QUIX, C.; JARKE, M. Design and Analysis of Quality Information for Data Warehouses, 1998.

JÖRG, T.; DESSLOCH, S. Formalizing ETL Jobs for Incremental Loading of Data Warehouses. In Proceedings der 13 GI-Fachtagung für Datenbanksysteme in Business, Technologie und Web (BTW 2009), Lecture Notes in Informatics, Volume P-144, 2. bis 6. März 2009, Münster, Germany. Pp.327-346

JÖRG, T.; DESSLOCH, S. *Near Real-Time Data Warehousing Using State-of-the-Art ETL Tools. In Workshop on Enabling Real-Time for Business Intelligence* (BIRTE 2009), August 2009, Lyon, France.

KIMBALL, R; ROSS, M. *The Data Warehouse Toolkit. Guia Completo para Modelagem Dimensional*. Tradução da 2. ed. Rio de Janeiro: Campus, 2002.

KIMBALL, Ralph; MERZ, Richard. *Data Webhouse: Construindo o Data Warehouse para a WEB.* Ed. Campus, Rio de Janeiro, 2000.

LEE, Y. W.; STRONG, D. M.; KAHN, B. K.; WANG, R. Y.. *AIMQ: A Methodology for Information Quality Assessment, Information & Management*, v.40, n.2, 2001, pp.133-146. Disponível em: http://web.mit.edu/tdqm/www/tdqmpub/AIMQJun02.pdf. Acesso em: 26 de junho de 2007.

LINTHICUM, David. Leveraging Information and Intelligence “Three Categories of Data Latency”. ebizQ The Insider’s Guide to Bussines and It Agility , October, 2009.

LOUIS, J.S..Implementing Real-Time Data Integration Solutions. In TDWI. What Works: Volume 25, May 2008.

MACHADO, Felipe N. R. Projeto de Data Warehouse: uma visão multidimensional. São Paulo: Érica, 2000.

MARTINS, D.; BAIÃO, F.; CAVALCANTI, M. C. *Aplicação de Técnicas de Distribuição e Paralelismo em Ambientes de BI 2.0 para suporte à Qualidade de Dados*. Workshop de Teses e Dissertações em Bancos de Dados - Simpósio Brasileiro de Bancos de Dados, Campinas, 2008.

MATTOSO, M.; ZIMBRRÃO G.; LIMA, A.A.B.; BAIÃO, F.; BRAGANHOLO, V.P.; AVELEDA,A.A.; MIRANDA, B.;ALMENTERO, B.K.; COSTA, M.N..*ParGRES:*

*Middleware para Processamento Paralelo de Consultas OLAP em Clusters de Banco de Dados*. In: Sessão de Demonstrações do SBBD, 2005, Uberlândia, MG.

NOVABASE. *Qualidade de Dados On-line*. Abril, 2003. Disponível em: http://www.novabase.pt/ConteudosHTML/MFDQOnline.pdf. Acesso em: 21/03/2008.

OLIVEIRA, W, J. *Data Warehouse*. 2. ed. Florianopolis: Visual Books, 2002.

ÖZSU, M. Tamer e VALDURIEZ, Patrick. *Princípios de Banco de Dados Distribuídos*. 2ª Edição, Editora Campus, 2001, Rio de Janeiro.

PENTAHO. *Suite de ferramentas ETL Pentaho Data Integration (PDI).* Disponível: http://kettle.pentaho.org. Acesso em: Janeiro, 2009.

POSTGRESQL. *Banco de dados PostgreSQL*. Disponível: http://www.postgresql.org.br. Acesso em: Janeiro, 2009.

REZENDE, D. A*.. Engenharia de Software e Sistemas de Informações*. Rio de Janeiro, Ed. Brasport, 1999.

SAQID, S W. ZHOU, X.,DENG,K.. Research *and Practice in Data Quality*. APWeb 2008: 41-42. 2008.

SAVLA, S. e NINAN,M. *MANAGING DATA QUALITY. SetlABS Briefiengs*, v.3, n.4. Out/Dez 2005. (p.57 à p.64).

STODDER, David. *The State of Business Intelligence.* Artigo publicado em 30/04/2008, no site da Intelligent Enterprise. Disponível: http://www.intelligententerprise.com/showArticle.jhtml?articleID=198701576&pgno=1.

STRONG, D.M; LEE, Y.W.; WANG, R.Y. *Data Quality in Context, Communications of The ACM*, v.40, n.5, 1997. Disponível em: http://web.mit.edu/tdqm/www/tdqmpub/StrongLeeWangCACMMay97.pdf. Acesso em: 26 de junho de 2007.

SIMMHAN, YOGESH L. *Provenance Framework in Support of Data Quality Estimation*. Tese de Doutorado, Indiana University, dezembro, 2007.

THO, NGUYEN M.; TJOA, A MIN. *Zero-Latency Data Warehousing for Heterogeneous Data Sources and Continuous Data Streams* Presented at iiWAS 2003, Fifth International Conference on Information and Web-based Applications and Services, Jakarta, Indonesia, September 15-17, 2003.

TPC, Transaction Processing Performance Council. *TPC BENCHMARK H (Decision Support), Standard Specification, versão 2.8.0.* Disponível em http://www.tpc.org/tpch. Acesso em: nov. 2008.

VASSILIADIS, P. *Data Warehouse Modeling and Quality Issues*. Tese de Doutorado, National Technical University of Athens, junho, 2000.

VASSILIADIS P., KARAGIANNIS A.,TZIOVARA V., SIMITRIS *A..Towards a Benchmark for ETL worflows*. In QDB'07, 2007.

WANG, R. Y; STRONG, D. M. Beyond Accuracy: *What Data Quality means to Data Consumers* – Journal of management information Systems, v.12, n.4, spring 1996Y.

ZHANG,Y. et al. (Eds.): *APWeb* 2008, LNCS 4976, pp. 41–42, 2008.

ZIEGLER, P.,DITTRITCH, K., *Data Integration — Problems, Approaches, and Perspectives*. In John Krogstie, Andreas L. Opdahl, and Sjaak Brinkkemper, editors, Conceptual Modelling in Information Systems Engineering, papes 39–58. Springer, Berlin, 2007.

ZIEGLER, P.,DITTRITCH, K., HUNT, E. A Call for Personal Semantic Data Integration. In Workshop on Information Integration Methods, Architectures, and Systems (IIMAS 2008) (in conjunction with ICDE 2008), Cancun, Mexico, 2008

## *ANEXO I – Script de Criação de Tabelas do Módulo de Perfis*

***{cria_base_perfis.sql}***

```
-- APAGA TABELAS

drop table APPATRIBUTOSDW;
drop table APPFATORES;
drop table APPCLASSES;
drop table APPSINAISCOMPARACAO;
drop table APPTIPOSDADOS;
drop table CONDICOES;
drop table RESTRICOESATRIBUTO;
drop table FILTROS;
drop table RESTRICOES;
drop table FATORESQUALIDADE;
drop table CLASSESFQ;
drop table PERFIS;
drop table USUARIOS;


-- CRIAÇÃO de TABELAS DO MÓDULO de PERFIS

-- TABELAS APOIO: Guardam opções p/ seleção no ambiente gráfico.

-- Table: APPATRIBUTOSDW
create table APPATRIBUTOSDW
(ID          SERIAL   not null,
 NOME        CHAR(30),
 DESCRICAO   CHAR(50),
 TIPO        CHAR(50),
 constraint  PK_APPATRIBUTOSDW primary key (ID));

-- Table: APPCLASSES
create table APPCLASSES
(ID_CLASSES  SERIAL   not null,
NOME         CHAR(50),
constraint   PK_APPCLASSES primary key (ID_CLASSES));

-- Table: APPFATORES
create table APPFATORES
(ID_FATORES    SERIAL   not null,
 ID_CLASSES    INT4     not null,
 NOME          CHAR(50),
 constraint    PK_APPFATORES primary key (ID_FATORES, ID_CLASSES));

-- Table: APPSINAISCOMPARACAO
create table APPSINAISCOMPARACAO
(ID          SERIAL   not null,
 SINAL       CHAR(5),
 constraint  PK_APPSINAISCOMPARACAO primary key (ID));

-- Table: APPTIPOSDADOS
create table APPTIPOSDADOS
(ID         SERIAL   not null,
 TIPO       CHAR(15),
 constraint PK_APPTIPOSDADOS primary key (ID));
```

**-- TABELAS DO USUÁRIO: Guardam dados relativos ao perfil cadastrado.**

```
-- Table: CLASSESFQ
create table CLASSESFQ
(ID          SERIAL   not null,
 ID_PERFIL   INT4     not null,
 NOME        CHAR(20),
 PESO        NUMERIC(3),
 constraint  PK_CLASSESFQ primary key (ID));

-- Table: CONDICOES
create table CONDICOES
(ID                    SERIAL   not null,
 ID_RESTRICOESATRIBUTO INT4     not null,
 TIPO_CONDICAO         CHAR(10),
 ATRIBUTO              CHAR(20),
 SINAL                 CHAR(4),
 VALOR1                CHAR(100),
 VALOR2                CHAR(100),
 PONTOS                NUMERIC(1),
 constraint            PK_CONDICOES primary key (ID));

-- Table: FATORESQUALIDADE
create table FATORESQUALIDADE
(ID              SERIAL   not null,
 ID_CLASSEFQ     INT4     not null,
 PESO            NUMERIC(3),
 NOME            CHAR(20),
constraint       PK_FATORESQUALIDADE primary key (ID));

-- Table: FILTROS
create table FILTROS
(ID              SERIAL   not null,
 ID_RESTRICAO    INT4     not null,
 TIPO_FILTRO     CHAR(10),
 ATRIBUTO        CHAR(20),
 SINAL           CHAR(4),
 VALOR1          CHAR(100),
 VALOR2          CHAR(100),
 constraint      PK_FILTROS primary key (ID));

-- Table: PERFIS
create table PERFIS
(ID           SERIAL   not null,
 ID_USUARIO   INT4     not null,
 NOME         CHAR(30),
 TIPO         CHAR(30),
 constraint   PK_PERFIS primary key (ID));

-- Table: RESTRICOES
create table RESTRICOES
(ID                  SERIAL   not null,
 ID_FATORQUALIDADE   INT4     not null,
 PESO                NUMERIC(3),
 constraint          PK_RESTRICOES primary key (ID));

-- Table: RESTRICOESATRIBUTO
create table RESTRICOESATRIBUTO
(ID                   SERIAL   not null,
 ID_RESTRICAO         INT4     not null,
```

```
 ATRIBUTO             CHAR(20),
 PONTOS_DEFAULT       NUMERIC(1),
 PESO                 NUMERIC(3),
 constraint PK_RESTRICOESATRIBUTO primary key (ID));

-- Table: USUARIOS
create table USUARIOS
(ID          SERIAL   not null,
 NOME        CHAR(30),
 constraint  PK_USUARIOS primary key (ID));

-- CRIAÇÃO DE CHAVES ESTRANGEIRAS

alter table APPFATORES
   add constraint FK_APPFATOR_REFERENCE_APPCLASS foreign key
(ID_CLASSES)
      references APPCLASSES (ID_CLASSES)
      on delete restrict on update restrict;

alter table CLASSESFQ
   add constraint FK_CLASSESF_REFERENCE_PERFIS foreign key (ID_PERFIL)
      references PERFIS (ID)
      on delete cascade on update restrict;

alter table CONDICOES
   add constraint FK_CONDICOES_REFERENCE_RESTRICA foreign key
(ID_RESTRICOESATRIBUTO)
      references RESTRICOESATRIBUTO (ID)
      on delete cascade on update restrict;

alter table FATORESQUALIDADE
   add constraint FK_FATORESQ_REFERENCE_CLASSESF foreign key
(ID_CLASSEFQ)
      references CLASSESFQ (ID)
      on delete cascade on update restrict;

alter table FILTROS
   add constraint FK_FILTROS_REFERENCE_RESTRICA foreign key
(ID_RESTRICAO)
      references RESTRICOES (ID)
      on delete cascade on update restrict;

alter table PERFIS
   add constraint FK_PERFIS_REFERENCE_USUARIOS foreign key
(ID_USUARIO)
      references USUARIOS (ID)
      on delete cascade on update restrict;

alter table RESTRICOES
   add constraint FK_RESTRICA_REFERENCE_FATORESQ foreign key
(ID_FATORQUALIDADE)
      references FATORESQUALIDADE (ID)
      on delete cascade on update restrict;

alter table RESTRICOESATRIBUTO
   add constraint FK_RESTRICA_REFERENCE_RESTRICA foreign key
(ID_RESTRICAO)
      references RESTRICOES (ID)
      on delete cascade on update restrict;
```

***{cria_base_perfis.sql}***

## *ANEXO II – Script de Criação de Tabelas do TPC-H*

***{cria_tpch.sql}***

```
-- APAGA AS TABELAS

DROP TABLE nation;
DROP TABLE region;
DROP TABLE part;
DROP TABLE partsupp;
DROP TABLE supplier;
DROP TABLE customer;
DROP TABLE orders;
DROP TABLE lineitem;


-- CRIAÇÃO DE TABELAS DO TPC-H

-- Table: nation
CREATE TABLE nation
( n_nationkey numeric NOT NULL,
  n_name char(25),
  n_regionkey int4,
  n_comment varchar(152),
  CONSTRAINT nation_pkey PRIMARY KEY (n_nationkey));

-- Table: region
CREATE TABLE region
( r_regionkey numeric NOT NULL,
  r_name char(25),
  r_comment varchar(152),
  CONSTRAINT region_pkey PRIMARY KEY (r_regionkey));

-- Table: part
CREATE TABLE part
( p_partkey numeric NOT NULL,
  p_name varchar(55),
  p_mfgr char(25),
  p_brand char(10),
  p_type varchar(25),
  p_size int4,
  p_container char(10),
  p_retailprice float8,
  p_comment varchar(23),
  CONSTRAINT part_pkey PRIMARY KEY (p_partkey));

-- Table: partsupp
CREATE TABLE partsupp
( ps_partkey numeric NOT NULL,
  ps_suppkey numeric NOT NULL,
  ps_availqty int4,
  ps_supplycost numeric,
  ps_comment varchar(199),
  CONSTRAINT partsupp_pkey PRIMARY KEY (ps_partkey, ps_suppkey));

-- Table: supplier
CREATE TABLE supplier
( s_suppkey numeric NOT NULL,
```

```
  s_name char(25),
  s_address varchar(40),
  s_nationkey numeric,
  s_phone char(15),
  s_acctbal float8,
  s_comment varchar(101),
  CONSTRAINT supplier_pkey PRIMARY KEY (s_suppkey));

-- Table: customer
CREATE TABLE customer
( c_custkey numeric NOT NULL,
  c_name varchar(25),
  c_address varchar(40),
  c_nationkey numeric,
  c_phone char(15),
  c_acctbal numeric,
  c_mktsegment char(10),
  c_comment varchar(117),
  CONSTRAINT customer_pkey PRIMARY KEY (c_custkey));

-- Table: orders
CREATE TABLE orders
( o_orderkey numeric NOT NULL,
  o_custkey numeric,
  o_orderstatus char(1),
  o_totalprice numeric,
  o_orderdate date,
  o_orderpriority char(15),
  o_clerk char(15),
  o_shippriority int4,
  o_comment varchar(79),
  CONSTRAINT orders_pkey PRIMARY KEY (o_orderkey));


-- Table: lineitem
CREATE TABLE lineitem
( l_orderkey numeric NOT NULL,
  l_partkey numeric NOT NULL,
  l_suppkey numeric NOT NULL,
  l_linenumber numeric,
  l_quantity numeric,
  l_extendedprice float8,
  l_discount float8,
  l_tax float8,
  l_returnflag char(1),
  l_linestatus char(1),
  l_shipdate date,
  l_commitdate date,
  l_receiptdate date,
  l_shipinstruct char(25),
  l_shipmode char(10),
  l_comment varchar(44),
  CONSTRAINT lineitem_pkey PRIMARY KEY (l_orderkey, l_partkey,
l_suppkey));
```

***{cria_tpch.sql}***

## *ANEXO III –Script de Criação do Esquema Dimensional*

***{cria_dim.sql}***

```
-- APAGA AS DIMENSÕES e FATO

DROP TABLE supplier_dim;
DROP TABLE customer_dim;
DROP TABLE part_dim;
DROP TABLE time_dim;
DROP TABLE f_lineorder;

-- CRIAÇÃO DIMENSÕES e FATO

-- Dimension: supplier_dim
CREATE TABLE supplier_dim
( s_suppkey numeric,
  s_name char(25),
  s_address varchar(40),
  s_phone char(15),
  s_acctbal float8,
  s_nation char(25),
  s_region char(25));

-- Dimension: customer_dim
CREATE TABLE customer_dim
( c_custkey numeric,
  c_name char(25),
  c_address varchar(40),
  c_phone char(15),
  c_acctbal float8,
  c_mktsegment char(10),
  c_nation char(25),
  c_region char(25));

-- Dimension part_dim
CREATE TABLE part_dim
( p_partkey numeric,
  p_name varchar(55),
  p_mfgr char(25),
  p_category char(10),
  p_type varchar(25),
  p_size int4,
  p_container char(10));

-- Dimension: time_dim
CREATE TABLE time_dim
( d_datekey int2,
  d_date timestamp,
  d_year int2,
  d_month int2,
  d_dayofyear int2,
  d_dayofmonth int2,
  d_dayofweek int2,
  d_weekofyear int2,
  d_dayofweekdesc varchar(30),
  d_dayofweekshortdesc varchar(3),
  d_monthdesc varchar(30),
  d_monthshortdesc varchar(3),
  d_quarter varchar(1));
```

```
-- Fact: f_lineorder
CREATE TABLE f_lineorder
( l_orderkey numeric NOT NULL,
  l_custkey numeric NOT NULL,
  l_partkey numeric NOT NULL,
  l_suppkey numeric NOT NULL,
  l_ordtotalprice numeric,
  l_orderdate date,
  l_orderpriority char(15),
  l_shippriority int4,
  l_linenumber numeric,
  l_quantity numeric,
  l_extendedprice float8,
  l_discount float8,
  l_tax float8,
  l_commitdate date,
  l_shipmode char(10));
```

***{cria_dim.sql}***

## *ANEXO IV – Carga Inicial das Dimensões*

***{carga_dim.sql}***

```
-- LIMPA AS DIMENSõES

delete from supplier_dim;
delete from customer_dim;
delete from part_dim;
commit;


--LIMPA LOG
delete from log;
commit;


-- CARGA supplier_dim
insert into supplier_dim (select s_suppkey, s_name, s_address,
                                 s_phone, s_acctbal,
                                 n_name as s_nation,
                                 r_name as s_region
                            from supplier, nation, region
                           where s_nationkey = n_nationkey
                             and n_regionkey = r_regionkey);
commit;


-- CARGA customer_dim
insert into customer_dim (select c_custkey, c_name, c_address,
                                 c_phone, c_acctbal, c_mktsegment,
                                 n_name as c_nation,
                                 r_name as c_region
                            from customer, nation, region
                           where c_nationkey = n_nationkey
                             and n_regionkey = r_regionkey);
commit;


-- CARGA part_dim
insert into part_dim (select p_partkey, p_name, p_mfgr,
                             p_brand as p_category, p_type,
                             p_size, p_container
                        from part);
commit;
```

***{carga_dim.sql}***

## *ANEXO V – Visões para Carga na Tabela Fato*

***{cria_visoes.sql}***

```
-- VISÃO DE DADOS PARA CARGA NA TABELA FATO EM AMBIENTE DW PADRÃO.
-- Dados Fontes: Esquema TPC-H.


CREATE  VIEW vw_padrao AS
      SELECT l.l_orderkey, l.o_custkey, l.l_partkey, l.l_suppkey,
             l.o_totalprice, l.o_orderdate, l.o_orderpriority,
             l.o_shippriority, l.l_linenumber, l.l_quantity,
             l.l_extendedprice, l.l_discount, l.l_tax,
             l.l_commitdate, l.l_shipmode
        FROM
             (SELECT lineitem.l_orderkey, lineitem.l_partkey,
                     lineitem.l_suppkey, lineitem.l_linenumber,
                     lineitem.l_quantity, lineitem.l_extendedprice,
                     lineitem.l_discount, lineitem.l_tax,
                     lineitem.l_commitdate, lineitem.l_shipmode,
                     orders.o_orderkey, orders.o_custkey,
                     orders.o_totalprice, orders.o_orderdate,
                     orders.o_orderpriority, orders.o_shippriority
                FROM orders, lineitem
               WHERE orders.o_orderkey = lineitem.l_orderkey) l,
             (SELECT customer.c_custkey, customer.c_name,
                     customer.c_address, customer.c_phone,
                     customer.c_acctbal, customer.c_mktsegment,
                     nation.n_name AS c_nation,
                     region.r_name AS c_region
                FROM customer, nation, region
               WHERE customer.c_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) c,
             (SELECT supplier.s_suppkey, supplier.s_name,
                     supplier.s_address, supplier.s_phone,
                     supplier.s_phone, nation.n_name AS s_nation,
                     region.r_name AS s_region
                FROM supplier, nation, region
               WHERE supplier.s_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) s,
             (SELECT part.p_partkey, part.p_name, part.p_mfgr,
                     part.p_brand AS p_category, part.p_type,
                     part.p_size, part.p_container
                FROM part) p
       WHERE l.o_custkey = c.c_custkey
          AND l.l_suppkey = s.s_suppkey
          AND l.l_partkey = p.p_partkey;
```

```
-- VISÃO DE DADOS PARA CARGA NA TABELA FATO EM AMBIENTE DW BRAHMA.
-- Dados Fontes: Esquema TPC-H.

-- TEMPORALIDADE DIÁRIA


CREATE  VIEW vw_diaria AS
      SELECT l.l_orderkey, l.o_custkey, l.l_partkey, l.l_suppkey,
             l.o_totalprice, l.o_orderdate, l.o_orderpriority,
             l.o_shippriority, l.l_linenumber, l.l_quantity,
             l.l_extendedprice, l.l_discount, l.l_tax,
             l.l_commitdate, l.l_shipmode
        FROM
             (SELECT lineitem.l_orderkey, lineitem.l_partkey,
                     lineitem.l_suppkey, lineitem.l_linenumber,
                     lineitem.l_quantity, lineitem.l_extendedprice,
                     lineitem.l_discount, lineitem.l_tax,
                     lineitem.l_commitdate, lineitem.l_shipmode,
                     orders.o_orderkey, orders.o_custkey,
                     orders.o_totalprice, orders.o_orderdate,
                     orders.o_orderpriority, orders.o_shippriority
                FROM orders, lineitem
               WHERE orders.o_orderkey = lineitem.l_orderkey) l,
             (SELECT customer.c_custkey, customer.c_name,
                     customer.c_address, customer.c_phone,
                     customer.c_acctbal, customer.c_mktsegment,
                     nation.n_name AS c_nation,
                     region.r_name AS c_region
                FROM customer, nation, region
               WHERE customer.c_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) c,
             (SELECT supplier.s_suppkey, supplier.s_name,
                     supplier.s_address, supplier.s_phone,
                     supplier.s_phone, nation.n_name AS s_nation,
                     region.r_name AS s_region
                FROM supplier, nation, region
               WHERE supplier.s_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) s,
             (SELECT part.p_partkey, part.p_name, part.p_mfgr,
                     part.p_brand AS p_category, part.p_type,
                     part.p_size, part.p_container
                FROM part) p
       WHERE l.o_custkey = c.c_custkey
         AND l.l_suppkey = s.s_suppkey
         AND l.l_partkey = p.p_partkey;
         AND (s.s_nation = 'RUSSIA'    OR
              c.c_nation = 'ARGENTINA' OR
              c.c_nation = 'BRAZIL'    OR
              c.c_nation = 'PERU');
```

```
-- VISÃO DE DADOS PARA CARGA NA TABELA FATO EM AMBIENTE DW BRAHMA.
-- Dados Fontes: Esquema TPC-H.

-- TEMPORALIDADE MENSAL


CREATE  VIEW vw_mensal AS
      SELECT l.l_orderkey, l.o_custkey, l.l_partkey, l.l_suppkey,
             l.o_totalprice, l.o_orderdate, l.o_orderpriority,
             l.o_shippriority, l.l_linenumber, l.l_quantity,
             l.l_extendedprice, l.l_discount, l.l_tax,
             l.l_commitdate, l.l_shipmode
        FROM
             (SELECT lineitem.l_orderkey, lineitem.l_partkey,
                     lineitem.l_suppkey, lineitem.l_linenumber,
                     lineitem.l_quantity, lineitem.l_extendedprice,
                     lineitem.l_discount, lineitem.l_tax,
                     lineitem.l_commitdate, lineitem.l_shipmode,
                     orders.o_orderkey, orders.o_custkey,
                     orders.o_totalprice, orders.o_orderdate,
                     orders.o_orderpriority, orders.o_shippriority
                FROM orders, lineitem
               WHERE orders.o_orderkey = lineitem.l_orderkey) l,
             (SELECT customer.c_custkey, customer.c_name,
                     customer.c_address, customer.c_phone,
                     customer.c_acctbal, customer.c_mktsegment,
                     nation.n_name AS c_nation,
                     region.r_name AS c_region
                FROM customer, nation, region
               WHERE customer.c_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) c,
             (SELECT supplier.s_suppkey, supplier.s_name,
                     supplier.s_address, supplier.s_phone,
                     supplier.s_phone, nation.n_name AS s_nation,
                     region.r_name AS s_region
                FROM supplier, nation, region
               WHERE supplier.s_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) s,
             (SELECT part.p_partkey, part.p_name, part.p_mfgr,
                     part.p_brand AS p_category, part.p_type,
                     part.p_size, part.p_container
                FROM part) p
       WHERE l.o_custkey = c.c_custkey
         AND l.l_suppkey = s.s_suppkey
         AND l.l_partkey = p.p_partkey;
         AND NOT s.s_nation = 'RUSSIA'
         AND NOT c.c_nation = 'ARGENTINA'
         AND NOT c.c_nation = 'BRAZIL'
         AND NOT c.c_nation = 'PERU'
         AND (c.c_region = 'EUROPE' OR
              c.c_region = 'AMERICA');
```

```
-- VISÃO DE DADOS PARA CARGA NA TABELA FATO EM AMBIENTE DW BRAHMA.
-- Dados Fontes: Esquema TPC-H.

-- TEMPORALIDADE ANUAL


CREATE  VIEW vw_anual AS
      SELECT l.l_orderkey, l.o_custkey, l.l_partkey, l.l_suppkey,
             l.o_totalprice, l.o_orderdate, l.o_orderpriority,
             l.o_shippriority, l.l_linenumber, l.l_quantity,
             l.l_extendedprice, l.l_discount, l.l_tax,
             l.l_commitdate, l.l_shipmode
        FROM
             (SELECT lineitem.l_orderkey, lineitem.l_partkey,
                     lineitem.l_suppkey, lineitem.l_linenumber,
                     lineitem.l_quantity, lineitem.l_extendedprice,
                     lineitem.l_discount, lineitem.l_tax,
                     lineitem.l_commitdate, lineitem.l_shipmode,
                     orders.o_orderkey, orders.o_custkey,
                     orders.o_totalprice, orders.o_orderdate,
                     orders.o_orderpriority, orders.o_shippriority
                FROM orders, lineitem
               WHERE orders.o_orderkey = lineitem.l_orderkey) l,
             (SELECT customer.c_custkey, customer.c_name,
                     customer.c_address, customer.c_phone,
                     customer.c_acctbal, customer.c_mktsegment,
                     nation.n_name AS c_nation,
                     region.r_name AS c_region
                FROM customer, nation, region
               WHERE customer.c_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) c,
             (SELECT supplier.s_suppkey, supplier.s_name,
                     supplier.s_address, supplier.s_phone,
                     supplier.s_phone, nation.n_name AS s_nation,
                     region.r_name AS s_region
                FROM supplier, nation, region
               WHERE supplier.s_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) s,
             (SELECT part.p_partkey, part.p_name, part.p_mfgr,
                     part.p_brand AS p_category, part.p_type,
                     part.p_size, part.p_container
                FROM part) p
       WHERE l.o_custkey = c.c_custkey
         AND l.l_suppkey = s.s_suppkey
         AND l.l_partkey = p.p_partkey;
         AND NOT s.s_nation = 'RUSSIA'
         AND NOT c.c_nation = 'ARGENTINA'
         AND NOT c.c_nation = 'BRAZIL'
         AND NOT c.c_nation = 'PERU'
         AND NOT c.c_region = 'EUROPE'
         AND NOT c.c_region = 'AMERICA'
         AND s.s_region = 'MIDDLE EAST';
```

```
-- VISÃO DE DADOS PARA CARGA NA TABELA FATO EM AMBIENTE DW BRAHMA.
-- Dados Fontes: Esquema TPC-H.

-- COMPLEMENTAR FINAL


CREATE  VIEW vw_anual_complementar AS
      SELECT l.l_orderkey, l.o_custkey, l.l_partkey, l.l_suppkey,
             l.o_totalprice, l.o_orderdate, l.o_orderpriority,
             l.o_shippriority, l.l_linenumber, l.l_quantity,
             l.l_extendedprice, l.l_discount, l.l_tax,
             l.l_commitdate, l.l_shipmode
        FROM
             (SELECT lineitem.l_orderkey, lineitem.l_partkey,
                     lineitem.l_suppkey, lineitem.l_linenumber,
                     lineitem.l_quantity, lineitem.l_extendedprice,
                     lineitem.l_discount, lineitem.l_tax,
                     lineitem.l_commitdate, lineitem.l_shipmode,
                     orders.o_orderkey, orders.o_custkey,
                     orders.o_totalprice, orders.o_orderdate,
                     orders.o_orderpriority, orders.o_shippriority
                FROM orders, lineitem
               WHERE orders.o_orderkey = lineitem.l_orderkey) l,
             (SELECT customer.c_custkey, customer.c_name,
                     customer.c_address, customer.c_phone,
                     customer.c_acctbal, customer.c_mktsegment,
                     nation.n_name AS c_nation,
                     region.r_name AS c_region
                FROM customer, nation, region
               WHERE customer.c_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) c,
             (SELECT supplier.s_suppkey, supplier.s_name,
                     supplier.s_address, supplier.s_phone,
                     supplier.s_phone, nation.n_name AS s_nation,
                     region.r_name AS s_region
                FROM supplier, nation, region
               WHERE supplier.s_nationkey = nation.n_nationkey
                 AND nation.n_regionkey = region.r_regionkey) s,
             (SELECT part.p_partkey, part.p_name, part.p_mfgr,
                     part.p_brand AS p_category, part.p_type,
                     part.p_size, part.p_container
                FROM part) p
       WHERE l.o_custkey = c.c_custkey
         AND l.l_suppkey = s.s_suppkey
         AND l.l_partkey = p.p_partkey;
         AND NOT s.s_nation = 'RUSSIA'
         AND NOT c.c_nation = 'ARGENTINA'
         AND NOT c.c_nation = 'BRAZIL'
         AND NOT c.c_nation = 'PERU'
         AND NOT c.c_region = 'EUROPE'
         AND NOT c.c_region = 'AMERICA'
         AND NOT s.s_region = 'MIDDLE EAST';


{cria_visoes.sql}
```

## *ANEXO VI – Carga na Tabela Fato*

```
-- Carga Diária para Tabela Fato: f_lineorder

CREATE FUNCTION carga_diaria (in ano int8, in mês int8, in dia int8)
$$
insert into
       f_lineorder (select *
                      from vw_diaria
                     where extract (year from o_orderdate)  = $1
                       and extract (month from o_orderdate) = $2
                       and extract (day from o_orderdate)   = $3);
commit;
$$

-- Carga Mensal para Tabela Fato: f_lineorder

CREATE FUNCTION carga_diaria (in ano int8, in mês int8, in dia int8)
$$
insert into
       f_lineorder (select *
                      from vw_mensal
                     where extract (year from o_orderdate)  = $1
                       and extract (month from o_orderdate) = $2);
commit;
$$


-- Carga Anual para Tabela Fato: f_lineorder

CREATE FUNCTION carga_anual (in ano int8, in mês int8, in dia int8)
$$
insert into
       f_lineorder (select *
                      from vw_anual
                     where extract (year from o_orderdate)  = $1);
commit;
$$



-- Carga Anual Complementar para Tabela Fato: f_lineorder

CREATE FUNCTION carga_anual_complementar (in ano int8)
$$
insert into
       f_lineorder (select *
                      from vw_anual_complementar
                     where extract (year from o_orderdate)  = $1);
commit;
$$
```

## *ANEXO VII – Arquivo de LOG (registros / tempo de execução)*

| idlog | atividade | numlines | startdate | enddate |
|---|---|---|---|---|
| 1 | Carga padrão 01/1996 | 76859 | 2009-06-17 15:56:00.078 | 2009-06-17 15:56:03.453 |
| 2 | Carga padrão 02/1996 | 73197 | 2009-06-17 16:05:55.218 | 2009-06-17 16:06:00.421 |
| 3 | Carga padrão 03/1996 | 76283 | 2009-06-17 16:16:13.406 | 2009-06-17 16:16:19.109 |
| 4 | Carga padrão 02/1996 | 74941 | 2009-06-17 16:26:19.000 | 2009-06-17 16:26:25.781 |
| 5 | Carga padrão 05/1996 | 76445 | 2009-06-17 16:36:37.281 | 2009-06-17 16:36:44.218 |
| 6 | Carga padrão 06/1996 | 75727 | 2009-06-17 16:47:00.109 | 2009-06-17 16:47:08.437 |
| 7 | Carga padrão 07/1996 | 76868 | 2009-06-17 18:51:11.765 | 2009-06-17 18:51:17.515 |
| 8 | Carga padrão 08/1996 | 79101 | 2009-06-17 19:01:50.796 | 2009-06-17 19:01:56.562 |
| 9 | Carga padrão 09/1996 | 75401 | 2009-06-17 19:11:57.234 | 2009-06-17 19:12:03.187 |
| 10 | Carga padrão 10/1996 | 77467 | 2009-06-17 19:22:19.968 | 2009-06-17 19:22:26.359 |
| 11 | Carga padrão 11/1996 | 75472 | 2009-06-17 19:32:45.890 | 2009-06-17 19:32:48.453 |
| 12 | Carga padrão 12/1996 | 77728 | 2009-06-17 19:43:10.296 | 2009-06-17 19:43:18.750 |
| 21 | Carga Brahma 01/1996 | 21183 | 2009-06-17 20:23:00.921 | 2009-06-17 20:23:02.281 |
| 22 | Carga Brahma 02/1996 | 19973 | 2009-06-17 20:23:02.312 | 2009-06-17 20:23:02.843 |
| 23 | Carga Brahma 03/1996 | 20617 | 2009-06-17 20:23:02.843 | 2009-06-17 20:23:03.390 |
| 24 | Carga Brahma 04/1996 | 20559 | 2009-06-17 20:23:03.390 | 2009-06-17 20:23:03.781 |
| 25 | Carga Brahma 05/1996 | 20665 | 2009-06-17 20:23:03.781 | 2009-06-17 20:23:05.437 |
| 26 | Carga Brahma 06/1996 | 20743 | 2009-06-17 20:23:05.453 | 2009-06-17 20:23:05.953 |
| 27 | Carga Brahma 07/1996 | 20895 | 2009-06-17 21:11:53.921 | 2009-06-17 21:11:58.015 |
| 28 | Carga Brahma 08/1996 | 21192 | 2009-06-17 21:11:58.046 | 2009-06-17 21:11:59.640 |
| 29 | Carga Brahma 09/1996 | 20259 | 2009-06-17 21:11:59.640 | 2009-06-17 21:12:00.562 |
| 30 | Carga Brahma 10/1996 | 20352 | 2009-06-17 21:12:00.562 | 2009-06-17 21:12:04.328 |
| 31 | Carga Brahma 11/1996 | 20510 | 2009-06-17 21:12:04.328 | 2009-06-17 21:12:06.531 |
| 32 | Carga Brahma 12/1996 | 21191 | 2009-06-17 21:12:06.531 | 2009-06-17 21:12:07.671 |
| 41 | Consulta padrão 01/1996 | 76859 | 2009-06-17 23:16:00.437 | 2009-06-17 23:26:35.312 |
| 42 | Consulta padrão 02/1996 | 73197 | 2009-06-17 23:26:35.312 | 2009-06-17 23:36:40.437 |
| 43 | Consulta padrão 03/1996 | 76283 | 2009-06-17 23:36:40.437 | 2009-06-17 23:47:04.281 |
| 44 | Consulta padrão 04/1996 | 74941 | 2009-06-17 23:47:04.281 | 2009-06-17 23:57:33.734 |
| 45 | Consulta padrão 05/1996 | 76445 | 2009-06-17 23:57:33.750 | 2009-06-18 00:08:13.250 |
| 46 | Consulta padrão 06/1996 | 75727 | 2009-06-18 00:08:13.265 | 2009-06-18 00:18:22.109 |
| 47 | Consulta padrão 07/1996 | 76868 | 2009-06-18 00:18:22.109 | 2009-06-18 00:28:37.625 |
| 48 | Consulta padrão 08/1996 | 79101 | 2009-06-18 00:28:37.625 | 2009-06-18 00:39:15.953 |
| 49 | Consulta padrão 09/1996 | 75401 | 2009-06-18 00:39:15.953 | 2009-06-18 00:49:24.531 |
| 50 | Consulta padrão 10/1996 | 77467 | 2009-06-18 00:49:24.531 | 2009-06-18 00:59:42.437 |
| 51 | Consulta padrão 11/1996 | 75472 | 2009-06-18 00:59:42.437 | 2009-06-18 01:10:01.484 |
| 52 | Consulta padrão 12/1996 | 77728 | 2009-06-18 01:10:01.484 | 2009-06-18 01:20:20.046 |
| 61 | Consulta Brahma 01/1996 | 21183 | 2009-06-17 22:21:34.203 | 2009-06-17 22:25:31.875 |
| 62 | Consulta Brahma 02/1996 | 19973 | 2009-06-17 22:25:31.890 | 2009-06-17 22:29:21.953 |
| 63 | Consulta Brahma 03/1996 | 20617 | 2009-06-17 22:29:21.953 | 2009-06-17 22:33:33.015 |
| 64 | Consulta Brahma 04/1996 | 20559 | 2009-06-17 22:33:33.015 | 2009-06-17 22:37:37.390 |
| 65 | Consulta Brahma 05/1996 | 20665 | 2009-06-17 22:37:37.390 | 2009-06-17 22:41:45.859 |
| 66 | Consulta Brahma 06/1996 | 20743 | 2009-06-17 22:41:45.859 | 2009-06-17 22:45:49.546 |
| 67 | Consulta Brahma 07/1996 | 20895 | 2009-06-17 22:50:44.718 | 2009-06-17 22:54:49.203 |
| 68 | Consulta Brahma 08/1996 | 21192 | 2009-06-17 22:54:49.203 | 2009-06-17 22:58:35.515 |
| 69 | Consulta Brahma 09/1996 | 20259 | 2009-06-17 22:58:35.515 | 2009-06-17 23:02:18.359 |
| 70 | Consulta Brahma 10/1996 | 20352 | 2009-06-17 23:02:18.359 | 2009-06-17 23:06:00.015 |
| 71 | Consulta Brahma 11/1996 | 20510 | 2009-06-17 23:06:00.015 | 2009-06-17 23:10:53.046 |
| 72 | Consulta Brahma 12/1996 | 21191 | 2009-06-17 23:10:53.046 | 2009-06-17 23:14:50.140 |

## *ANEXO VIII – Arquivo de LOG (pontuação perfis)*

| idlog | descricao | score |
|---|---|---|
| 1 | Score Diário Perfil A - 01/12/1996 BRAHMA | 4,10 |
| 2 | Score Diário Perfil A - 02/12/1996 BRAHMA | 3,30 |
| 3 | Score Diário Perfil A - 03/12/1996 BRAHMA | 4,00 |
| 4 | Score Diário Perfil A - 04/12/1996 BRAHMA | 4,00 |
| 5 | Score Diário Perfil A - 05/12/1996 BRAHMA | 4,10 |
| 6 | Score Diário Perfil A - 06/12/1996 BRAHMA | 4,00 |
| 7 | Score Diário Perfil A - 07/12/1996 BRAHMA | 4,10 |
| 8 | Score Diário Perfil A - 08/12/1996 BRAHMA | 4,00 |
| 9 | Score Diário Perfil A - 09/12/1996 BRAHMA | 4,00 |
| 10 | Score Diário Perfil A - 10/12/1996 BRAHMA | 4,00 |
| 11 | Score Diário Perfil A - 11/12/1996 BRAHMA | 4,10 |
| 12 | Score Diário Perfil A - 12/12/1996 BRAHMA | 4,10 |
| 13 | Score Diário Perfil A - 13/12/1996 BRAHMA | 3,20 |
| 14 | Score Diário Perfil A - 14/12/1996 BRAHMA | 4,00 |
| 15 | Score Diário Perfil A - 15/12/1996 BRAHMA | 3,30 |
| 16 | Score Diário Perfil A - 16/12/1996 BRAHMA | 4,00 |
| 17 | Score Diário Perfil A - 17/12/1996 BRAHMA | 4,00 |
| 18 | Score Diário Perfil A - 18/12/1996 BRAHMA | 4,10 |
| 19 | Score Diário Perfil A - 19/12/1996 BRAHMA | 4,10 |
| 20 | Score Diário Perfil A - 20/12/1996 BRAHMA | 4,00 |
| 21 | Score Diário Perfil A - 21/12/1996 BRAHMA | 4,10 |
| 22 | Score Diário Perfil A - 22/12/1996 BRAHMA | 4,10 |
| 23 | Score Diário Perfil A - 23/12/1996 BRAHMA | 4,10 |
| 24 | Score Diário Perfil A - 24/12/1996 BRAHMA | 4,10 |
| 25 | Score Diário Perfil A - 25/12/1996 BRAHMA | 4,10 |
| 26 | Score Diário Perfil A - 26/12/1996 BRAHMA | 3,30 |
| 27 | Score Diário Perfil A - 27/12/1996 BRAHMA | 4,00 |
| 28 | Score Diário Perfil A - 28/12/1996 BRAHMA | 4,10 |
| 29 | Score Diário Perfil A - 29/12/1996 BRAHMA | 4,10 |
| 30 | Score Diário Perfil A - 30/12/1996 BRAHMA | 4,00 |
| 31 | Score Diário Perfil A - 31/12/1996 BRAHMA | 4,00 |
| 32 | Score Diário Perfil B - 01/12/1996 BRAHMA | 0,42 |
| 33 | Score Diário Perfil B - 02/12/1996 BRAHMA | 0,72 |
| 34 | Score Diário Perfil B - 03/12/1996 BRAHMA | 0,48 |
| 35 | Score Diário Perfil B - 04/12/1996 BRAHMA | 0,42 |
| 36 | Score Diário Perfil B - 05/12/1996 BRAHMA | 0,48 |
| 37 | Score Diário Perfil B - 06/12/1996 BRAHMA | 0,42 |
| 38 | Score Diário Perfil B - 07/12/1996 BRAHMA | 0,42 |
| 39 | Score Diário Perfil B - 08/12/1996 BRAHMA | 0,54 |
| 40 | Score Diário Perfil B - 09/12/1996 BRAHMA | 0,72 |
| 41 | Score Diário Perfil B - 10/12/1996 BRAHMA | 0,42 |
| 42 | Score Diário Perfil B - 11/12/1996 BRAHMA | 0,60 |
| 43 | Score Diário Perfil B - 12/12/1996 BRAHMA | 0,48 |
| 44 | Score Diário Perfil B - 13/12/1996 BRAHMA | 0,42 |
| 45 | Score Diário Perfil B - 14/12/1996 BRAHMA | 0,42 |
| 46 | Score Diário Perfil B - 15/12/1996 BRAHMA | 0,42 |
| 47 | Score Diário Perfil B - 16/12/1996 BRAHMA | 0,78 |
| 48 | Score Diário Perfil B - 17/12/1996 BRAHMA | 0,72 |
| 49 | Score Diário Perfil B - 18/12/1996 BRAHMA | 0,42 |

| | | |
|---|---|---|
| 50 | Score Diário Perfil B - 19/12/1996 BRAHMA | 0,42 |
| 51 | Score Diário Perfil B - 20/12/1996 BRAHMA | 0,42 |
| 52 | Score Diário Perfil B - 21/12/1996 BRAHMA | 0,54 |
| 53 | Score Diário Perfil B - 22/12/1996 BRAHMA | 0,72 |
| 54 | Score Diário Perfil B - 23/12/1996 BRAHMA | 0,72 |
| 55 | Score Diário Perfil B - 24/12/1996 BRAHMA | 0,54 |
| 56 | Score Diário Perfil B - 25/12/1996 BRAHMA | 0,24 |
| 57 | Score Diário Perfil B - 26/12/1996 BRAHMA | 0,48 |
| 58 | Score Diário Perfil B - 27/12/1996 BRAHMA | 0,60 |
| 59 | Score Diário Perfil B - 28/12/1996 BRAHMA | 0,42 |
| 60 | Score Diário Perfil B - 29/12/1996 BRAHMA | 0,60 |
| 61 | Score Diário Perfil B - 30/12/1996 BRAHMA | 0,42 |
| 62 | Score Diário Perfil B - 31/12/1996 BRAHMA | 0,60 |
| 63 | Score Diário Perfil C - 01/12/1996 BRAHMA | 7,00 |
| 64 | Score Diário Perfil C - 02/12/1996 BRAHMA | 6,60 |
| 65 | Score Diário Perfil C - 03/12/1996 BRAHMA | 6,20 |
| 66 | Score Diário Perfil C - 04/12/1996 BRAHMA | 6,00 |
| 67 | Score Diário Perfil C - 05/12/1996 BRAHMA | 6,00 |
| 68 | Score Diário Perfil C - 06/12/1996 BRAHMA | 5,80 |
| 69 | Score Diário Perfil C - 07/12/1996 BRAHMA | 5,80 |
| 70 | Score Diário Perfil C - 08/12/1996 BRAHMA | 5,80 |
| 71 | Score Diário Perfil C - 09/12/1996 BRAHMA | 5,80 |
| 72 | Score Diário Perfil C - 10/12/1996 BRAHMA | 5,80 |
| 73 | Score Diário Perfil C - 11/12/1996 BRAHMA | 5,80 |
| 74 | Score Diário Perfil C - 12/12/1996 BRAHMA | 5,80 |
| 75 | Score Diário Perfil C - 13/12/1996 BRAHMA | 5,60 |
| 76 | Score Diário Perfil C - 14/12/1996 BRAHMA | 5,60 |
| 77 | Score Diário Perfil C - 15/12/1996 BRAHMA | 5,60 |
| 78 | Score Diário Perfil C - 16/12/1996 BRAHMA | 5,60 |
| 79 | Score Diário Perfil C - 17/12/1996 BRAHMA | 5,60 |
| 80 | Score Diário Perfil C - 18/12/1996 BRAHMA | 5,60 |
| 81 | Score Diário Perfil C - 19/12/1996 BRAHMA | 5,60 |
| 82 | Score Diário Perfil C - 20/12/1996 BRAHMA | 5,60 |
| 83 | Score Diário Perfil C - 21/12/1996 BRAHMA | 5,60 |
| 84 | Score Diário Perfil C - 22/12/1996 BRAHMA | 5,60 |
| 85 | Score Diário Perfil C - 23/12/1996 BRAHMA | 5,60 |
| 86 | Score Diário Perfil C - 24/12/1996 BRAHMA | 5,60 |
| 87 | Score Diário Perfil C - 25/12/1996 BRAHMA | 5,60 |
| 88 | Score Diário Perfil C - 26/12/1996 BRAHMA | 5,60 |
| 89 | Score Diário Perfil C - 27/12/1996 BRAHMA | 5,60 |
| 90 | Score Diário Perfil C - 28/12/1996 BRAHMA | 5,60 |
| 91 | Score Diário Perfil C - 29/12/1996 BRAHMA | 5,60 |
| 92 | Score Diário Perfil C - 30/12/1996 BRAHMA | 5,60 |
| 93 | Score Diário Perfil C - 31/12/1996 BRAHMA | 5,60 |
| 94 | Score Mensal Perfil D - Janeiro/1996 Tradicional | -2,76 |
| 95 | Score Mensal Perfil D - Fevereiro/1996 Tradicional | -3,58 |
| 96 | Score Mensal Perfil D - Março/1996 Tradicional | -2,88 |
| 97 | Score Mensal Perfil D - Abril/1996 Tradicional | -3,58 |
| 98 | Score Mensal Perfil D - Maio/1996 Tradicional | -2,70 |
| 99 | Score Mensal Perfil D - Junho/1996 Tradicional | -3,58 |
| 100 | Score Mensal Perfil D - Julho/1996 Tradicional | -2,70 |
| 101 | Score Mensal Perfil D - Agosto/1996 Tradicional | -2,88 |

| | | |
|---|---|---|
| 102 | Score Mensal Perfil D - Setembro/1996 Tradicional | -3,40 |
| 103 | Score Mensal Perfil D - Outubro/1996 Tradicional | -2,00 |
| 104 | Score Mensal Perfil D - Novembro/1996 Tradicional | 0,62 |
| 105 | Score Mensal Perfil D - Dezembro/1996 Tradicional | 1,44 |
| 106 | Score Mensal Perfil E - Janeiro/1996 Tradicional | 0,75 |
| 107 | Score Mensal Perfil E - Fevereiro/1996 Tradicional | 0,75 |
| 108 | Score Mensal Perfil E - Março/1996 Tradicional | 1,00 |
| 109 | Score Mensal Perfil E - Abril/1996 Tradicional | 1,00 |
| 110 | Score Mensal Perfil E - Maio/1996 Tradicional | 1,00 |
| 111 | Score Mensal Perfil E - Junho/1996 Tradicional | 1,00 |
| 112 | Score Mensal Perfil E - Julho/1996 Tradicional | 1,00 |
| 113 | Score Mensal Perfil E - Agosto/1996 Tradicional | 1,00 |
| 114 | Score Mensal Perfil E - Setembro/1996 Tradicional | 1,00 |
| 115 | Score Mensal Perfil E - Outubro/1996 Tradicional | 1,00 |
| 116 | Score Mensal Perfil E - Novembro/1996 Tradicional | 1,00 |
| 117 | Score Mensal Perfil E - Dezembro/1996 Tradicional | 1,00 |
| 118 | Score Mensal Perfil D - Janeiro/1996 BRAHMA | -2,76 |
| 119 | Score Mensal Perfil D - Fevereiro/1996 BRAHMA | -3,58 |
| 120 | Score Mensal Perfil D - Março/1996 BRAHMA | -2,88 |
| 121 | Score Mensal Perfil D - Abril/1996 BRAHMA | -3,58 |
| 122 | Score Mensal Perfil D - Maio/1996 BRAHMA | -2,70 |
| 123 | Score Mensal Perfil D - Junho/1996 BRAHMA | -3,58 |
| 124 | Score Mensal Perfil D - Julho/1996 BRAHMA | -2,70 |
| 125 | Score Mensal Perfil D - Agosto/1996 BRAHMA | -2,88 |
| 126 | Score Mensal Perfil D - Setembro/1996 BRAHMA | -3,40 |
| 127 | Score Mensal Perfil D - Outubro/1996 BRAHMA | -2,00 |
| 128 | Score Mensal Perfil D - Novembro/1996 BRAHMA | 0,62 |
| 129 | Score Mensal Perfil D - Dezembro/1996 BRAHMA | 1,44 |
| 130 | Score Mensal Perfil E - Janeiro/1996 BRAHMA | 0,75 |
| 131 | Score Mensal Perfil E - Fevereiro/1996 BRAHMA | 0,75 |
| 132 | Score Mensal Perfil E - Março/1996 BRAHMA | 1,00 |
| 133 | Score Mensal Perfil E - Abril/1996 BRAHMA | 1,00 |
| 134 | Score Mensal Perfil E - Maio/1996 BRAHMA | 1,00 |
| 135 | Score Mensal Perfil E - Junho/1996 BRAHMA | 1,00 |
| 136 | Score Mensal Perfil E - Julho/1996 BRAHMA | 1,00 |
| 137 | Score Mensal Perfil E - Agosto/1996 BRAHMA | 1,00 |
| 138 | Score Mensal Perfil E - Setembro/1996 BRAHMA | 1,00 |
| 139 | Score Mensal Perfil E - Outubro/1996 BRAHMA | 1,00 |
| 140 | Score Mensal Perfil E - Novembro/1996 BRAHMA | 1,00 |
| 141 | Score Mensal Perfil E - Dezembro/1996 BRAHMA | 1,00 |

## *ANEXO IX – Scripts para Pontuação dos Perfis*

***{cria_funcoes_score.sql}***

```
-- Cálculo de Score Perfil A
CREATE OR REPLACE FUNCTION calcula_score_p1(OUT score int8) AS
$BODY$
select
(((((   ((select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_orderpriority='1-URGENT')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_orderpriority='2-HIGH')*5
   +(select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_orderpriority='3-MEDIUM')*3
   +(select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_orderpriority<>'1-URGENT'
        and l_orderpriority<>'2-HIGH' and l_orderpriority<>'3-
MEDIUM')*1
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_nation='RUSSIA')
) * 50)
+
((   ((select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_shipmode<>'MAIL')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where s_nation='RUSSIA'  and l_orderpriority='MAIL')*3
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_nation='RUSSIA')
) * 50))*20)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where l_extendedprice>=15000)*7
   +(select count(l_orderkey) from vw_tabelafato2
      where l_extendedprice<15000)* (-7) )
  / (select count(l_orderkey) from vw_tabelafato2)
) * 100)*80))

$BODY$
  LANGUAGE 'sql' VOLATILE;


-- Cálculo de Score Perfil B
CREATE OR REPLACE FUNCTION calcula_score_p2(OUT score int8) AS
$BODY$
select
(((((   ((select count(l_orderkey) from vw_tabelafato2
      where c_nation='BRAZIL'  and c_mktsegment='AUTOMOBILE')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='BRAZIL'  and c_mktsegment='HOUSEHOLD')*3
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='BRAZIL'  and c_mktsegment<>'AUTOMOBILE'
        and c_mktsegment<>'HOUSEHOLD')*0
    )
  / (select count(l_orderkey)  from vw_tabelafato2
```

```
      where c_nation='BRAZIL' )
) * 60)
+
((   ((select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment='HOUSEHOLD')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment='AUTOMOBILE')*5
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment<>'AUTOMOBILE' and
c_mktsegment<>'HOUSEHOLD')*0
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where c_nation='ARGENTINA')
) * 40))*30)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where l_commitdate<=l_orderdate+60)*7
   +(select count(l_orderkey) from vw_tabelafato2
      where l_commitdate>l_orderdate+60)*(-7))
  / (select count(l_orderkey) from vw_tabelafato2)
) * 100)*70))
$BODY$
  LANGUAGE 'sql' VOLATILE;


-- Cálculo de Score Perfil C
CREATE OR REPLACE FUNCTION calcula_score_p3(IN d_date date, OUT score
int8) AS
$BODY$
select
(((   ((select count(l_orderkey) from vw_tabelafato2
      where c_nation='PERU'  and l_orderpriority is not null)*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='PERU'  and l_orderpriority is null)*(-7)
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where c_nation='PERU')
) * 100)*80)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where l_orderdate=$1)*7
   +(select count(l_orderkey) from vw_tabelafato2
      where l_orderdate=($1)-1)*4
   +(select count(l_orderkey) from vw_tabelafato2
      where l_orderdate=($1)-2)*1
   +(select count(l_orderkey) from vw_tabelafato2
      where l_orderdate<($1)-2)*(0)
    )
  / (select count(l_orderkey) from vw_tabelafato2)
) * 100)*20)
$BODY$
  LANGUAGE 'sql' VOLATILE;


-- Cálculo de Score Perfil D
CREATE OR REPLACE FUNCTION calcula_score_p4(OUT score int8) AS
$BODY$
select
(((((   ((select count(l_orderkey) from vw_tabelafato2
```

```
      where c_nation='BRAZIL'  and c_mktsegment='AUTOMOBILE')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='BRAZIL'  and c_mktsegment='HOUSEHOLD')*3
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='BRAZIL'  and c_mktsegment<>'AUTOMOBILE'
        and c_mktsegment<>'HOUSEHOLD')*0
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_nation='BRAZIL')
) * 60)
+
((   ((select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment='HOUSEHOLD')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment='AUTOMOBILE')*5
   +(select count(l_orderkey) from vw_tabelafato2
      where c_nation='ARGENTINA'  and c_mktsegment<>'AUTOMOBILE'
        and c_mktsegment<>'HOUSEHOLD')*0
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_nation='ARGENTINA')
) * 40))*30)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where  extract (month from l_commitdate) = extract (month from
l_orderdate))*7
   +(select count(l_orderkey) from vw_tabelafato2
      where extract (month from l_commitdate) > extract (month from
l_orderdate))*(-7))
  / (select count(l_orderkey) from vw_tabelafato2)
) * 100)*70))
$BODY$
  LANGUAGE 'sql' VOLATILE;


-- Cálculo de Score Perfil E
CREATE OR REPLACE FUNCTION calcula_score_p5(IN d_date date, OUT score
int8) AS
$BODY$
select
(((((   ((select count(l_orderkey) from vw_tabelafato2
      where c_region='AMERICA'  and c_mktsegment='BUILDING')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_region='AMERICA'  and c_mktsegment='FURNITURE')*3
   +(select count(l_orderkey) from vw_tabelafato2
      where c_region='AMERICA'  and c_mktsegment<>'BUILDING'
        and c_mktsegment<>'FURNITURE')*0
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where c_region='AMERICA')
) * 50)
+
((   ((select count(l_orderkey) from vw_tabelafato2
      where c_region='EUROPE'  and c_mktsegment='MACHINERY')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where c_region='EUROPE'  and c_mktsegment='BUILDING')*3
   +(select count(l_orderkey) from vw_tabelafato2
      where c_region='EUROPE'  and c_mktsegment<>'MACHINERY'
        and c_mktsegment<>'BUILDING')*0
    )
```

```
  / (select count(l_orderkey)  from vw_tabelafato2
      where c_region='EUROPE')
) * 50))*50)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where extract (month from l_orderdate)= extract (month from
($1))   )*6
   +(select count(l_orderkey) from vw_tabelafato2
      where extract (month from l_orderdate)= extract (month from
($1))-1 )*3
   +(select count(l_orderkey) from vw_tabelafato2
      where extract (month from l_orderdate)= extract (month from
($1))-2)*0
   +(select count(l_orderkey) from vw_tabelafato2
      where extract (month from l_orderdate) < extract (month from
($1))-2 )*(-5)
)
   / (select count(l_orderkey) from vw_tabelafato2 )
) * 100)*50))
$BODY$
  LANGUAGE 'sql' VOLATILE;


-- Cálculo de Score Perfil F
CREATE OR REPLACE FUNCTION calcula_score_p6(OUT score int8) AS
$BODY$
select
(((((   ((select count(l_orderkey) from vw_tabelafato2
      where s_region='MIDDLE EAST'  and l_orderpriority<>'5-LOW')*5
   +(select count(l_orderkey) from vw_tabelafato2
      where s_region='MIDDLE EAST'  and l_orderpriority='5-LOW')*(-5)
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_region='MIDDLE EAST')
) * 50)
+
((   ((select count(l_orderkey) from vw_tabelafato2
      where s_region='MIDDLE EAST'  and l_shipmode<>'RAIL')*7
   +(select count(l_orderkey) from vw_tabelafato2
      where s_region='MIDDLE EAST'  and l_shipmode='RAIL')* (-7)
    )
  / (select count(l_orderkey)  from vw_tabelafato2
      where s_region='MIDDLE EAST')
) * 50))*50)

+
(((   ((select count(l_orderkey) from vw_tabelafato2
      where l_extendedprice>=10000  )*7
   +(select count(l_orderkey) from vw_tabelafato2
      where l_extendedprice>=10000   )*(-3)
)
   / (select count(l_orderkey) from vw_tabelafato2 )
) * 100)*50))
$BODY$
  LANGUAGE 'sql' VOLATILE;
```

## *ANEXO X – Criação de Arquivos de LOG*

***{cria_log_A.sql}***

```
-- CRIA TABELA DE LOG_A (registros / tempo de execução)
CREATE TABLE log_A
(idlog      int8  not null,
 atividade  varchar(255),
 numlines   int8,
 startdate  timestamp,
 enddate    timestamp);
```

***{cria_log_A.sql}***

***{cria_log_B.sql}***

```
-- CRIA TABELA DE LOG_B (pontuação dos perfis)
CREATE TABLE log_A
(idlog      int8  not null,
 descricao  varchar(255),
 score      int8);
```

***{cria_log_B.sql}***