



UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Câmpus de São José do Rio Preto

Paulo Scarpelini Neto

Estratégia para Extração, Transformação e Armazenamento em Data
Warehouse ativo baseada em políticas configuráveis de propagação de
dados

São José do Rio Preto
2013

Paulo Scarpelini Neto

Estratégia para Extração, Transformação e Armazenamento em Data
Warehouse ativo baseada em políticas configuráveis de propagação de
dados

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração – Computação Aplicada, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Orientador: Prof. Dr. Carlos Roberto Valêncio

São José do Rio Preto
2013

Paulo Scarpelini Neto

Estratégia para Extração, Transformação e Armazenamento em Data Warehouse ativo baseada em políticas configuráveis de propagação de dados

Dissertação apresentada como parte dos requisitos para obtenção do título de Mestre em Ciência da Computação, junto ao Programa de Pós-Graduação em Ciência da Computação, Área de Concentração – Computação Aplicada, do Instituto de Biociências, Letras e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, Campus de São José do Rio Preto.

Banca Examinadora

Prof. Dr. Carlos Roberto Valêncio
UNESP – São José do Rio Preto
Orientador

Prof. Dr. José Marcio Machado
UNESP – São José do Rio Preto

Prof^a. Dr^a. Marilde Terezinha Prado Santos
UFSCar – São Carlos

São José do Rio Preto
21 de fevereiro de 2013

RESUMO

Em arquiteturas de *Data Warehouse* os dados são integrados periodicamente por meio da execução de processos de Extração, Transformação e Armazenamento – ETA. A atualização desses dados de forma periódica provoca um problema referente à desatualização das informações, uma vez que as ferramentas ETAs são executadas geralmente uma vez ao dia. A crescente necessidade do mundo corporativo por análises sobre dados cada vez mais recentes evidencia a necessidade de arquiteturas DW que possuam um maior grau de atualização. Nesse contexto, surgiram os DW ativos cujo intervalo de tempo entre as execuções do processo ETAs diminuem significativamente. Para suportar o aumento da frequência das atualizações, surge a necessidade de criação de critérios para priorização dos dados a serem transferidos, uma vez que a transferência de todo e qualquer dado inserido nas fontes pode sobrecarregar os sistemas envolvidos. O trabalho proposto consiste na construção de uma estratégia denominada ETA-PoCon, que tem como objetivo a execução do processo ETA em DW ativos. A estratégia distingue-se das existentes, pois possui políticas configuráveis de propagação de informações com base em parâmetros como intervalo de tempo em que a informação deve ser transferida, volume que deve ser propagado e relevância dos dados em relação às informações contidas no repositório. É demonstrado por meio dos experimentos executados que a aplicação da estratégia proporciona uma redução considerável no número de transferência de dados ao DW, uma vez que em todos os resultados a redução no total de atualizações foi superior a 60%.

ABSTRACT

In Data Warehouse (DW) architectures data are periodically integrated by means of execution of Extraction, Transformation and Load (ETL) processes which lead to the problem of outdated information. The growing necessity in the corporate world for analysis of constantly renewed data bespeaks of the need for DW architectures with higher level of freshness. In that context, active DWs appeared having a significant reduction of the interval of time between the executions of ETL processes. To support the increased frequency of the refreshment it's necessary to create criteria to prioritise the data to be transferred, since a transfer of all and any data from the sources would overload the systems. This work consists of the construction of a strategy called ETL-PoCon to execute ETL processes in active DWs. The original contribution of this work is to provide a strategy that considerably reduces the quantity of data transfers to active DWs, besides maintaining a satisfactory level of data freshness. Said reduction is obtained by means of configurable policies of data propagation based on parameters such as: the time in which the data must be transferred, the volume to be propagated and the relevance of the data regarding to the information stored in the DW. Experiments have shown that the ETL-PoCon strategy significantly contributes towards a reduction of the overload on the systems involved in the active DW environment, since all results presented a reduction higher than 60% in the amount of DW refreshments.

Índice

Índice	i
Lista de Figuras	iv
Lista de Tabelas.....	vi
Lista de Siglas.....	vii
Capítulo 1 Introdução	1
1.1. Considerações Iniciais	1
1.2. Motivação e escopo.....	2
1.3. Objetivos.....	3
1.4. Organização do Trabalho	3
Capítulo 2 Conceitos Fundamentais.....	5
2.1. Considerações iniciais	5
2.2. Sistemas de Integração de Dados	6
2.3. Arquitetura Data Warehouse.....	7
2.4. Extração, Transformação e Armazenamento - ETA	8
2.4.1. Extração	8
2.4.2. Transformação	8
2.4.3. Armazenamento	9
2.4.4. Execução do processo ETA.....	9
2.5. Data Warehouse Ativo	10
2.5.1. Extração	11
2.5.2. Transformação	12
2.5.3. Armazenamento	13
2.6. Priorização de dados sensíveis	13
2.7. Trabalhos correlatos.....	14
2.8. Considerações finais	17
Capítulo 3 Estratégia para Extração, Transformação e Armazenamento em DW Ativo - ETA-PoCon	18
3.1. Considerações iniciais	18
3.2. Definição do problema	19
3.2.1. Frequência de atualização	20
3.2.2. Volume.....	20
3.2.3. Relevância	21

3.3. Visão geral da estratégia ETA-PoCon	21
3.3.1. Mapeamentos.....	22
3.3.2. Intervalo de atualização - T.....	23
3.3.3. Análise do volume do delta - $V(\Delta)$	24
3.3.4. Análise da relevância do delta - $R(\Delta)$	26
3.4. Ferramenta para Extração, Transformação e Armazenamento - FETA.....	28
3.4.1. Arquitetura da FETA.....	29
3.4.2. Processo de Extração.....	30
3.4.3. Processo de Transformação e Armazenamento.....	32
3.4.4. Mecanismo de disparo de transferência de dados.....	32
3.5. Considerações finais	35
Capítulo 4 Experimentos e Resultados	36
4.1. Considerações iniciais	36
4.2. Ambiente utilizado	36
4.2.1. Bases de dados	36
4.2.2. Hardware utilizado.....	38
4.3. Aplicação de política baseada em volume - $V(\Delta)$	38
4.3.1. Experimento I.....	38
4.3.2. Experimento II.....	40
4.3.3. Experimento III.....	41
4.3.4. Discussão dos resultados.....	42
4.4. Aplicação de política baseada em Relevância - $R(\Delta)$	43
4.4.1. Experimento I.....	44
4.4.2. Experimento II.....	46
4.4.3. Experimento III.....	47
4.4.4. Discussão dos resultados.....	49
4.5. Validação da política por relevância.....	50
4.5.1. Experimento I.....	52
4.5.2. Experimento II.....	53
4.5.3. Experimento III.....	54
4.5.4. Experimento IV.....	55
4.5.5. Discussão dos resultados.....	56

4.6. Considerações finais	57
Capítulo 5 Conclusões	58
5.1. Trabalhos Futuros.....	60
Referências Bibliográficas	62

Lista de Figuras

Figura 2.1 - Arquitetura simplificada de um <i>Data Warehouse</i> (Adaptado de [SAN_08])...	7
Figura 2.2 – Arquitetura DW proposta por Javed (adaptado de [JAV_10]).....	15
Figura 2.3 – Arquitetura de DW com múltiplos níveis de cache (adaptado [ZHU_08])	16
Figura 3.1 - (a) Estrutura de um DW clássico. (b) Estrutura de um DW com a estratégia a proposta	22
Figura 3.2 - Esquema de mapeamento entre uma fonte de dados e um repositório	23
Figura 3.3 - Esquema de mapeamento de uma loja de departamentos.....	24
Figura 3.4 - Gráfico que representa a evolução do $V(\Delta)$ no decorrer do tempo	26
Figura 3.5 - Esquema de representação de relevância de registros	26
Figura 3.6 - Expressão para cálculo de relevância	27
Figura 3.7 - Gráfico que representa a evolução do $V(\Delta)$ no decorrer do tempo	28
Figura 3.8 – Arquitetura da FETA.....	29
Figura 3.9 - Modelo Entidade-Relacionamento da base para armazenamento dos mapeamentos	30
Figura 3.10 - Exemplo de <i>trigger</i> utilizada no processo de extração	31
Figura 4.1 - Esquemas utilizados durante os testes	37
Figura 4.2- Gráfico da evolução do $V(\Delta)$ durante o experimento I	39
Figura 4.3 - Gráfico da evolução do $V(\Delta)$ durante o experimento II	40
Figura 4.4 - Gráfico da evolução do $V(\Delta)$ durante o experimento III.....	42
Figura 4.5 - Gráfico que demonstra o total de atualizações executadas com e sem a utilização do parâmetro $V(\Delta)$ durante os experimentos I, II e III	43
Figura 4.6 - Gráfico da evolução do $R(\Delta)$ durante o experimento I.....	44
Figura 4.7 - Gráfico da evolução do $V(\Delta)$ durante o experimento I	45
Figura 4.8 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento II	46
Figura 4.9 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento II	47
Figura 4.10 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento III... ..	48
Figura 4.11 - Gráfico da evolução do $V(\Delta)$ durante a segunda etapa do experimento III... ..	48
Figura 4.12 - Gráfico que demonstra o total de atualizações executadas com e sem a utilização do parâmetro $R(\Delta)$ durante os experimentos I, II e III.....	49
Figura 4.13 – Gráfico da evolução do $R(\Delta)$ durante o experimento I	52
Figura 4.14 – Gráfico da evolução do $R(\Delta)$ durante o experimento I com destaque para os dados sensíveis transferidos no instante 20	53

Figura 4.15 - Gráfico da evolução do $R(\Delta)$ durante o experimento II com destaque para os dados transferidos na instante 22, 34 e 40.	54
Figura 4.16- Gráfico da evolução do $R(\Delta)$ durante o experimento III com destaque para os dados transferidos na instante 27.....	55
Figura 4.17 - Gráfico da evolução do $R(\Delta)$ durante o experimento IV com destaque para os dados transferidos nos instantes 13 e 20.....	56
Figura 4.18 - Totais de atualizações executadas nos experimentos I, II, III e IV	57

Lista de Tabelas

Tabela 3.1 – Trabalhos correlatos.....	19
Tabela 3.2 – Relevância das tuplas da Tabela 1	27
Tabela 4.1 - Parâmetros utilizados no experimento I.....	38
Tabela 4.2 – Configurações do experimento I.....	39
Tabela 4.3 – Parâmetros utilizados no experimento II	40
Tabela 4.4 – Configurações do experimento II	40
Tabela 4.5 - Parâmetros utilizados no experimento III	41
Tabela 4.6 - Parâmetros utilizados no experimento I.....	44
Tabela 4.7 – Configurações do experimento I.....	44
Tabela 4.8 - Parâmetros utilizados no experimento II.....	46
Tabela 4.9 – Configurações do experimento II	46
Tabela 4.10 - Parâmetros utilizados no experimento III	47
Tabela 4.11 – Configurações do experimento I.....	48
Tabela 4.12 - Relatório das 10 empresas com maior número de acidentes.....	51
Tabela 4.13 - Parâmetros utilizados nos quatro experimentos	51
Tabela 4.14 – Configurações utilizadas nos quatro experimentos	51
Tabela 5.1 – Comparação entre o trabalho desenvolvido e o mecanismo descrito por Che [Che_10].....	60

Lista de Siglas

DW	<i>Data Warehouse</i>
DWA	<i>Data Warehouse Ativo</i>
ETA	<i>Extração, Transformação e Armazenamento</i>
SID	<i>Sistema de Integração de Dados</i>
SIVAT	<i>Sistema de Informação e Vigilância de Acidentes de Trabalho</i>

Capítulo 1 Introdução

1.1. Considerações Iniciais

O constante crescimento de organizações comerciais, bem como o aumento no número de fusões entre elas, evidencia a necessidade de integração e compartilhamento de grandes massas de dados provenientes de fontes heterogêneas e distribuídas. Nesse contexto, os Sistemas de Integração de Dados – SIDs vêm conquistando espaço significativo à medida que as corporações entendem os benefícios de unir seus dados de forma automatizada, o que permite uma análise rápida e precisa de suas informações [HAL_06].

A comunidade científica tem apresentado inúmeras abordagens a cerca da construção de SIDs e cada uma delas possui vantagens e desvantagens podendo ser melhor aplicada de acordo com as necessidades dos usuários finais. A arquitetura *Data Warehouse* - DW, cuja característica refere-se à abordagem materializada, é a que mais se destaca, pois nesse tipo de sistema os dados são extraídos das fonte e transferidos a um repositório central. Nessa arquitetura são aplicados processos de Extração, Transformação e Armazenamento – ETA, que são responsáveis por transferir os dados das fontes para o repositório e tratar os conflitos gerados pela heterogeneidade das bases [JAV_10]. Os desafios que circundam a construção de ferramentas para implementação do processo ETA, bem como as abordagens existentes para a atualização de um DW têm destaque na comunidade científica.

A grande maioria dos DW é atualizada com execuções periódicas de ferramentas de ETA. Esse tipo de abordagem é chamado de execução em modo *off-line*, visto que para transferir os dados para um repositório é necessário manter os sistemas fontes desativados [GUE_11]. A crescente necessidade de grandes corporações em diminuir o tempo entre a geração de um dado e a análise de sua informação torna a execução do processo ETA em modo *off-line* inviável e exige a criação de abordagens capazes de transferir dados em pequenos intervalos de tempo sem a necessidade de desativação dos sistemas envolvidos [VAS_09].

Diante desse panorama, surgiram os DW ativos cujo intervalo de tempo entre as execuções das ferramentas ETAs diminuem de dias para horas, ou até mesmo minutos, porém esse aumento no grau de atualização dos dados integrados afeta diretamente o processo ETA, que passa a executar com maior frequência a transferência de um conjunto de dados reduzido. Com atualizações frequentes, surge a necessidade de criação de critérios para priorização dos dados a serem transferidos, uma vez que a transferência de todo e qualquer dado inserido nas fontes sobrecarregaria os sistemas envolvidos. Com isso, as ferramentas que executam o processo ETA devem possuir estratégias que priorizem a propagação de dados considerados sensíveis [NGU_06].

1.2. Motivação e escopo

Os trabalhos encontrados na literatura relacionados ao desenvolvimento de DW ativos são, em sua maioria, voltados à construção de arquiteturas para suporte ao novo grau de atualização do DW. São desenvolvidos poucos trabalhos, técnicas e estratégias que permitam a execução do processo ETA somente quando relevante, ou seja, execução apenas sobre dados que afetem diretamente as análises executadas sobre o repositório.

Algumas estratégias adotadas na atualização de DW ativos baseiam-se apenas no aumento da frequência de execução das ferramentas ETA. Com isso, atualizações antes executadas uma vez ao dia passam a ser executadas em intervalos de alguns minutos. Com esse tipo de estratégia a frequência de atualização do DW é pré-definida e pode haver o consumo de recursos em operações sobre dados não relevantes.

Desse modo, a adoção de estratégias que não possuam uma frequência de atualização pré-definida se mostra interessante, uma vez que diminui as transferências desnecessárias e reduz o consumo de recursos dos sistemas fontes. Vale salientar que, a não existência de um intervalo de atualização pré-estabelecido exige que a estratégia possua mecanismos que decidam o momento em que o processo ETA deve ser disparado.

A criação de um mecanismo capaz de definir a frequência de atualização do DW é uma tarefa não trivial que deve contar com a análise de inúmeras variáveis, presentes tanto no DW como nas fontes de dados.

1.3. Objetivos

O trabalho consiste na construção de uma estratégia para execução do processo ETA voltada à DW ativos, denominada ETA-PoCon. A estratégia distingue-se das existentes, pois possui políticas configuráveis de propagação de informações. Para cada mapeamento efetuado entre uma base de dados fonte e o armazém de dados, o usuário tem a possibilidade de definir a melhor estratégia para a transferência dos dados, com base em parâmetros como intervalo de tempo em que a informação deve ser transferida, volume de dados que deve ser propagado a cada transferência e relevância dos dados em relação às informações contidas no repositório.

A contribuição original desse trabalho é oferecer uma estratégia que permita a redução na frequência de execução do processo ETA em DW ativos, o que mantém um grau satisfatório de atualização das informações no DW e evita a sobrecarga gerada aos sistemas envolvidos. A diminuição na frequência de atualização é baseada na transferência prioritária de dados considerados sensíveis, ou seja, dados que quando transferidos agregam valor às informações contidas no DW.

A estratégia pode ser adaptada ao ambiente em que é utilizada e espera-se que sua aplicação diminua a frequência de execução do processo ETA sobre dados não relevantes e, conseqüentemente, reduza a possibilidade de sobrecarga dos sistemas envolvidos.

1.4. Organização do Trabalho

O trabalho está organizado em cinco capítulos descritos a seguir:

- *Capítulo 2 – Conceitos fundamentais:* são descritos os conceitos básicos sobre os temas abordados no trabalho. Inicialmente uma introdução aos sistemas de integração de dados e à arquitetura DW. Em seguida, é apresentada uma breve descrição de cada um dos processos de responsabilidade das ferramentas ETA, são descritos também os DW ativos e os desafios encontrados na construção desse tipo de arquitetura. Por fim, são apresentados trabalhos correlatos.
- *Capítulo 3 – Estratégia para Extração, Transformação e Armazenamento em DW Ativo - ETA-PoCon:* é detalhada a estratégia proposta no trabalho e apresentada a

ferramenta implementada para execução dos testes e validações;

- *Capítulo 4 – Testes e resultados*: são apresentados os experimentos desenvolvidos e os resultados obtidos com a aplicação da estratégia proposta;
- *Capítulo 5 – Conclusões*: são discutidas as conclusões sobre o trabalho e sugestões de trabalhos futuros.

Capítulo 2 Conceitos Fundamentais

2.1. Considerações iniciais

Atualmente ambientes de *Data Warehouse* - DW estão presentes em grande parte das corporações e esse tipo de arquitetura permite análises profundas e seguras sobre o crescente volume de dados gerados por todos os setores e departamentos de uma empresa [JAV_10]. Essa arquitetura, apesar de bem sucedida, possui problemas relacionados ao grau de atualização das informações resultante da execução periódica dos processos de Extração, Transformação e Armazenamento – ETA.

Com a crescente demanda do mundo comercial por respostas rápidas aos eventos ocorridos em seus negócios, surge a necessidade de novas estratégias para construção de arquiteturas DW [VAS_09]. Nesse contexto, são desenvolvidas inúmeras pesquisas cujo objetivo refere-se à elaboração de abordagens de execução do processo ETA que aumentem a frequência de atualização dos DW.

Nesse capítulo são descritos os impactos causado no processo ETA convencional quando se adicionam essas novas abordagens. Para isso, inicialmente são apresentados os conceitos principais de Sistemas de Integração de Dados – SIDs e DW, a descrição de cada uma das etapas do processo ETA e, posteriormente, são apresentadas as pesquisas desenvolvidas nesse contexto.

2.2. Sistemas de Integração de Dados

O armazenamento de grandes massas de dados apresenta a necessidade de integração e compartilhamento de informações que muitas vezes estão alocadas em fontes heterogêneas e distribuídas [HAL_06]. Com o intuito de prover uma visão geral sobre as informações, muitas organizações aplicam grande parte de seus recursos humanos em processo de integração das informações [YUN_10]. Com isso, a integração de dados de forma automatizada vem se tornando uma necessidade cada vez maior.

Para a comunidade científica, a integração de dados também se mostra bastante relevante, ao passo que permite a junção de resultados de pesquisas realizadas de forma independente, o que pode trazer contribuições mais significativas a diversas áreas do conhecimento [JAR_03].

Os Sistemas de Integração de Dados – SIDs, que têm como objetivo fornecer uma visão global sobre diversas fontes de dados, permitem a recuperação de informações relevantes a um determinado contexto. Um SID é capaz de homogeneizar dados de fontes distintas independentemente da forma e estrutura nas quais esses dados estão armazenados, o que torna a construção de um sistema desse nível uma tarefa não trivial [HAL_06].

Para a construção de SIDs há duas abordagens principais: virtual e materializada. Essa classificação leva em consideração o local onde os dados estão armazenados no momento da consulta. Na abordagem virtual, os dados são retornados diretamente das fontes de dados, ou seja, no momento da consulta, os dados estão em suas bases de origem (fontes). Por outro lado, os SIDs que implementam a abordagem materializada não retornam dados diretamente das fontes. As consultas são executadas em um repositório central, cujos dados relevantes ao ambiente de integração são previamente extraídos das fontes, tratados e armazenados no repositório.

Uma das vantagens da abordagem materializada é o baixo custo computacional das consultas. Como os dados são retornados diretamente de um repositório sem tratamentos em tempo de execução, o tempo de resposta é consideravelmente menor. Além disso, os dados armazenados em um repositório permitem uma série de análises diferenciadas como a aplicação de técnicas de prospecção de dados. No entanto, uma vez que os dados são extraídos das fontes, tratados e inseridos no repositório, as informações podem se tornar desatualizadas com o passar do tempo e exigir atualizações periódicas, o que caracteriza uma das maiores desvantagens dessa abordagem [ZHE_09]. A principal arquitetura que implementa essa abordagem é a arquitetura de *Data Warehouse*. Sistemas desse tipo são

amplamente utilizados por grandes corporações, já que permitem uma análise apurada e fiel de diversas fontes de dados. Na próxima seção são apresentados os detalhes da arquitetura DW.

2.3. Arquitetura Data Warehouse

As arquiteturas de *Data Warehouse* – DW são vastamente aplicadas no gerenciamento e na organização de grandes massas de dados, já que foram elaboradas de forma a facilitar a extração de relatórios e análises avançadas. Um DW pode ser entendido como um repositório de dados integrados, não volátil, que objetiva o apoio à decisão em grandes corporações [JAV_10] [VIQ_11] [XU_11a]. Em outras palavras, DW é um ambiente bastante abrangente que inclui processos, ferramentas e tecnologias necessárias para extrair dados de sistemas fonte e armazená-los em dispositivos específicos fornecendo aos usuários finais análises aprofundadas de todo o negócio [NGU_06].

Desse modo, a arquitetura de DW conta com uma base de dados central, cujos dados das fontes são extraídos, tratados e armazenados no repositório. Na Figura 2.1 é apresentado o esquema simplificado de um DW. Sua arquitetura tem um custo computacional relativamente baixo a cada consulta; por outro lado, há a constante necessidade de atualizações, já que com o passar do tempo os dados transferidos ao repositório se tornam desatualizados.

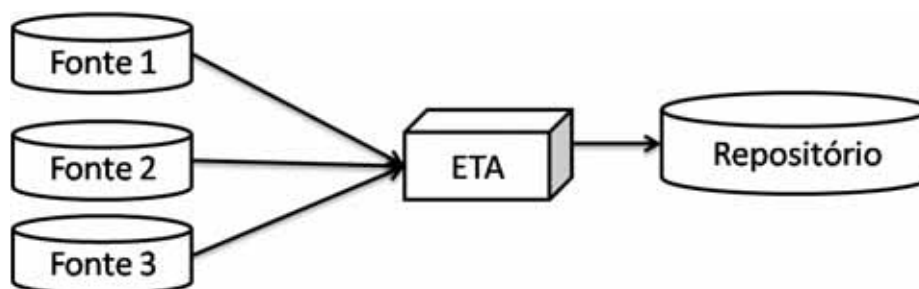


Figura 2.1 - Arquitetura simplificada de um *Data Warehouse* (Adaptado de [SAN_08])

Devido às demandas de cada organização, a construção de um DW deve passar pela análise de pessoas que possuam conhecimento do escopo dos dados que serão integrados. Desse modo, inúmeros trabalhos são desenvolvidos com intuito de propor arquiteturas específicas para a construção de DW em diferentes setores tecnológicos [QIA_09] [ZHE_09].

2.4. Extração, Transformação e Armazenamento - ETA

Na Figura 2.1, é ilustrado o processo de Extração, Transformação e Armazenamento – ETA, que pode ser considerado como a chave para extrair dados de fontes distintas e transferi-los de forma homogênea para um repositório central [JAV_10].

A construção e manutenção de ferramentas responsáveis pela execução do processo ETA é parte fundamental para o sucesso na construção de um DW. Em alguns casos, o desenvolvimento dessas ferramentas chega a consumir 80% dos recursos aplicados no projeto de DW [LUJ_04]. A seguir são descritos os processos de extração, transformação e armazenamento.

2.4.1. *Extração*

A princípio, a extração pode ser considerada a tarefa mais simples dentre as executadas por uma ferramenta ETA. O objetivo desse processo é identificar e extrair conjuntos de dados alterados em cada uma das fontes. Geralmente possui como principal entrave o grande volume de dados a ser manipulado. Outro problema a ser considerado é o fato do processo de extração sempre exigir uma interação com a fonte de dados [SAN_08].

Basicamente há duas principais abordagens quanto ao processo de extração dos dados da fonte. A primeira delas sugere a extração de todos os dados a cada execução da ferramenta ETA. No entanto, tal abordagem se mostra ineficiente para a manipulação de grandes volumes de dados. Já a segunda abordagem realiza a extração apenas dos dados alterados desde a última atualização do repositório. A implementação dessa abordagem não é uma tarefa simples, podendo fazer uso de *triggers* responsáveis por armazenar as modificações ou ainda arquivos de *log*, em que são registrados os comandos de inserção e alteração das informações.

2.4.2. *Transformação*

Após a extração, os dados são transferidos para o processo de homogeneização. Nessa etapa, os dados são transformados e, em alguns casos, limpos. O processo de transformação pode ser considerado o ponto principal de uma ferramenta ETA, já que é nesse processo que diversos conflitos existentes entre as fontes de dados e o repositório central são solucionados ou tratados.

O grau de complexidade do processo de transformação está diretamente relacionado aos tipos de conflitos existentes entre as fontes e o repositório central. Os sistemas que

alimentam as fontes de dados podem ser construídos de forma totalmente independente, e, por isso, podem-se ocasionar conflitos de modelo, sintáticos e lógicos [SCA_09]. Ao longo dos anos, inúmeros trabalhos vêm sendo desenvolvidos com foco no tratamento desses possíveis conflitos [YUN_10] [SAL_08].

2.4.3. *Armazenamento*

O armazenamento dos dados é o processo final da execução de uma ferramenta ETA. Após os dados serem homogeneizados, os mesmos devem ser inseridos no repositório central. Assim como o processo de extração a princípio, o armazenamento pode ser analisado como um processo trivial, mas possui um entrave na manipulação de grandes volumes de dados. Outro desafio do processo de armazenado é o fato de que, em alguns casos, deve ser capaz de organizar a ordem nas quais as informações devem ser inseridas, uma vez que podem existir dependências entre elas.

2.4.4. *Execução do processo ETA*

O processo ETA funciona como porta de entrada para a alimentação de um DW. Em ambientes de DW convencionais, esse processo é executado periodicamente, na maioria dos casos com intervalo de tempo de um dia. Nesse caso, os dados são extraídos, transformados e armazenados em lotes. Esse tipo de abordagem de transferência dos dados pode ser chamado de *off-line*, já que enquanto o processo ETA é executado, as aplicações que alimentam as fontes de dados devem estar desativadas a fim de evitar inconsistência nos dados a serem migrados [GUE_11] [SUN_12] [XU_11b][JAV_10][SIM_10]. Um dos grandes problemas encontrados em DW convencionais é o grau de atualização do repositório, pois, uma vez que as ferramentas ETA são executadas periodicamente, os dados integrados ficam desatualizados por um determinado período de tempo [VAS_09].

O atraso existente entre o surgimento de um dado em um sistema fonte e a ação tomada como consequência da análise dessa informação é chamado de latência de reação. Quanto maior a latência de reação de uma corporação, menor o valor de suas informações, ou seja, quanto mais uma corporação demora a reagir aos eventos de suas fontes de dados, menos competitiva ela se torna [NGU_06]. Com isso, surge a necessidade de análise de informações com grau de atualização cada vez maior, o que impulsiona o surgimento de novas abordagens quanto à execução do processo ETA [SON_10][VAS_09].

Além disso, a execução de ferramentas ETA no modo *off-line* apresenta um problema relacionado ao tempo de execução, em que muitas vezes os lotes de dados a serem

transferidos possuem um volume grande e, atrelado aos processos de transformação pode elevar o tempo de execução ao ponto de se tornar inviável [JAV_10]. Vale ressaltar que, no modo *off-line*, as aplicações fontes de dados devem estar desativadas, fator de alto impacto em corporações que possuem atividades que não podem ser cessadas por longos períodos.

2.5. *Data Warehouse Ativo*

A crescente necessidade de análises sobre informações atuais tem exigido a criação de novas abordagens a cerca do modo de atualização de DW [SHI_09]. Evidencia-se que os atrasos gerados por atualizações em modo *off-line* criam ambientes que não mais suportam as necessidades do mundo comercial [VAS_09].

Uma das possíveis soluções para esse novo grau de atualização exigido pelos usuários é a diminuição do intervalo de tempo entre as execuções de ferramentas ETA. Essa nova abordagem pode ser chamada de atualização em semi tempo-real (“*Near real-time*”), ou atualização em *microbatch* [KIM_04]. Desse modo, a latência entre a geração de uma informação e tomada de decisão é diminuída.

Nesse contexto, as arquiteturas DW estão em evolução, a carga de dados a ser transferida a cada atualização é reduzida e a frequência de atualização é aumentada. A comunidade científica ainda não apresentou uma abordagem completa e bem definida para a construção de DW com atualizações em semi tempo-real e por isso é possível encontrar diferentes denominações para esse tipo de ambiente, tais como: *Data Warehouse* com Latência Zero [NGU_06], *Real-Time Data Warehouse* [ZHU_08], *Semi Real-Time Data Warehouse* [THO_10] [CHE_10] [BOR_11] e *Data Warehouse Ativo* [VAS_09] [FAN_12] [BRO_02], sendo que o último será a denominação adotada nesse trabalho.

Em *Data Warehouse Ativos* – DWAs, o grau de desatualização do repositório diminui de dias, para horas, ou até mesmo minutos. Construir ferramentas capazes de executar o processo ETA em curtos períodos de tempo não é uma tarefa simples, tanto no que diz respeito à arquitetura quanto a algoritmos. Os desafios tecnológicos a cerca da construção de um DW com atualizações em semi tempo-real são muitos, uma vez que a implementação de uma abordagem de atualização, diferente da execução em modo *off-line*, acarreta em inúmeros novos requisitos, que afetam toda a arquitetura de um sistema de integração. A seguir, é apresentada uma análise dos impactos causados em cada uma das etapas do processo ETA [VAS_09].

2.5.1. *Extração*

Em geral, a extração de dados é a etapa em que há interação com os sistemas fontes, ou seja, os sistemas que efetuam transações com ambientes externos e geram informações. Dentre os principais pontos críticos dessa etapa, valem ser citados:

- **Mínimo de alteração no desempenho dos sistemas fontes** – Esses sistemas são responsáveis por gerar informações a partir de transações com ambientes externos (clientes, fornecedores e etc.). A sobrecarga das fontes de dados pode afetar diretamente o desempenho na execução de suas tarefas prioritárias, impactando assim todo o sistema de integração e, por esse motivo, a construção de um mecanismo de extração de dados deve levar em consideração questões como o uso de CPU e memória. Vale ressaltar que, diferentemente do modo *off-line*, uma ferramenta ETA com atualização em tempo-real executaria o processo de extração inúmeras vezes ao dia;
- **Impossibilidade de alteração nos sistemas fontes** – Na grande maioria dos casos, trata-se de sistemas legados e que não podem ser alterados livremente. O mecanismo de extração deve possuir uma arquitetura capaz de identificar alterações nos dados e extraí-los sem a necessidade de alterações nas configurações dos sistemas fontes;
- **Integridade dos dados** – O módulo de extração dos dados deve ser capaz de manter a integridade das informações transferidas ao repositório. As alterações nas fontes devem ser propagadas ao DW apenas uma vez, sendo indispensável que todas as alterações sejam transformadas e transferidas. Diferentemente do modo *off-line*, os dados alterados em uma das fontes podem não ser transferidos em apenas uma execução da ferramenta ETA.

As atuais técnicas existentes no processo de extração em modo *off-line* não são satisfatórias quando aplicadas em ambientes com atualizações em curtos períodos de tempo (semi tempo-real). Por outro lado, algumas dessas técnicas se destacam e, com algumas adaptações, podem ser consideradas boas alternativas à implementação de mecanismos de extração que atendam aos requisitos necessários. Algumas dessas técnicas promissoras são:

- *Log Sniffing* – Trata-se do uso de arquivos de *logs* para comparações e identificações de mudanças. Embora ineficiente em sistemas em que não há registros de alterações na forma de *logs*, pode ser considerada uma solução razoável, já que a implementação afetaria minimamente os sistemas fontes;

- *Triggers* – O uso de *triggers*, que são disparadas a cada alteração na fonte, também pode ser considerado uma abordagem razoável, pois podem registrar as alterações por meio de tabelas específicas ou arquivos que seriam posteriormente analisados pelo mecanismo de extração. Apesar dessa abordagem ser aplicável apenas em sistemas em que há a presença de Sistemas Gerenciadores de Banco de Dados – SGBDs relacionais, trata-se de uma abordagem promissora à medida que pode ser implementada sem grandes impactos aos sistemas fontes;

Em resumo, os problemas relacionados à extração dos dados referem-se à construção de mecanismos capazes de efetuar o processo de extração sem exigir alterações nos sistemas fontes, e o mais importante, sem impactar significativamente a eficiência dos sistemas fontes. A seguir, são apresentados os desafios relacionados à fase de transformação.

2.5.2. Transformação

A etapa de transformação dos dados pode ser considerada como o principal processo de responsabilidade de ferramentas ETA. Nessa fase, os dados extraídos das diversas fontes são transformados e adaptados ao esquema do repositório. Apesar disso, os impactos causados pela adoção de uma estratégia de atualização em tempo-real não afetam diretamente as tecnologias adotadas no processo de transformação. De um modo geral, os impactos causados a essa etapa são em sua maioria relacionados às questões de desempenho. A seguir são descritas as principais diferenças entre um processo de transformação executado por uma ferramenta ETA convencional e uma que executa atualizações em tempo-real:

- Em ferramentas de tempo-real, a frequência de atualização é significativamente maior. Por outro lado, o volume de dados a ser processado, assim como o tempo de processamento, são significativamente menores. Desse modo, as ferramentas de transformação devem ter suas estruturas alteradas para se adaptarem a esses novos requisitos.
- O pequeno volume de dados a ser processado a cada iteração permite que ferramentas ETA com atualizações em tempo-real executem grande parte das operações de transformação em dados alocados na memória principal. Com isso, os problemas que exigem alocação de memória ganham destaque.

Vale ressaltar que, de maneira resumida, a principal diferença entre uma ferramenta ETA com atualizações em tempo-real e uma ferramenta convencional é que a primeira é focada em atingir alto desempenho a fim de alcançar altas taxas de transferências de dados, enquanto a segunda é focada em transformar todo o conjunto de dados extraídos em uma janela de tempo pré-estabelecida. Nesse contexto, as tecnologias empregadas nas operações de transformações e tratamento de conflitos não são afetadas e as adaptações são restritas a questões de desempenho e estruturas de dados. A seguir, são apresentados os desafios que circundam o processo de armazenamento.

2.5.3. *Armazenamento*

O processo de armazenamento é a fase final do processo ETA, em que os dados transformando são inseridos em um repositório. Ferramentas ETA convencionais possuem mecanismos específicos para a execução desse processo, comumente denominados “*loaders*”, que são construídos para efetuar a inserção dos dados em um ambiente *off-line*.

Na maioria dos casos, a estratégia adotada nos *loaders* consiste em remover todo o conteúdo do repositório e, logo após, reinseri-lo juntamente com os dados novos. Essa estratégia ganhou destaque à medida que se percebeu que remover dados e reinseri-los juntamente com seus índices pode ser menos custoso que manter os dados com atualizações de forma incremental.

Esse tipo de estratégia não pode ser adotado em ferramentas ETA com atualizações em tempo-real, cujo processo de inserção dos dados deve ser executado enquanto o repositório está ativo. Nesse caso, não há possibilidade de remover dados e inseri-los novamente, uma vez que os usuários finais do DW estarão executando consultas normalmente enquanto os novos dados são inseridos. Surge então o desafio de se criar estratégias de atualizações que não exigem que o repositório se mantenha *off-line* durante a inserção dos dados.

2.6. *Priorização de dados sensíveis*

Como mencionado anteriormente, a construção de DWA envolve alterações em toda a arquitetura de um DW convencional. Dentre os vários novos requisitos adicionados ao ambiente, destacam-se [VAS_09]:

- como o processo de migração dos dados das fontes para o repositório deixa de ser executado em modo *off-line*, as ferramentas ETA passam a disputar recursos com as atividades prioritárias dos sistemas envolvidos;
- há um aumento considerável na frequência de execução do processo ETA.

Desse modo, é possível identificar que o processo ETA executado em um DWA deve possuir estratégias capazes de encontrar um equilíbrio entre a frequência de execução do processo ETA e o impacto causado em todos os sistemas envolvidos. Em outras palavras, um DWA deve possuir estratégias para propagação de informações que não sobrecarreguem os sistemas fontes e mantenham um grau satisfatório de atualização dos dados no repositório.

Uma possível estratégia para evitar a sobrecarga dos sistemas envolvidos é a priorização de dados considerados sensíveis, ou seja, dados que quando transferidos afetarão diretamente os resultados e análises extraídas do DW. Construir um mecanismo capaz de identificar o grau de sensibilidade das informações não é uma tarefa fácil, uma vez que esse grau está diretamente relacionado com inúmeros fatores que vão desde a frequência de consultas ao DW até o grau de relevância das alterações efetuadas nos sistemas fontes. Portanto, a elaboração de estratégias para atualizações de DWA implicam no monitoramento desses inúmeros fatores [CHE_10] [NGU_06].

2.7. Trabalhos correlatos

Os trabalhos encontrados na literatura relacionados à construção de DWA são bastante abrangentes. A comunidade científica ainda não apresentou uma abordagem completa para a construção desse tipo de ambiente, parte dos trabalhos são propostas de arquiteturas DW que suportem um alto grau de atualização [JAV_10] [THO_10], enquanto outros focam em estratégias para a atualização do DW sem fortes destaques às alterações na arquitetura [CHE_10] [SAN_09].

Nguyen [NGU_09] apresenta um levantamento do estado da arte relacionado à DW com latência zero. O autor assume que apesar das atualizações em modo *off-line* ainda suportarem muitas organizações, os DW estão ampliando suas capacidades para suportar não somente decisões estratégicas, mas também os processos operacionais das organizações. Nesse contexto, são descritos cinco fases da evolução da arquitetura DW em que é possível verificar que o surgimento das novas abordagens para atualização de DW permitirá a diminuição na latência entre os eventos ocorridos nos negócios e as ações tomadas como consequência.

Vassiliadis [VAS_09] resume todos os problemas e desafios relacionados à implementação de ferramentas para execução do processo ETA para DW com atualizações em tempo-real. O autor parte dos problemas presentes em ferramentas ETA convencionais para identificar os desafios tecnológicos presentes em cada uma das etapas de um processo ETA de tempo-real, mencionados nas seções anteriores deste trabalho. Além disso, são apresentadas algumas arquiteturas para DW de tempo-real e uma análise detalhada do estado da arte a cerca de ferramentas ETA.

Em [JAV_10] é apresentada uma proposta de arquitetura DW que une a abordagem convencional com a abordagem de atualizações em tempo-real. Na Figura 2.2 é demonstrada a arquitetura proposta cujos dados que devem ser transferidos em tempo-real são extraídos das fontes e armazenados no arquivo estruturado “N_XML_FILE”, enquanto os dados que devem ser transferidos apenas em modo *off-line* são armazenados no arquivo “R_XML_FILE”.

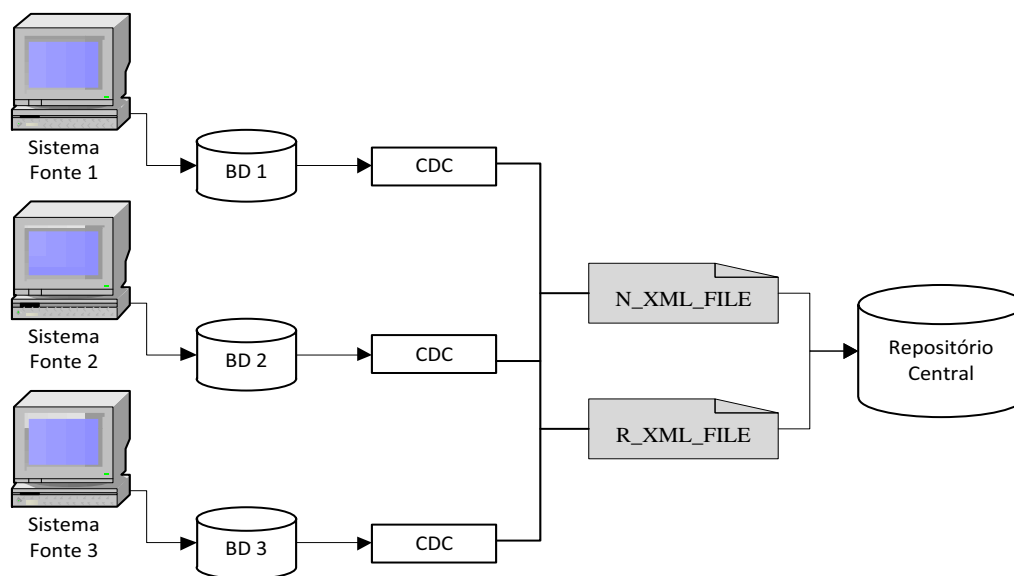


Figura 2.2 – Arquitetura DW proposta por Javed (adaptado de [JAV_10])

Com essa abordagem o sistema consegue atingir um maior grau de atualização do DW, além de permitir que o processo ETA em modo *off-line* seja executado em menor tempo, uma vez que parte dos dados já foi transferida em tempo-real. Apesar do trabalho contribuir com uma arquitetura que se diferencia das existentes, mesclando a abordagem convencional com a abordagem de tempo-real, a estratégia não possui nenhum tipo de tratamento para evitar a sobrecarga dos sistemas envolvidos. Com isso, o volume de dados a ser transferido pode se expandir ao ponto de afetar o desempenho de todo o ambiente.

Outra arquitetura é proposta em [ZHU_08] que contempla o processo de extração executado por um *web service* e a transferência dos dados é feita com uso de arquivos XML. O foco do trabalho é a criação de uma estrutura formada por vários níveis de *cache*, cada um contendo dados alterados em um certo período e, considerando que a cada dez minutos os dados são extraídos das fontes, os níveis de *cache* 1, 2 e 3 conterão dados com atrasos de 10, 20 e 30 minutos respectivamente.

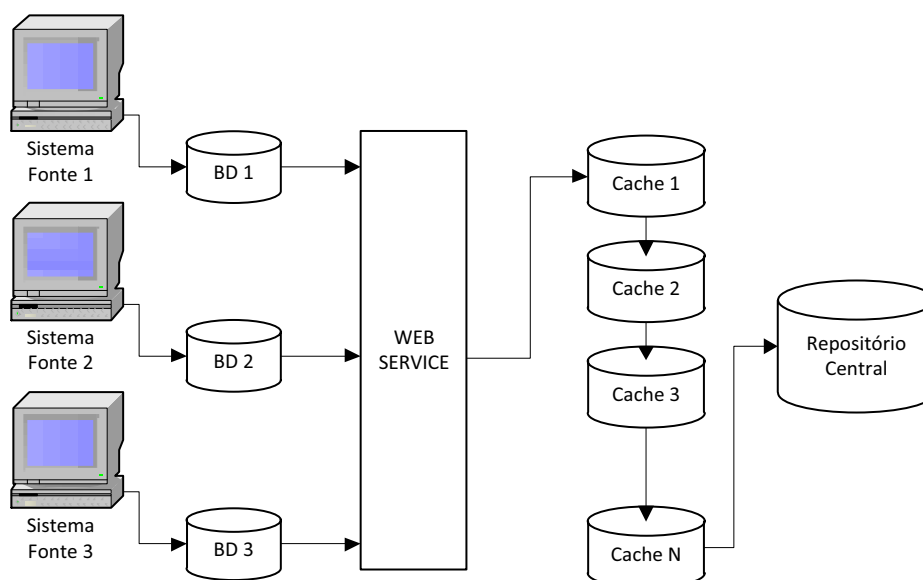


Figura 2.3 – Arquitetura de DW com múltiplos níveis de cache (adaptado [ZHU_08])

Segundo o autor, essa estrutura facilita a implementação de ferramentas ETA de tempo-real, além de facilitar a consulta aos dados por meio dos vários níveis de *cache*. Porém, nenhum experimento é apresentado para comprovar a eficiência da estrutura proposta.

Diferentemente dos trabalhos citados anteriormente, [THO_10] apresenta um estudo focado na etapa de armazenamento de DWA. O fato de repositório ser atualizado em tempo-real pode exigir a adoção de uma estratégia de inserção dos dados de forma incremental, ou seja, não há possibilidade de se excluir todos os dados e recriar o repositório, como é feito em algumas ferramentas ETA convencionais. A inserção dos dados de forma incremental pode ocasionar algumas inconsistências, uma vez que, enquanto os dados são transferidos para o repositório, os sistemas fontes se mantêm ativos. O trabalho é focado na apresentação de várias abordagens para o tratamento dessas inconsistências, juntamente com as vantagens e desvantagens de cada uma.

O trabalho apresentado em [CHE_10] consiste na elaboração de um mecanismo de atualização de DWA que efetua a análise de alguns parâmetros para definir a frequência de atualização do repositório. Em resumo, o mecanismo monitora os estados das fontes de

dados e o repositório, analisando o impacto causado caso o repositório seja atualizado, o número de registros que seriam afetados e a frequência na qual os dados são consultados. A partir da combinação desses três fatores, o mecanismo é capaz de definir quando a atualização deve ser efetuada em tempo-real ou em semi tempo-real (pequenos intervalos de tempo). O autor afirma que com a aplicação desse mecanismo é possível melhorar o custo de processamento das transferências de dados sem diminuir o grau de atualização do DW.

O mecanismo descrito em [CHE_10] possui um problema relacionado ao volume de dados a ser processado, pois a aferição do impacto que uma atualização pode causar no repositório muitas vezes pode contar com consultas a volumes de dados expressivos, tornando os cálculos executados pelo mecanismo mais custoso que a própria transferência dos dados. Outro ponto negativo do mecanismo é que a aferição do impacto da atualização leva em consideração o conteúdo do repositório, sem considerar a estrutura do mesmo, o que faz com que a adaptação do mecanismo em diferentes contextos seja custosa. Além disso, o mecanismo é especializado à DW construídos sobre o modelo dimensional, o que impede a aplicação dele em DW de outros modelos.

2.8. Considerações finais

Neste capítulo, foram apresentados os principais conceitos referentes aos Sistemas de Integração de Dados. Em seguida, a arquitetura DW foi descrita e o processo de Extração, Transformação e Armazenamento foi detalhado com ênfase no problema gerado pela execução desse processo em modo *off-line*. É possível identificar que a atual necessidade de grandes corporações em efetuar análise sobre dados com alto grau de atualização evidencia os problemas existentes na atualização periódica dos repositórios.

Foi também apresentada uma análise dos novos requisitos adicionados ao processo ETA quando aplicado em arquiteturas de DWA, além de descreveras alterações exigidas em cada uma das fases do processo. As ferramentas ETA devem possuir mecanismos capazes de encontrar um ponto de equilíbrio entre o grau de atualização e o consumo de recursos dos sistemas envolvidos, o que evidencia a importância da priorização de dados sensíveis na busca por um ponto de equilíbrio.

Por fim, foram descritos trabalhos encontrados na literatura que estão diretamente relacionados com os conceitos apresentados em todo o capítulo. Identifica-se que os trabalhos desenvolvidos recentemente são abrangentes e incluem propostas de arquitetura para novos ambientes de DW, além de estratégias de priorização de dados sensíveis.

Capítulo 3 Estratégia para Extração, Transformação e Armazenamento em DW Ativo - ETA-PoCon

3.1. Considerações iniciais

Como descrito no capítulo anterior Sistemas de Integração de Dados, em particular os ambientes de DW, surgem como auxílio aos negócios de grandes corporações, uma vez que permitem análises aprofundadas sobre dados gerados por fontes de informações distintas e heterogêneas. A necessidade de análises sobre dados atualizados impulsiona os estudos que contemplam novas abordagens para o processo de atualizações dos DW. Nesse contexto surgem novas propostas para execução do processo ETA e novas categorias de DW, dentre elas os DWA.

Em ambientes com essa nova abordagem, a frequência de execução do processo ETA é aumentada, e os sistemas fontes não podem ser desativados durante a transferência dos dados. Conseqüentemente, o equilíbrio entre o grau de atualização do DW e o consumo dos recursos dos sistemas envolvidos surge como um novo requisito, isso devido à frequência de atualização do DW que pode sobrecarregar os sistemas fontes e afetar suas tarefas prioritárias.

Nesse capítulo é apresentada a estratégia denominada ETA-PoCon, que se resume em uma estratégia para execução do processo ETA com políticas configuráveis de propagação dos dados que permitem ao usuário configurar o processo de modo a executar apenas transferências consideradas relevantes. Com essa abordagem, espera-se diminuir a frequência de execução do processo ETA sem perdas no grau de atualização do DW.

3.2. Definição do problema

Os trabalhos encontrados na literatura relacionados ao desenvolvimento de DWA são, em sua maioria, voltados à construção de arquiteturas para suporte ao novo grau de atualização do DW. Como demonstrado na Tabela 3.1, apesar de alguns trabalhos mencionarem o problema da sobrecarga sobre os sistemas envolvidos, poucos desenvolvem técnicas e estratégias que permitam a execução do processo ETA somente quando relevante, ou seja, execução do processo ETA apenas sobre dados que afetem diretamente as análises executadas sobre o repositório.

Tabela 3.1 – Trabalhos correlatos

<i>Trabalho</i>	<i>Menciona problema relacionado à sobrecarga</i>	<i>Apresenta solução</i>
Nguyen [NGU_06]	Não	Não
Javed [JAV_10]	Sim	Não
Vassiliadis [VAS_09]	Sim	Não
Zhu [ZHU_08]	Sim	Não
Che [CHE_10]	Sim	Sim

Algumas estratégias adotadas na atualização de DWA baseiam-se apenas no aumento da frequência de execução das ferramentas ETA, com isso as atualizações antes executadas uma vez ao dia passam a ser executadas em intervalos de alguns minutos ou segundos. Nesse tipo de estratégia, a frequência de atualização do DW é pré-definida e pode haver o consumo de recursos em operações sobre dados não relevantes às análises executadas sobre o repositório.

Para exemplificar, considere que uma loja de departamentos constituída por inúmeras unidades mantém um DWA para análise sobre seus dados de vendas. Suponha que essa companhia deseja analisar apenas dados de unidades que alcançaram um volume de 100 vendas por hora. Caso a atualização do DW adote a abordagem descrita anteriormente, o processo ETA seria executado em intervalos de tempo pré-definido sem análise sobre o volume de vendas das unidades. Dessa forma, seriam executadas operações sobre dados não relevantes (vendas de unidades com volume menor que 100 vendas/hora) e conseqüentemente recursos seriam consumidos desnecessariamente.

Outras estratégias menos otimizadas quanto à frequência de atualização, adotam políticas que transferem os dados instantes após a inserção dos mesmos nos sistemas fontes – o que agrava ainda mais o problema de consumo desnecessário de recurso.

Desse modo, a adoção de estratégias que não possuam uma frequência de atualização pré-definida se mostra relevante, uma vez que diminui as transferências desnecessárias e reduz o consumo de recursos dos sistemas fontes. Cabe salientar que, a não existência de um intervalo de atualização pré-estabelecido exige que a estratégia possua mecanismos que decidam o momento em que o processo ETA deve ser disparado. A criação de um mecanismo capaz de definir a frequência de atualização do DW é uma tarefa não trivial que deve contar com a análise de inúmeras variáveis. A seguir, são descritas algumas dessas variáveis.

3.2.1. *Frequência de atualização*

O intervalo de execução do processo ETA está relacionado ao grau de atualização das informações contidas no repositório, as estratégias adotadas para atualização de DWA possuem frequências de atualização pré-definidas. Mecanismos de disparo do processo ETA devem possuir estruturas capazes de permitir diferentes frequências de atualizações, ou seja, cada mapeamento entre uma fonte de dados e o repositório deve possuir um intervalo específico de atualização. Assim, é possível adaptar o processo ETA à necessidade do negócio. Em resumo, dados com maior grau de importância devem possuir maior frequência de atualização.

3.2.2. *Volume*

Diferentemente dos DW convencionais, em que, a cada atualização do repositório, todos os dados das fontes são extraídos e tratados, no processo ETA executado em DWA apenas os dados modificados após a última atualização devem ser transferidos.

Geralmente, esse conjunto de dados a ser transferido é denominado delta (Δ), e a quantidade de tuplas contida nesse conjunto é denominada volume do delta ($V(\Delta)$) – em linhas gerais o delta é o conjunto de tuplas alteradas desde a última execução do processo ETA e deve ser extraído de cada uma das fontes.

A análise do $V(\Delta)$ é importante ao mecanismo que define a frequência de atualização, visto que o Δ representa o conjunto de tuplas desatualizadas no DW. Dessa forma o $V(\Delta)$ é diretamente proporcional ao grau de desatualização do repositório – quanto maior a quantidade de tuplas no delta, maior será o impacto causado no repositório após a execução do processo ETA.

3.2.3. Relevância

Outro fator a ser considerado é a relevância do delta ($R(\Delta)$). Tuplas alteradas nas fontes podem ocasionar diferentes impactos no repositório, pois uma tupla pode ser mais ou menos relevante que outra. A análise da relevância é bastante complexa e pode envolver questões semânticas do repositório e das fontes.

Por outro lado, a análise da relevância das tuplas pode ser realizada por meio da estrutura do banco de dados, no caso de DW estruturados em SGBDs relacionais. Nesse caso, assume-se que o grau de relevância de uma tupla é diretamente proporcional à quantidade de tuplas que fazem referência a ela, ou seja, quanto maior o número de tuplas que se relacionam com a tupla alterada, maior sua relevância.

3.3. Visão geral da estratégia ETA-PoCon

A análise apresentada na seção anterior evidencia a necessidade de criação de uma estratégia para execução do processo ETA capaz de definir a frequência de atualização do DW com base em análises sobre as variáveis envolvidas. A ETA-PoCon consiste na criação de uma estratégia de atualização com foco na elaboração de um mecanismo que controle o disparo da transferência dos dados.

Esse mecanismo possui alguns parâmetros de entrada configurados pelo usuário e efetua análise sobre as variáveis mencionadas anteriormente (volume e relevância) para definir a frequência de atualização do DW. Além disso, a estratégia adota uma abordagem em que é possível determinar intervalos de atualização específicos para cada mapeamento entre as fontes de dados e o repositório.

Na Figura 3.1a é apresentado o esquema clássico de um DW, enquanto na Figura 3.1b é apresentado o esquema de um DW com a utilização da estratégia ETA-PoCon, em que é possível identificar a presença do mecanismo controlador envolvendo os deltas de cada uma das fontes, o processo de Transformação e Armazenamento além do próprio repositório. Essa representação é feita com intuito de demonstrar que o mecanismo possui interação direta com os deltas (para aferição do volume e relevância) e com o repositório (para aferição da relevância). Além disso, o mecanismo é responsável por efetuar o disparo das etapas de Transformação e Armazenamento, enquanto a etapa de Extração é executada continuamente. A seguir, são definidos alguns conceitos fundamentais para o entendimento da estratégia.

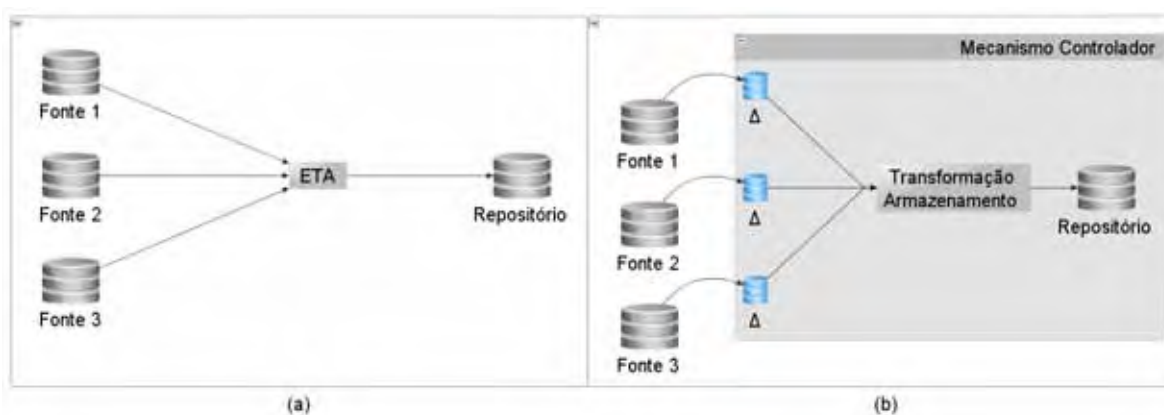


Figura 3.1 - (a) Estrutura de um DW clássico. (b) Estrutura de um DW com a estratégia a proposta

3.3.1. Mapeamentos

Um mapeamento é uma estrutura que define as operações que devem ser aplicadas sobre os dados de uma ou mais tabelas pertencentes à fonte de dados, além de definir a tabela do repositório onde os dados devem ser inseridos. É por meio dos mapeamentos que os administradores do DW definem quais dados devem ser transferidos, quais operações devem ser executadas e onde os dados devem ser armazenados. Na Figura 3.2 é apresentado o esquema simplificado de dois mapeamentos. É possível verificar que o mapeamento pode ser realizado utilizando uma única tabela fonte – *Mapeamento 1* – ou ainda utilizando duas ou mais tabelas fonte – *Mapeamento 2*. Em ambos os casos, define-se quais operações devem ser aplicadas sobre os dados.

Mapeamentos entre bases de dados relacionais são objetos de estudo bastante discutidos na comunidade científica, porém, neste trabalho são tratados de forma simplificada e indicam apenas onde os dados de uma ou mais tabelas fonte devem ser inseridos no repositório. Essa simplificação do termo é possível uma vez que o trabalho é

focado em questões de disparos do processo ETA e não nas metodologias utilizadas nas tarefas contidas nesse processo.

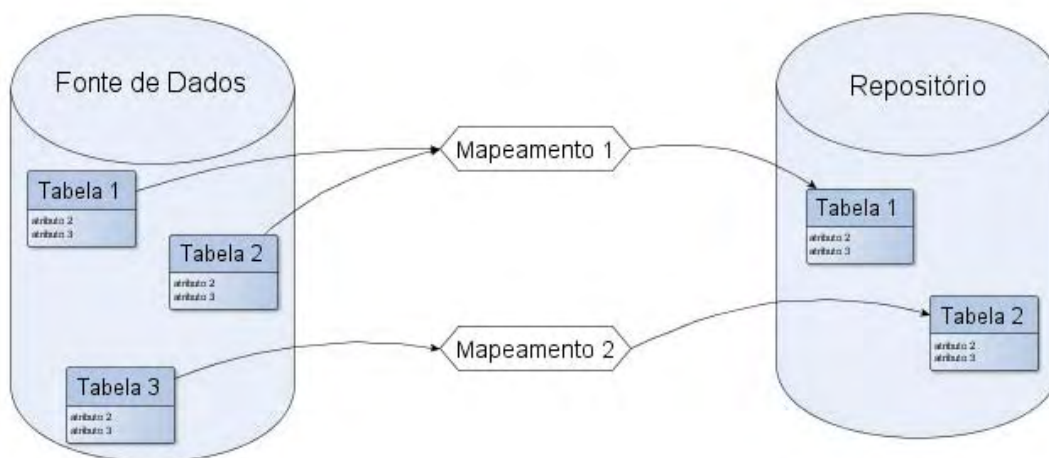


Figura 3.2 - Esquema de mapeamento entre uma fonte de dados e um repositório

3.3.2. *Intervalo de atualização - T*

A grande maioria dos negócios possuem dados com diferentes níveis de necessidade de atualização. Desse modo, a estratégia proposta deve tratar cada mapeamento de forma individual, e um dos parâmetros definidos pelo usuário é o intervalo de tempo no qual os dados de cada mapeamento devem ser transferidos. Portanto, define-se como T_i o intervalo de tempo no qual os dados do mapeamento i devem ser transferidos para o repositório.

Para exemplificar o funcionamento do mecanismo quanto ao intervalo de atualização individual a cada mapeamento, considere que uma grande companhia, formada por inúmeras lojas de departamento, mantém um DWA para efetuar análises sobre seus dados de vendas. Considere também, que a base de dados de uma de suas lojas (“Loja 1”) e a base de dados do repositório estão representadas na Figura 3.3.

No esquema da base da “Loja 1”, a tabela “Produto” armazena informações de cada um dos produtos contidos na loja, enquanto a tabela “Clientes” é composta por informações referentes aos clientes que já efetuaram alguma compra na loja. Por fim, a tabela “Venda” representa a relação do produto com o cliente. No DW, são efetuadas apenas análises sobre as informações de vendas e clientes, assim as únicas tabelas presentes no esquema do repositório são “Venda” e “Cliente”. No mapeamento 1 são definidas as operações que devem ser aplicadas nos dados de venda, enquanto o mapeamento 2 são definidas as operações sobre os dados de clientes.

As consultas executadas sobre o DW referentes a dados de vendas são fundamentais à tomada de decisões de curto prazo, como promoções relâmpago e anúncios de produtos específicos dentro da própria loja. Por outro lado, as consultas executadas referentes aos dados de clientes são utilizadas para análise de perfil, e pode ser executada uma vez ao dia.

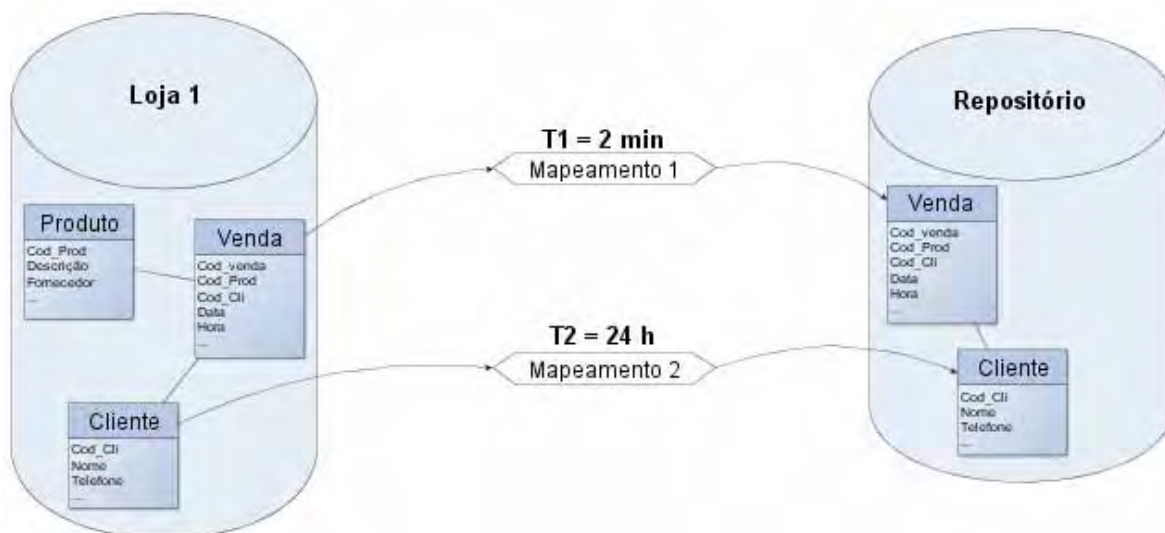


Figura 3.3 - Esquema de mapeamento de uma loja de departamentos

No cenário apresentado, fica clara a necessidade de um mecanismo que permita a definição de diferentes frequências de atualização para os dados de vendas e clientes. Nesse caso, na aplicação do mecanismo proposto o usuário definiria T1 em dois minutos e T2 em 24 horas, o mecanismo então dispararia a transferência dos dados referentes ao mapeamento 1 a cada dois minutos e o processo referente ao mapeamento 2 apenas uma vez ao dia. Vale ressaltar que no cenário utilizado no exemplo, o mapeamento 1 possui uma frequência muito maior que o mapeamento 2. Porém, em aplicações mais complexas pode ocorrer de diferentes mapeamentos possuírem frequências de atualização mais próximas.

Com a aplicação de frequências diferentes para cada mapeamento, espera-se que a estratégia ETA-PoCon permita uma diminuição no consumo desnecessário de recurso, uma vez que o usuário pode definir frequências maiores para dados mais relevantes e necessários. Porém, cabe ao usuário a definição da frequência de cada mapeamento.

3.3.3. *Análise do volume do delta - $V(\Delta)$*

Como mencionado, o $V(\Delta)$ é diretamente proporcional ao impacto causando no repositório após a transferência do delta. Assim, a definição de um valor mínimo que deve

ser atingido pelo $V(\Delta)$ para ser transferido está relacionada ao impacto que o Δ em questão causará no repositório. Na estratégia ETA-PoCon, além de permitir a definição de intervalos de tempo individuais para cada mapeamento, permite-se ao usuário a definição do volume mínimo de dados que o delta deve possuir para ser transferido. Em outras palavras, a cada mapeamento o usuário tem a opção de definir o mínimo de tuplas que devem ser alteradas para que o processo seja disparado.

Define-se como $V(\Delta_i)$ o volume mínimo que o Δ referente ao mapeamento i deve atingir para disparar o processo de transferência. Em resumo, a cada intervalo definido por T_i , o mecanismo deve verificar o volume do Δ e, caso esse seja maior ou igual ao $V(\Delta_i)$, o processo ETA referente ao mapeamento é executado, caso contrário, nenhuma operação é realizada e os dados não são transferidos.

Para exemplificar a relevância do uso do $V(\Delta_i)$, considere os esquemas e os mapeamentos apresentados na Figura 3.3. A cada dois minutos o processo é disparado e os dados do “Mapeamento 1” são transferidos para o repositório. Desse modo, o processo pode efetuar operações sobre um conjunto de dados não representativo em relação ao volume de informações contidas na tabela “Venda” do repositório, ou seja, o volume de tuplas a ser transferido pode não afetar significativamente as análises executadas sobre o repositório.

Nesse caso, o usuário tem a opção de definir um volume mínimo de tuplas que permita o disparo do processo ETA. Por exemplo, o usuário poderia definir o $V(\Delta_1)$ como 100 e o mecanismo entraria em um ciclo em que a cada dois minutos o volume do Δ é aferido e, caso o volume seja maior ou igual a 100 tuplas, as informações são transferidas.

Com a aplicação desse parâmetro, espera-se permitir a diminuição do número de execuções desnecessárias do processo ETA, baseada na transferência apenas de um número de tuplas que resultem em um impacto significativo. Desse modo, o volume do delta tende a crescer até que atinja o volume considerado relevante, e somente após esse marco é que os dados são então transferidos e o volume do delta é reduzido. O gráfico apresentado na Figura 3.4 representa o crescimento esperado do volume do delta, nessa representação os dados foram transferidos cinco vezes, cada transferência pode ser identificada por um pico da curva apresentada.

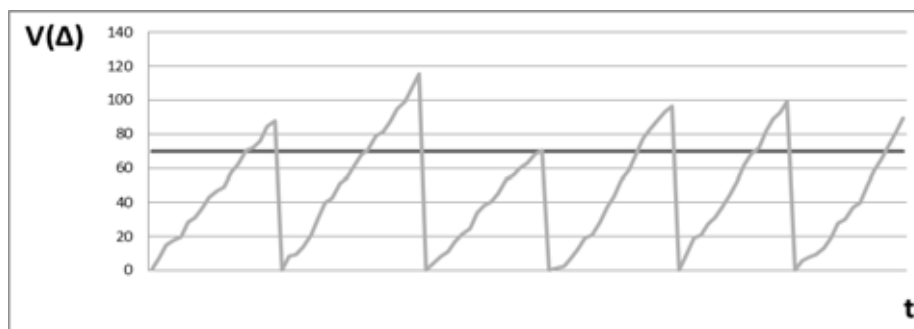


Figura 3.4 - Gráfico que representa a evolução do $V(\Delta)$ no decorrer do tempo

3.3.4. Análise da relevância do delta - $R(\Delta)$

A relevância do delta é outro parâmetro a ser considerado quando se deseja diminuir a execução do processo ETA sobre dados não impactantes. A relevância de uma tupla está diretamente relacionada ao impacto causado quando essa é transferida ao repositório. Na abordagem proposta a análise leva em consideração o número de referências feitas à tupla em questão.

Para exemplificar, considere o esquema apresentado na Figura 3.5. A “Tabela 2” possui uma restrição de chave estrangeira no campo “id_tabela1” referente à chave primária da “Tabela 1”. Como pode ser observado, os registros 1, 2 e 3 da “Tabela 2” fazem referência ao registro 1 da “Tabela 1”, enquanto os registro 4 e 5 fazem referência ao registro 2 da “Tabela 1” e assim sucessivamente.

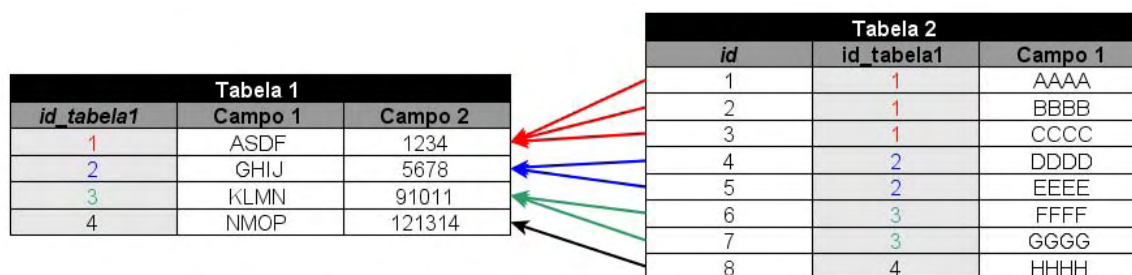


Figura 3.5 - Esquema de representação de relevância de registros

Nesse esquema, a análise sobre a “Tabela 1” permite afirmar que o registro 1 é o mais relevante, uma vez que o mesmo é referenciado três vezes na “Tabela 2”. Na mesma tabela, os registros 2 e 3 teriam o mesmo grau de relevância, pois ambos são referenciados por dois registros da “Tabela 2”. Por fim, o registro 4 teria o menor grau de relevância, sendo referência por apenas um registro na “Tabela 2”.

Esse grau de relevância é estabelecido com base no impacto causado na base de dados caso uma das tuplas seja alterada ou removida. Uma alteração na tupla 1 da “Tabela

1” afeta de forma indireta as informações de três registros, enquanto uma alteração na tupla 2 afeta as informações de apenas 2 registros.

Vale ressaltar que, comumente, uma mesma tabela pode ser referenciada por inúmeras outras. Dessa forma, considerando Tref como o conjunto de tabelas que referenciam uma tabela T, a relevância de uma tupla da tabela T pode ser definida pela expressão apresentada na Figura 3.6.

$$R_{\text{tupla A}} = \frac{\text{Total de tuplas em Tref que referenciam a tupla A}}{\text{Total de tuplas em Tref}}$$

Figura 3.6 - Expressão para cálculo de relevância

Utilizando essa expressão é possível calcular a relevância de cada uma das tuplas da “Tabela 1”, pertencente ao esquema da Figura 3.5. Nesse caso, Tref é constituído apenas pela “Tabela 2” que possui um total de oito tuplas. Na Tabela 3.2 são apresentados o grau de relevância de cada uma das tuplas da “Tabela 1”.

Tabela 3.2 – Relevância das tuplas da Tabela 1

Tupla	Cálculo	Relevância
1	$\frac{3}{8}$	0,375
2	$\frac{2}{8}$	0,25
3	$\frac{2}{8}$	0,25
4	$\frac{1}{8}$	0,13

Portanto, define-se como $R(\Delta_i)$ a relevância mínima que o Δ referente ao mapeamento i deve atingir para disparar o processo de transferência referente ao mapeamento i . $R(\Delta)$ pode ser calculado somando-se a relevância de cada tupla pertencente ao Δ . O valor de $R(\Delta_i)$ sempre estará entre zero e um, pois é definido pelo percentual de tuplas que referencia alguma tupla do Δ_i em relação ao total de tuplas do Tref.

Desse modo, $R(\Delta_i)$ com valor 0,1 indica que o mecanismo deve transferir os dados do Δ_i somente se esses forem referenciados por 10% do total de tuplas Tref. Em outras palavras, o mecanismo deve efetuar a transferência apenas se o impacto causado nas tabelas que referenciam a tabela destino seja maior ou igual a 10%. Da mesma forma, $R(\Delta_i)$ com valor 1 indica que o mecanismo deve transferir o Δ_i apenas se o impacto for de

100%, ou seja, todas as tuplas das tabelas que referenciam a tabela alvo fazem referência a uma tupla do Δ_i .

Diferentemente do $V(\Delta_i)$, o parâmetro $R(\Delta_i)$ é relativo ao repositório. Para a definição da relevância do Δ , o mecanismo deve efetuar aferições tanto nas fontes de dados, para identificação das tuplas que constituem o Δ , quanto no repositório para identificação do total de tuplas que referenciam o Δ .

O funcionamento da análise do $R(\Delta_i)$ é análogo ao processo executado na verificação do $V(\Delta_i)$, em que a cada intervalo definido por T_i , o mecanismo deve verificar a relevância do Δ . Caso esse seja maior ou igual ao $R(\Delta_i)$, o processo referente ao mapeamento é executado, caso contrário, nenhuma operação é realizada e os dados não são transferidos.

Com a aplicação do parâmetro $R(\Delta_i)$, assim como na aplicação do $V(\Delta_i)$, espera-se diminuir o número de execuções desnecessárias do processo ETA, uma vez que somente dados que atinjam o grau de relevância definida pelo usuário irão disparar o processo. Assim, a relevância do Δ_i tende a crescer até que seja atingido o limite definido pelo usuário, os dados são então transferidos e a relevância do delta retorna a zero. É representado no gráfico da Figura 3.7 o crescimento esperado da relevância do delta, cujos dados foram transferidos sete vezes, cada uma identificada como um pico da curva apresentada.

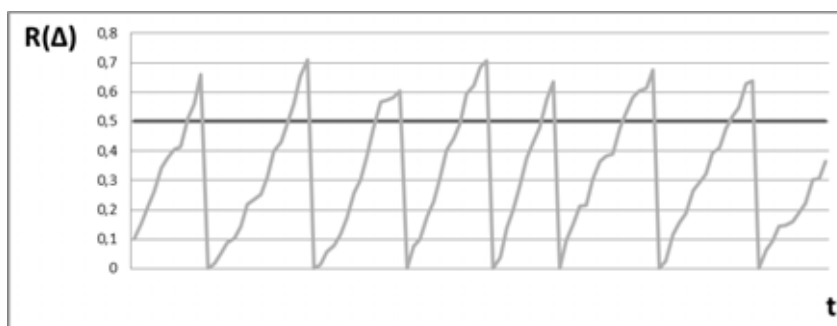


Figura 3.7 - Gráfico que representa a evolução do $V(\Delta)$ no decorrer do tempo

3.4. Ferramenta para Extração, Transformação e Armazenamento - FETA

A fim de validar a estratégia proposta, o desenvolvimento do trabalho contou com a implementação de uma ferramenta ETA para atualizações em curtos períodos de tempo, denominada FETA. A ferramenta é especializada em fontes de dados e repositórios construídos sobre bases de dados relacionais e permite, além da definição de mapeamentos,

a configuração das políticas de propagação definidas na estratégia ETA-PoCon, baseando-se nos parâmetros definidos anteriormente: T , $V(\Delta)$ e $R(\Delta)$.

3.4.1. Arquitetura da FETA

Na Figura 3.8a é apresentada a arquitetura da ferramenta quanto à interação com o usuário administrador do *Data Warehouse*, que é responsável por mapear e configurar as políticas de propagação de dados de cada mapeamento. É possível identificar a presença de duas interfaces: mapeamento e controle, além de uma base de dados para armazenamento dos mapeamentos. A seguir, são descritas as funções de cada um desses elementos:

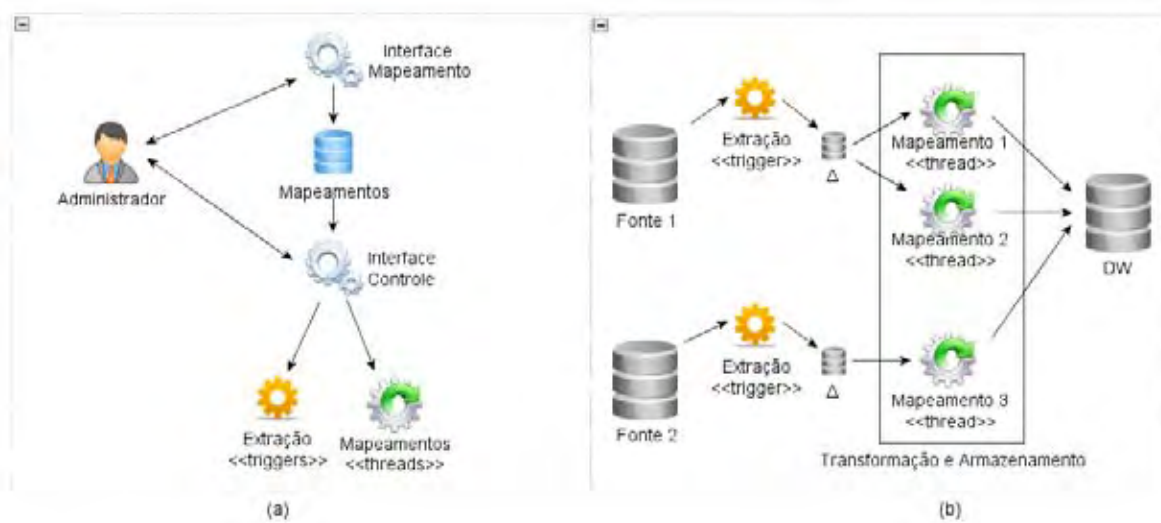


Figura 3.8 – Arquitetura da FETA

- **Interface Mapeamento:** responsável por permitir que o usuário configure os mapeamentos entre qualquer uma das fontes e o repositório. Possui um esquema em que é possível selecionar as tabelas fonte, a tabela destino, as operações a serem aplicadas sobre os dados e os parâmetros de propagação dos dados T , $V(\Delta)$ e $R(\Delta)$;
- **Interface Controle:** tem como objetivo permitir ao usuário o gerenciamento dos processos ETA referentes a cada mapeamento, permite ao usuário inicializar e pausar a transferência dos dados de qualquer um dos mapeamentos. Para tanto, esse elemento é responsável pela criação de *triggers* nas bases fontes e criação de *threads* que em conjunto executam os processos ETA. O funcionamento desses processos será descrito nas próximas seções;
- **Mapeamentos:** o esquema da base de dados de mapeamentos é apresentado na Figura 3.9. Todas as tabelas têm como objetivo armazenar as configurações definidas pelo usuário via **Interface Mapeamento**, e dar suporte aos mecanismos criados pela **Interface Controle**.

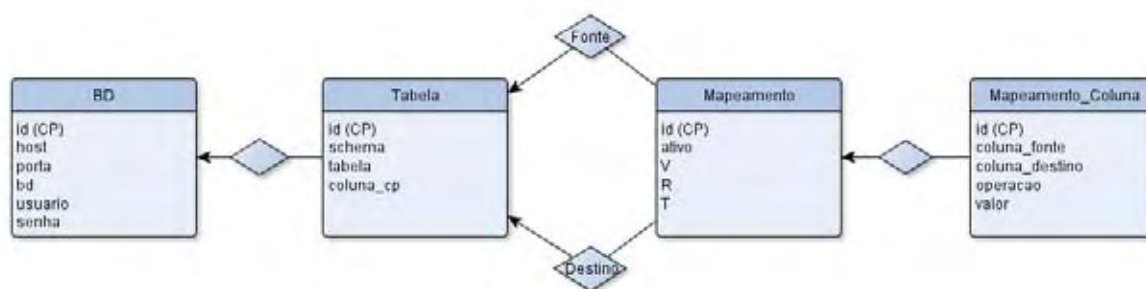


Figura 3.9 - Modelo Entidade-Relacionamento da base para armazenamento dos mapeamentos

Na Figura 3.8b é apresentada a arquitetura da ferramenta em tempo de execução, ou seja, a arquitetura criada para transferir os dados das fontes para o repositório utilizando as configurações definidas pelo usuário. A estrutura é criada pela **Interface Controle** no instante em que o usuário ativa a execução dos processos definidos para os mapeamentos. As *triggers* demonstradas na figura são responsáveis pelo processo de extração dos dados nas fontes, processo descrito na seção 3.4.2, enquanto as *threads* criadas são constituídas pelo mecanismo de transferência dos dados específico a cada mapeamento, descrito na seção 3.4.3.

Na implementação dos módulos da ferramenta foi utilizada a linguagem de programação JAVA, devido principalmente à característica da linguagem em possuir milhares de *Application Programming Interfaces* – APIs que permitem a construção de aplicativos de forma portátil e modularizada. Dentre as APIs utilizadas, destaca-se o *Java Database Connectivity* ou JDBC, que além de permitir a conexão com inúmeros SGBDs de forma homogênea, disponibiliza um conjunto de métodos que facilitam o acesso aos metadados das bases de dados relacionais.

Tanto a **Interface Mapeamento**, quanto a **Interface Controle** foram implementadas em plataforma *WEB*, o que exigiu a aplicação da tecnologia *JavaServer Pages* – JSP e das bibliotecas *JavaScript*: *jQuery* e *EXT*. O SGBD utilizado na construção da ferramenta e nos testes apresentados no próximo capítulo foi o *PostgreSQL*, a escolha teve como base a facilidade na manipulação desse gerenciador e o fato de ser livre de licenças.

3.4.2. Processo de Extração

Na estratégia proposta, parte do processo de extração dos dados nas fontes é executado continuamente, ou seja, independentemente da transferência ou não, os dados alterados são marcados. Para contemplar essa característica na ferramenta foi escolhida a

utilização de *triggers* que têm como objetivo marcar em cada uma das fontes os dados alterados desde a última atualização do DW, ou seja, construir o delta (Δ).

A escolha da aplicação de *triggers* foi realizada com base em dois principais pontos: a criação desse tipo de mecanismo permite a identificação de dados alterados sem a necessidade de alterações nos sistemas fontes, sendo necessária apenas a criação de funções no próprio banco de dados; as *triggers* criadas podem ser otimizadas de modo a afetar minimamente as operações prioritárias do banco de dados, o que reduz o consumo de recurso do SGBD.

A **Interface Controle** identifica por meio da base de mapeamentos as tabelas que serão utilizadas como fonte de dados e, para cada uma delas, cria uma *trigger* que dispara uma função a cada operação de INSERT, UPDATE ou DELETE executada sobre a tabela. Essa função, por sua vez, armazena em uma tabela específica os dados que foram alterados e qual operação foi realizada, correspondendo ao Δ da tabela fonte.

Na Figura 3.10a é exemplificada a criação de uma dessas *triggers* e na Figura 3.10b é exemplificada a função disparada a cada operação sobre a tabela. A criação dessas estruturas é específica a cada tabela fonte, uma vez que manipulam as chaves primárias e exige que **Interface Controle** acesse os metadados de cada fonte e gere um código específico para cada uma, não sendo possível a criação de estruturas genéricas a todas as tabelas.

```

CREATE TRIGGER cdc_public_TABELA
AFTER INSERT OR UPDATE OR DELETE
ON TABELA
FOR EACH ROW
EXECUTE PROCEDURE cdc.public_TABELA('chave_primaria');

```

(a)

```

CREATE OR REPLACE FUNCTION cdc.public_TABELA()
RETURNS trigger AS
$BODY$
BEGIN
    IF (TG_OP = 'DELETE') THEN
        INSERT INTO cdc.cdc (instancia, tabela, operacao, coluna_pk, id_tupla)
        VALUES (TG_TABLE_SCHEMA, TG_TABLE_NAME, TG_OP, 'id', OLD.id::text);
    ELSE
        INSERT INTO cdc.cdc (instancia, tabela, operacao, coluna_pk, id_tupla)
        VALUES (TG_TABLE_SCHEMA, TG_TABLE_NAME, TG_OP, 'id', NEW.id::text);
    END IF;
    RETURN NULL;
END;
$BODY$
LANGUAGE plpgsql;

```

(b)

Figura 3.10 - Exemplo de *trigger* utilizada no processo de extração

3.4.3. *Processo de Transformação e Armazenamento*

Os processos de Transformação e Armazenamento são executados por meio de *threads* criados especificamente para cada mapeamento. Como sugerido pela estratégia ETA-PoCon, cada mapeamento é tratado de forma individual, as configurações definidas pelo usuário são aplicadas independentemente dos atributos e características dos outros mapeamentos.

A escolha da utilização de *threads* foi realizada com intuito de atender a característica de independência entre os mapeamentos. Além disso, erros que ocasionalmente possam ocorrer no processo de um mapeamento não afetam as operações de outros mapeamentos, o que contribui para manutenção da consistência das informações e continuidade do processo de atualização do DW.

Como fora mencionado, os processos ETA são executados sobre grandes conjuntos de dados. A aplicação de *threads* otimiza o consumo de recursos de processamento quando executada em CPUs que possuem vários núcleos de processamento. Nessa configuração, os processos de cada mapeamento são distribuídos entre os núcleos de processamento, o que diminui consideravelmente o tempo de execução das operações de transformação.

As *threads*, assim como as *triggers*, são criadas pela **Interface Controle** com base na configuração de mapeamentos definida pelo usuário. Ao inicializar um ou mais mapeamentos, a interface busca na base de mapeamentos as tabelas fontes, a tabela destino, as operações e os parâmetros de transferência e, logo em seguida cria para cada mapeamento uma *thread* responsável por transferir os dados aplicando as operações e levando em consideração as políticas de propagação. O funcionamento de cada uma dessas *threads* é descrito na próxima seção.

3.4.4. *Mecanismo de disparo de transferência de dados*

O mecanismo que define a frequência de atualização do DW é executado por cada uma das *threads*. Para atender às características definidas pela estratégia, o mecanismo é constituído por um algoritmo que, a partir das definições do usuário, mantém um ciclo em que, a cada intervalo de tempo, é definido se o delta referente ao mapeamento de responsabilidade da *thread* deve ou não ser transferido ao repositório. A seguir, é apresentado o algoritmo executado:

Algoritmo**Entrada:**

T – Tempo definido pelo usuário para o mapeamento (*T_i*)

V – *V*(Δ_i) definido pelo usuário – caso 0, usuário não definiu política por volume

R – *R*(Δ_i) definido pelo usuário – caso 0, usuário não definiu política por relevância

Stop – variável de controle utilizada para inicializar e pausar a execução de cada thread

Procedimento:

Enquanto(*stop* diferente de 1)

Boolean propaga = 1;

Boolean propaga_volume = 0;

Boolean propaga_relevancia = 0;

Delta = conjunto de tuplas alteradas desde a última propagação;

Se (*V*)

propaga_volume = *verificaVolumeTuplas*(*V*, *Delta*);

propaga = 0;

Se (*R*)

Propaga_relevancia = *verificaRelevânciaTuplas*(*R*, *Delta*);

propaga = 0;

propaga = (*propaga* OU *propaga_volume* OU *propaga_relevancia*)

Se (*propaga*)

propagaDelta(*Delta*); //função que transfere os dados

sleep (*T*); //função que para a execução durante *T* segundos

Nesse algoritmo, é possível identificar uma chamada à função *propagaDelta()*, que tem como objetivo transferir ao repositório os dados das tuplas que constituem o delta. Essa função é formada pelo processo de transformação e armazenamento, enquanto todas as outras operações constituem o mecanismo de controle. Vale lembrar que este trabalho não tem foco nos processos de transformação e, portanto, a função *propagaDelta()* não será discutida em detalhes.

É possível identificar também a função *verificaVolumeTuplas()*, responsável por analisar o volume do delta e retornar 1, caso o volume tenha atingido o *V*(Δ_i) e 0 caso contrário. A seguir é apresentado o algoritmo executado por essa função:

Algoritmo - verificaVolumeTuplas**Entrada:***V – $V(\Delta_i)$ definido pelo usuário**Delta – Dados alterados desde a última atualização***Procedimento:***Volume = Total de tuplas do Delta;**Se (Volume \geq V)**Retorna 1;**Senão**Retorna 0;*

Além da função *verificaVolumeTuplas()*, o primeiro algoritmo apresentado possui uma chamada à função *verificaRelevânciaTuplas()*, cujo objetivo é a aferição da relevância do conjunto de tuplas do delta. A relevância é definida de acordo com a estratégia proposta, ou seja, é dada pela relação entre o total de tuplas que poderia fazer referência ao delta e o total que de fato fazem referência. A seguir é descrito o algoritmo executado por essa função:

Algoritmo**Entrada:***R – $R(\Delta_i)$ definido pelo usuário**Delta – Dados alterados desde a última atualização***Procedimento:***String vetor_cp_tuplas[]; //vetor para armazenar chaves primárias das tuplas do delta**Inteiro total_tuplas, total_ref;**Real relevancia;**Para cada tupla t_i do Delta**cp_tupla = valor da chave primária de t_i ;**Adiciona cp_tupla ao vetor vetor_cp_tuplas[];**total_tuplas = 0;**total_ref = 0;**Tref = conjunto de tabelas no DW que referenciam a tabela destino;**Para cada tab_i em Tref**total_tuplas += total de tuplas na tabela tab_i;**total_ref += total de tuplas da tabela tab_i que referenciam alguma tupla*

```

do vetor_cp_tuplas;
Se (total_tuplas ==0 || total_ref ==0)
    Retorna 0;
relevancia = total_ref / total_tuplas;
Se (relevancia >= R)
    Retorna 1;
Senão
    Retorna 0;

```

Esse algoritmo pode ser considerado o mais complexo dentre os apresentados, uma vez que sua execução conta com acessos à estrutura do repositório para identificação das tabelas que referenciam a tabela destino. Além disso, são efetuadas consultas para contagem do total de tuplas de cada uma dessas tabelas e o total de tuplas que referenciam o delta.

3.5. Considerações finais

Neste capítulo, foi descrito o problema existente na definição de estratégias que permitam a atualização de DWA e evitam a sobrecarga dos sistemas envolvidos. Mostrou-se também que uma das alternativas na solução desse problema é a criação de estratégias que priorizem dados considerados sensíveis, e assim seja diminuída a execução de operações sobre dados não relevantes. Essa priorização não é uma tarefa trivial e deve contar com análises sobre variáveis como volume dos dados a ser transferido e relevância desse volume no repositório.

Frente aos problemas descritos, foi apresentada a estratégia ETA-PoCon que define a maneira como volume e relevância devem ser tratados e qual o resultado esperado desse tratamento. Por fim, foi discutida em detalhes a implementação da FETA construída com intuito de validar a estratégia proposta. A arquitetura da ferramenta, as tecnologias utilizadas e alguns dos principais algoritmos implementados, foram descritos e discutidos.

Capítulo 4 Experimentos e Resultados

4.1. Considerações iniciais

Nesse capítulo são apresentados os testes e resultados obtidos com a aplicação da estratégia ETA-PoCon por meio da utilização da ferramenta descrita no capítulo anterior. Para a execução dos testes foram elaborados experimentos que utilizaram uma base de dados fonte extraída de um sistema real e um DW construído exclusivamente para os testes. Os primeiros experimentos apresentados tratam da aplicação das políticas baseadas em tempo, em seguida são apresentados os testes executados para verificação das políticas baseadas em relevância e por fim apresentam-se alguns experimentos para validação dos resultados obtidos com uso da relevância quanto à semântica dos dados transferidos. Além disso, são descritas algumas análises e discussões sobre os resultados obtidos.

4.2. Ambiente utilizado

A seguir são descritas as bases de dados utilizadas nos experimentos e a configuração do *hardware* da máquina onde os testes foram executados.

4.2.1. Bases de dados

A base de dados utilizada como fonte foi extraída do Sistema de Informação e Vigilância de Acidentes de Trabalho – SIVAT e é composta por acidentes de trabalho registrados em mais de 100 municípios do interior do estado de São Paulo. A base

armazena mais de 70 mil acidentes e possui uma estrutura que possibilita a configuração da estratégia e validação por meio da execução de simulações de inserções e alterações nos dados.

No ambiente utilizado para os testes foram considerados os esquemas apresentados na Figura 4.1. Tanto na base de dados fonte quanto no repositório foram consideradas apenas as tabelas “Empresa” e “Ficha”. A tabela “Ficha” é utilizada para o armazenamento das notificações de acidentes de trabalho e possui informações como nome do acidentado, local do acidente, data e etc. Já a tabela “Empresa” armazena informações da empresa que emprega o acidentado e possui informações como nome da empresa e ramo de atividade. Vale ressaltar que a cardinalidade entre “Empresa” e “Ficha” é de N para 1, ou seja, uma empresa pode empregar vários empregados acidentados, enquanto um acidente pode envolver apenas uma empresa empregadora.

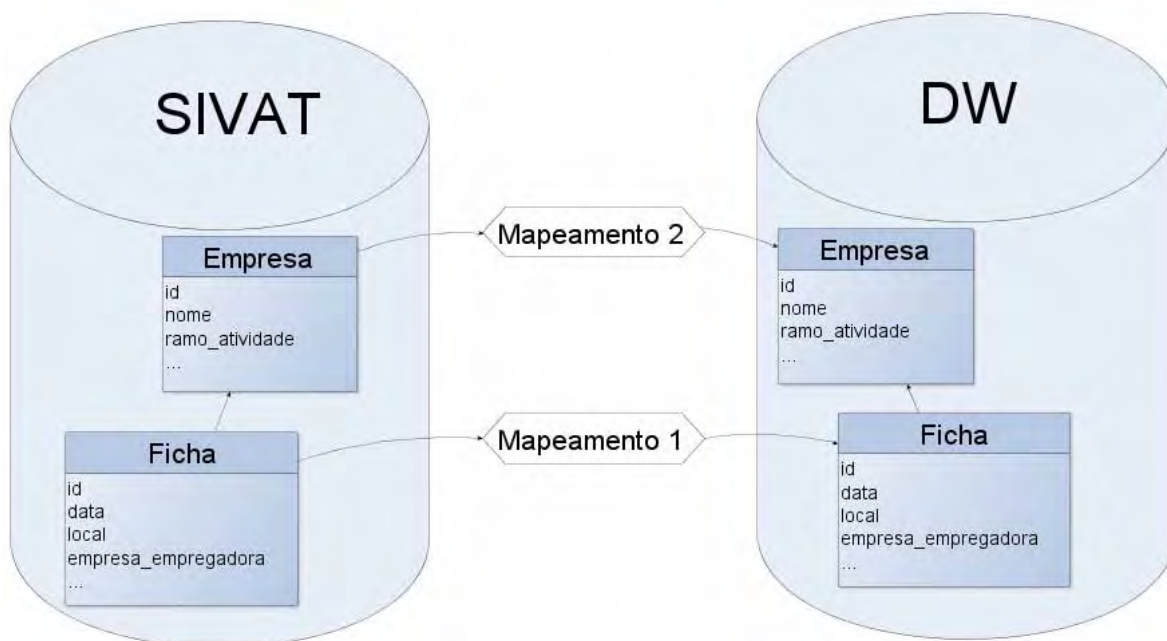


Figura 4.1 - Esquemas utilizados durante os testes

Na Figura 4.1 são representados dois mapeamentos, sendo que o Mapeamento 1 especifica que os dados da tabela “Ficha” da base SIVAT devem ser transformados e inseridos na tabela “Ficha” e o Mapeamento 2 especifica que os dados da tabela empresa devem ser transferidos à tabela de mesmo nome no DW.

Os experimentos descritos nesse capítulo contaram com a simulação de transações executadas sobre a base de dados fonte (SIVAT) e análise dos valores medidos do volume e da relevância do delta durante essas operações. De maneira simplificada, as transações executadas resumem-se em operações de INSERT, UPDATE ou DELETE na tabela

“Empresa”. No início de cada experimento, a base de dados fonte e o DW são colocados em um mesmo estado, ou seja, possuem exatamente os mesmos dados.

Com essa configuração é possível efetuar análises sobre um ambiente próximo de um ambiente real de DW ativo, cujo repositório já está construído e é necessário executar transferências dos dados das fontes que são continuamente atualizadas pelos sistemas que geram os dados. Além disso, as operações executadas são geradas aleatoriamente, o que permite aproximar ainda mais as simulações de um ambiente real.

4.2.2. Hardware utilizado

Para os experimentos realizados a ferramenta FETA foi configurada em uma máquina com processador Intel Core i5 M460, que possui quatro núcleos de processamento, e memória principal de quatro Gigabytes. O sistema operacional utilizado foi o Windows 7 Ultimate, e como SGBD utilizou-se o PostgreSQL 8.4.

4.3. Aplicação de política baseada em volume - $V(\Delta)$

Nessa seção serão descritos os experimentos executados com objetivo de validar a utilização da estratégia ETA-PoCon quanto à aplicação das políticas baseadas em volume. A seguir são apresentados cada um dos experimentos e, para tanto, demonstram-se os parâmetros aplicados e a configuração utilizada. Ao final da seção é apresentada uma discussão sobre os resultados obtidos.

4.3.1. Experimento I

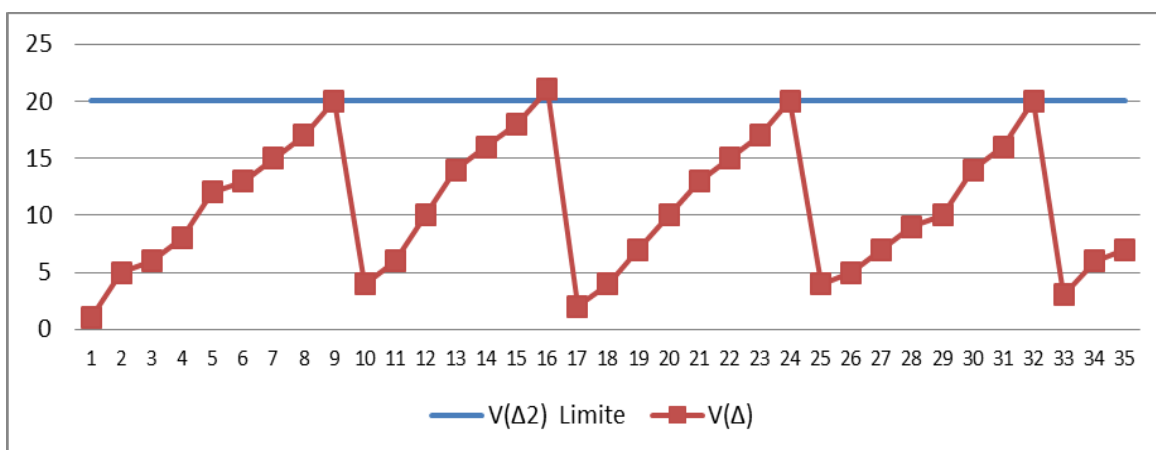
Na Tabela 4.1 são apresentados os parâmetros utilizados nesse experimento. Não foram utilizados valores referentes à relevância com intuito de validar o funcionamento da proposta quanto à aplicação da política por volume. Na Tabela 4.2 são apresentadas as informações referentes à base de dados fonte, o repositório e as transações executadas, enquanto na Figura 4.2 é apresentado o gráfico da evolução do $V(\Delta)$ durante o experimento.

Tabela 4.1 - Parâmetros utilizados no experimento I

<i>Mapeamento(i)</i>	<i>Ti</i>	<i>V(Δi)</i>
2	5 segundos	20

Tabela 4.2 – Configurações do experimento I

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
1016	683	500	10 Minutos	0,83

Figura 4.2- Gráfico da evolução do $V(\Delta)$ durante o experimento I

No gráfico apresentado, os pontos em vermelho representam o volume medido a cada ciclo do mecanismo que controla a transferência dos dados referente ao Mapeamento 2 do esquema apresentado na Figura 4.1. A linha em azul apresenta o parâmetro $V(\Delta 2)$, ou seja, o limite máximo que o volume do delta deve atingir para disparar a transferência. Portanto, cada um dos pontos identifica um ciclo do mecanismo controlador em que é executada uma aferição ao volume e é definido se as alterações contidas no delta devem ou não ser transferidas.

O gráfico apresentado na Figura 4.2 demonstra que o DW foi atualizado quatro vezes, sendo que as atualizações podem ser identificadas por cada pico da curva da linha em vermelho. Tal característica do gráfico evidencia a aplicação da estratégia proposta, em que as operações sobre a fonte são adicionadas ao delta até que o volume seja maior ou igual ao limite definido ($V(\Delta 2)$). Nesse momento, os dados são transferidos e o volume é reduzido.

É possível verificar que a estratégia baseada em volume permite diminuir o total de atualizações do DW. A não utilização da política por volume ocasionaria a transferência dos dados a cada ciclo do mecanismo controlador e, dessa forma, seriam executadas 35 transferências, enquanto com a aplicação do $V(\Delta 2)$ foram apenas 4 atualizações. Portanto,

o primeiro experimento demonstrou uma redução de aproximadamente 91% no total de atualizações do DW.

Vale ressaltar ainda que em ferramentas cuja estratégia de atualização do DW adota uma política em que todo dado inserido na fonte é imediatamente transferido ao repositório, o número de atualizações seria igual ao total de transações executadas, ou seja, seriam executadas 500 atualizações.

4.3.2. Experimento II

O segundo experimento seguiu a mesma linha do anterior, tanto o estado inicial da fonte e do repositório quanto os parâmetros foram alterados a fim de verificar se o comportamento apresentado no experimento I se repete em diferentes situações. Os parâmetros e configurações utilizados são apresentados na Tabela 4.3 e Tabela 4.4 respectivamente. O gráfico da evolução do $V(\Delta)$ é apresentado na Figura 4.3.

Tabela 4.3 – Parâmetros utilizados no experimento II

$Mapeamento(i)$	Ti	$V(\Delta i)$
2	5 segundos	100

Tabela 4.4 – Configurações do experimento II

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
22700	10633	1000	5 Minutos	3,3

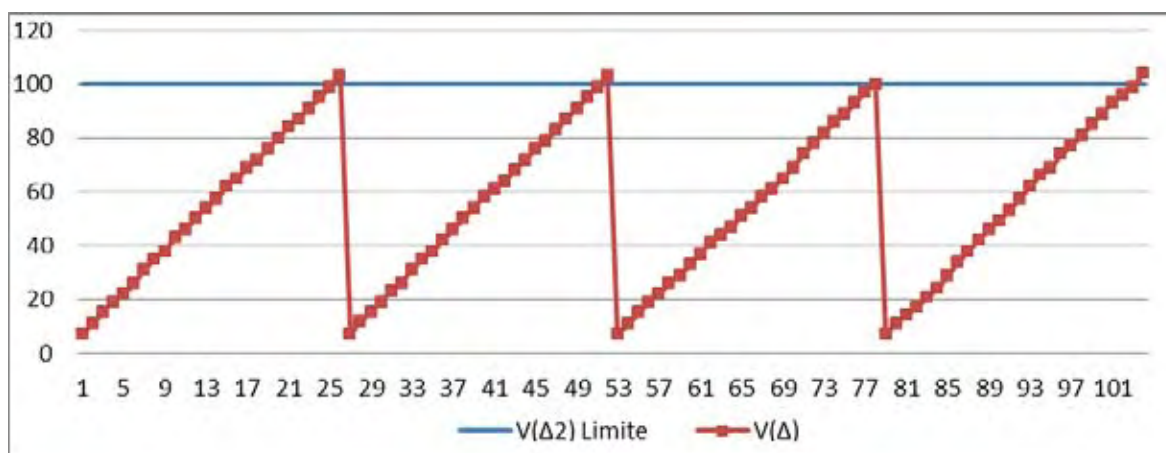


Figura 4.3 - Gráfico da evolução do $V(\Delta)$ durante o experimento II

A diferença entre esse experimento e o anterior resume-se em um aumento do parâmetro $V(\Delta_2)$ de 20 para 100 tuplas e um aumento na taxa de execução de transações de 0,8 tuplas por segundo para 3,3.

A análise do gráfico permite identificar que, assim como no experimento anterior, houve um total de quatro atualizações do DW. Se comparado com o total de atualizações que seriam executadas sem a aplicação da política baseada em tempo é possível afirmar que houve uma redução de aproximadamente 96%, uma vez que seriam executadas 100 atualizações. Por outro lado, o gráfico também demonstra que, em média entre uma atualização e outra, foram executados 25 ciclos do mecanismo controlador e como nesse caso o ciclo foi definido como 5 segundos, identifica-se que entre uma atualização e outra se passaram em média 2 minutos.

Portanto, nesse experimento o comportamento da política baseada em volume foi mantido, ou seja, houve uma redução considerável no total de atualizações executadas. Porém, há uma forte ligação entre $V(\Delta)$ definido e o intervalo entre as atualizações e, dessa forma, a escolha do $V(\Delta)$ pode fazer com que o intervalo entre as atualizações aumente ao ponto de tornar inviável a aplicação da estratégia.

4.3.3. *Experimento III*

Semelhante ao experimento II, como pode ser verificado na Tabela 4.5, foi efetuada uma alteração no parâmetro $V(\Delta_2)$ reduzindo-o para 10 tuplas. As configurações utilizadas foram mantidas assim como descritas na Tabela 4.4. O objetivo foi verificar o comportamento da estratégia com a definição de um volume relativamente baixo.

Tabela 4.5 - Parâmetros utilizados no experimento III

<i>Mapeamento(i)</i>	<i>Ti</i>	<i>V(Δi)</i>
2	5 segundos	10

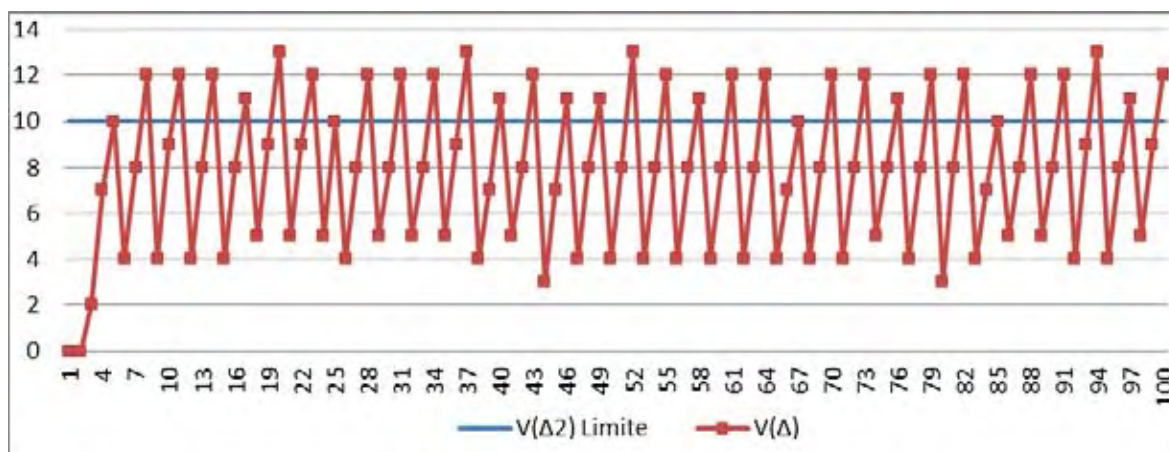


Figura 4.4 - Gráfico da evolução do $V(\Delta)$ durante o experimento III

No gráfico apresentado na Figura 4.4, referente à evolução do $V(\Delta)$ durante o experimento III, é possível identificar um comportamento diferente dos experimentos anteriores. Foram efetuadas 33 transferências, enquanto a não utilização da estratégia executaria um total de 100 atualizações, ou seja, a redução foi de apenas 66%. A diminuição no total de atualizações é reduzida porque o $V(\Delta 2)$ definido é relativamente baixo, ou seja, a taxa de transações executada sobre a fonte faz com que em poucos segundos o volume de delta atinja o limite definido.

Nesse caso, a utilização da política baseada em volume, apesar de diminuir o total de atualizações, pode não ser interessante. Vale ressaltar que a estratégia adiciona operações para aferição do volume do delta, e, caso a redução no total de atualizações não seja satisfatória, a estratégia se torna ineficiente por adicionar mais operações ao processo e aumentar o consumo de recursos na fonte de dados; consequentemente afetará suas tarefas prioritárias.

4.3.4. *Discussão dos resultados*

Os experimentos descritos demonstraram que a política baseada em tempo definida na estratégia ETA-PoCon pode ser uma boa alternativa para a redução do total de atualizações de DW ativos, cujos resultados obtidos são apresentados na Figura 4.5. Porém, a escolha do parâmetro $V(\Delta)$ é complexa devido ao fato de que o volume do delta está diretamente relacionado à taxa de alterações executadas na fonte de dados e afeta diretamente o intervalo de tempo entre as atualizações.

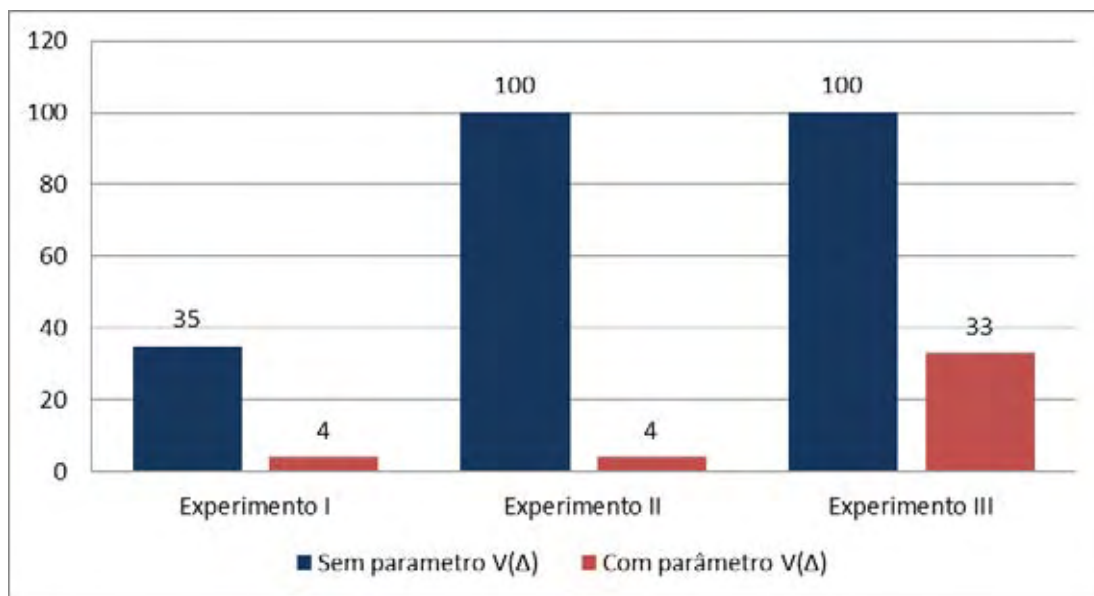


Figura 4.5 - Gráfico que demonstra o total de atualizações executadas com e sem a utilização do parâmetro $V(\Delta)$ durante os experimentos I, II e III

Em ambientes cuja taxa de transação varia constantemente, ou seja, em ambientes em que alguns períodos há uma grande concentração de operações nas fontes de dados e em outros a taxa é reduzida, a aplicação da política por volume pode não ser interessante, uma vez que com muitas transações o volume rapidamente atingiria o $V(\Delta)$ e o gráfico do volume seria semelhante ao apresentado no experimento III. Além disso, com número reduzido de operações, o DW passaria a ser atualizado em intervalos de tempo longos – como demonstrado no gráfico do experimento II.

Logo, a utilização dessa política se torna mais relevante em aplicações em que a taxa de transações é constante. Nesses ambientes, o administrador do DW pode efetuar uma análise sobre a taxa de transação e o intervalo de atualização desejado e definir um valor para o $V(\Delta)$ que permita uma redução considerável no total de atualizações do DW e a consequente diminuição na sobrecarga dos sistemas.

4.4. Aplicação de política baseada em Relevância - $R(\Delta)$

Nessa seção, são descritos os experimentos realizados com intuito de analisar o comportamento das variáveis envolvidas quando aplicada a política baseada em relevância. Assim como na seção anterior, são descritos cada um dos experimentos juntamente com os parâmetros e as configurações utilizadas. O objetivo principal é verificar se a análise da relevância das informações permite reduzir o total de atualizações por meio da priorização de informações sensíveis. Todos os experimentos descritos nessa seção utilizaram o

esquema apresentado na Figura 4.1 e contaram com a mesma metodologia aplicada nos experimentos apresentados anteriormente.

4.4.1. Experimento I

O experimento I objetivou efetuar uma primeira análise da política por relevância. Na Tabela 4.6 são apresentados os parâmetros aplicados no experimento. Note que não houve aplicação de parâmetro para volume. Na Tabela 4.7 são apresentadas as configurações utilizadas e na Figura 4.6 é apresentado o gráfico da evolução do $R(\Delta)$ durante a execução do experimento.

Tabela 4.6 - Parâmetros utilizados no experimento I

$Mapeamento(i)$	T_i	$R(\Delta_i)$
2	10 segundos	1%

Tabela 4.7 – Configurações do experimento I

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
1016	683	100	10 Minutos	6

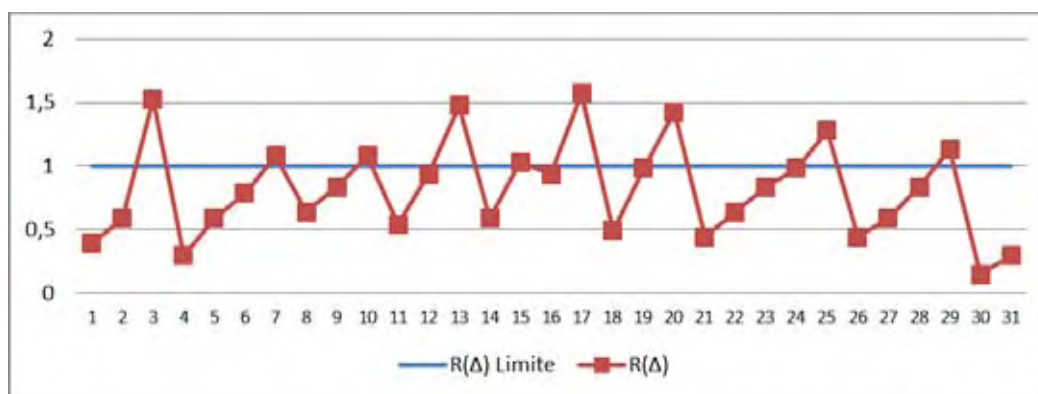


Figura 4.6 - Gráfico da evolução do $R(\Delta)$ durante o experimento I

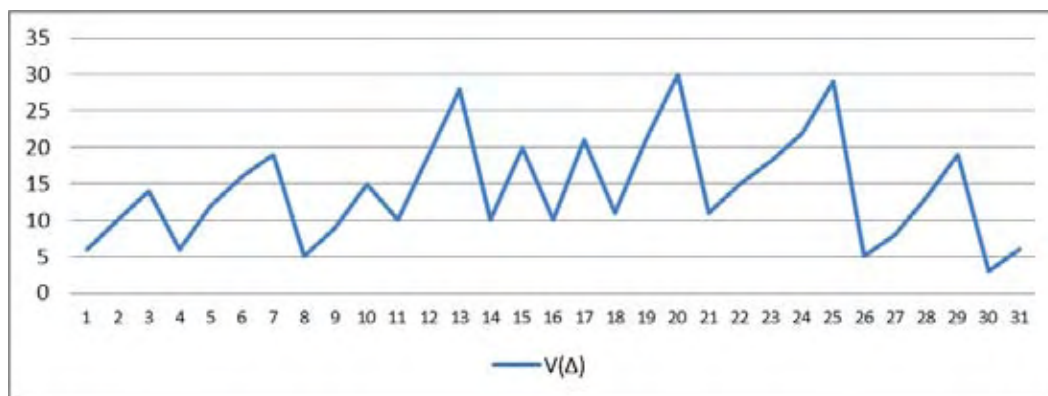


Figura 4.7 - Gráfico da evolução do $V(\Delta)$ durante o experimento I

Assim como nos gráficos apresentados anteriormente, cada ponto do gráfico da evolução do $R(\Delta)$ representa um ciclo do mecanismo controlador executado a cada 10 segundos (T_2). Nesse caso, a cada ciclo é efetuada uma aferição à relevância do delta e, caso a relevância atinja o $R(\Delta_2)$, os dados são transferidos e a relevância é reduzida. Com isso, os picos da curva em vermelho representam o instante em que o delta foi transferido ao repositório.

O gráfico do experimento I permite identificar que foram executadas nove transferências, enquanto a não aplicação da estratégia resultaria em um total de 31. Portanto, a diminuição no total de atualizações foi de aproximadamente 71%. A análise sobre a relevância é relativa ao DW, ou seja, ao definir o limite de 1% de relevância, exige-se que o impacto causado no DW ao transferir esses dados seja de 1% do total possível.

A análise do gráfico demonstra que a relevância do delta assim como o volume nos experimentos anteriores tende a crescer até atingir o limite e depois é fortemente reduzida após a transferência dos dados. Em alguns pontos em que houve a transferência dos dados, a relevância aferida atingiu 1,5% e em outros o valor foi de aproximadamente 1%. Essa diferença se dá pelo fato de que, em alguns períodos, foram executadas operações sobre “Empresas” mais relevantes, ou seja, tuplas da tabela “Empresa” referenciadas por um número alto de fichas de notificações de acidente. Em outras palavras, a relevância do delta é diretamente proporcional ao número de acidentes registrados para as empresas que constituem o delta.

Na Figura 4.7, é apresentado o gráfico da evolução do $V(\Delta)$ durante a execução do experimento I. Nesse caso, o volume aferido é utilizado apenas para análises sobre o comportamento, não sendo parâmetro de decisão para a transferência ou não dos dados. É possível observar que no instante três o delta atingiu uma relevância de aproximadamente 1,5% com um volume menor que 15 tuplas, enquanto no instante 20 a relevância atingiu os mesmos 1,5%, entretanto, o volume do delta era de 30 tuplas. Essa característica do gráfico

evidencia a estratégia utilizada para priorização de dados sensíveis, ou seja, um volume pequeno do delta pode atingir a relevância desejada à medida que as transações executadas foram sobre dados bastante referenciados.

O experimento I apresentou um comportamento esperado, pois há uma diminuição considerável no total de atualizações e é possível verificar que a relevância do delta independe do volume.

4.4.2. Experimento II

Nesse experimento, seguiu-se a mesma linha do anterior, porém, o parâmetro T foi reduzido e as transações executadas foram aumentadas para 200. Na Tabela 4.8 são apresentados os parâmetros utilizados e na Tabela 4.9 as configurações. Nas Figura 4.8 e Figura 4.9 são apresentados os gráficos da evolução do $R(\Delta)$ e $V(\Delta)$ respectivamente.

Tabela 4.8 - Parâmetros utilizados no experimento II

<i>Mapeamento(i)</i>	<i>Ti</i>	<i>R(Δi)</i>
2	5 segundos	1%

Tabela 4.9 – Configurações do experimento II

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
1016	683	200	5 Minutos	0,33

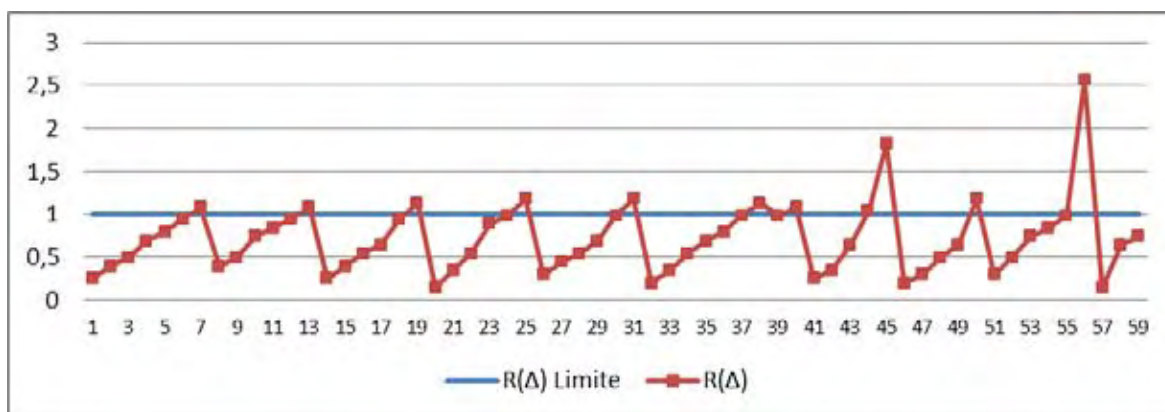


Figura 4.8 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento II

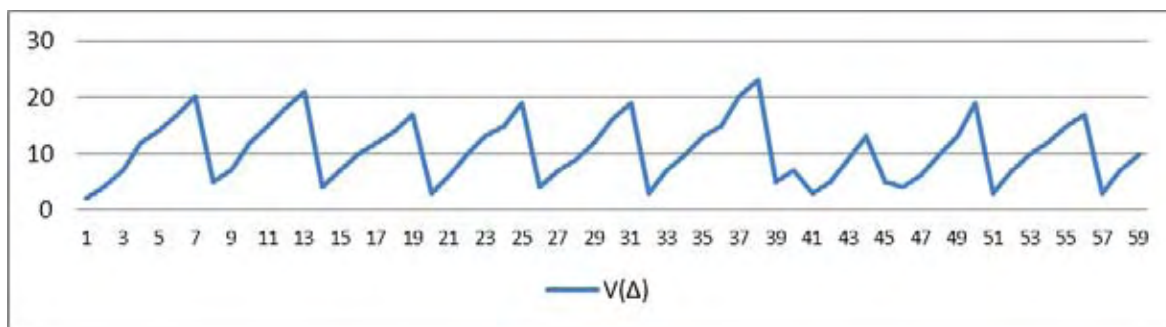


Figura 4.9 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento II

Nesse experimento, o gráfico da evolução do $R(\Delta)$ permite afirmar que foram executadas 10 transferências dos dados. Se comparado com o total de transferências executadas sem a aplicação desse parâmetro, é possível afirmar que houve uma redução de 83%, uma vez que seriam executadas 59 atualizações.

Assim como no experimento I, é possível verificar que, em alguns momentos, a relevância esperada é atingida com um número reduzido de tuplas no delta. Por exemplo, no instante 38 a relevância de 1% foi atingida com aproximadamente 25 tuplas, enquanto no instante 40 a relevância se aproximou de 2% com menos de 10 tuplas. Desse modo, enfatiza-se a independência entre o volume e a relevância definida na estratégia proposta.

É demonstrado no experimento II o mesmo comportamento do primeiro, ou seja, mesmo com a redução no parâmetro T e aumento no total de operações simuladas, o $R(\Delta_2)$ definido permite uma redução considerável no total de atualizações e a relevância do delta independe do volume de transações executadas sobre a fonte de dados.

4.4.3. Experimento III

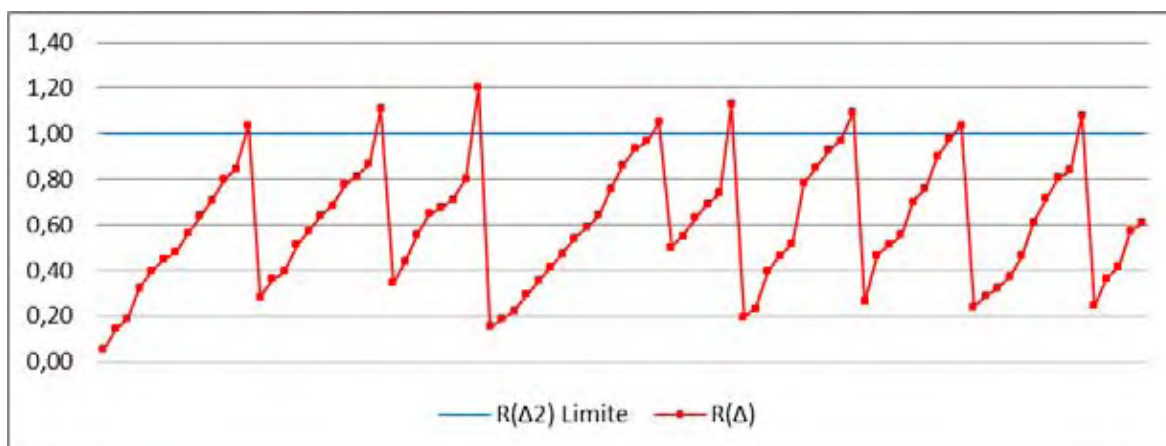
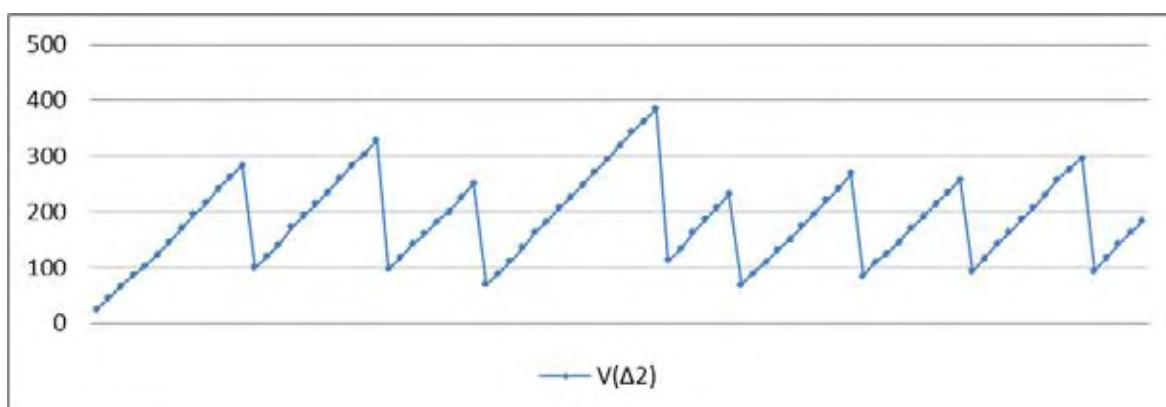
Neste experimento foram efetuadas alterações apenas na configuração do ambiente utilizado. A quantidade de tuplas nas tabelas ficha e empresa foram aumentadas consideravelmente, assim como a taxa de transações executadas sobre a fonte. O objetivo do experimento é verificar se em um ambiente com maior número de tuplas e transações, a característica da política por relevância é mantida. Na Tabela 4.10 são apresentados os parâmetros utilizados e na Tabela 4.11 as configurações.

Tabela 4.10 - Parâmetros utilizados no experimento III

$Mapeamento(i)$	T_i	$R(\Delta_i)$
2	5 segundos	1%

Tabela 4.11 – Configurações do experimento I

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
73625	18756	2380	10 Minutos	3,96

Figura 4.10 - Gráfico da evolução do $R(\Delta)$ durante a segunda etapa do experimento IIIFigura 4.11 - Gráfico da evolução do $V(\Delta)$ durante a segunda etapa do experimento III

É demonstrado no gráfico apresentado na Figura 4.10 que nesse experimento foram executadas oito transferências. No total, o mecanismo executou 60 aferições à relevância, não especificadas no gráfico porque o tornaria ilegível. Desse modo, conclui-se que a redução no total de atualizações foi de 86%. É possível verificar também que o volume do delta no momento da transferência dos dados variou de 400 a 200 tuplas, o que demonstra um padrão semelhante aos anteriores.

4.4.4. *Discussão dos resultados*

Nos três experimentos apresentados verificou-se uma redução considerável no total de atualizações quando comparado com o comportamento do experimento sem a utilização da política por relevância. Na Figura 4.12 são apresentados os resultados obtidos. Os experimentos evidenciaram também que a estratégia mantém certo grau de independência entre o percentual de relevância e o volume do delta e dessa forma, o mecanismo controlador efetua a transferência dos dados priorizando os dados mais referenciados, ou seja, os dados que mais afetam as análises sobre o DW.

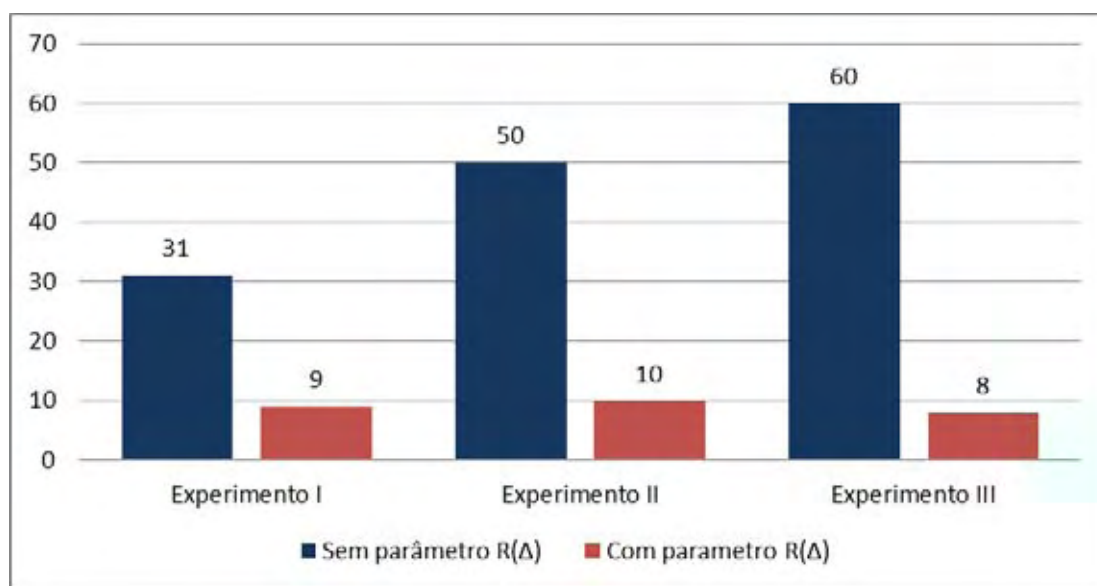


Figura 4.12 - Gráfico que demonstra o total de atualizações executadas com e sem a utilização do parâmetro $R(\Delta)$ durante os experimentos I, II e III

Como o $V(\Delta)$, o parâmetro $R(\Delta)$ deve ser definido pelo usuário administrador do DW e, caso escolhido um percentual relativamente baixo, o DW será atualizado inúmeras vezes e a estratégia pode se tornar não interessante e, caso escolhido um percentual alto, podem ocorrer intervalos entre as transações elevados ao ponto de tornar inviável a aplicação desse parâmetro. A definição do $R(\Delta)$ pode se tornar uma tarefa complexa.

Por outro lado, o uso da relevância transfere os dados com prioridade para dados sensíveis. Quando a relevância do delta atinge o percentual especificado, os dados são migrados, ou seja, independentemente da frequência de atualização que será estabelecida, a migração dos dados sempre priorizará os dados com maior número de referências.

4.5. Validação da política por relevância

As análises sobre os experimentos apresentados na seção anterior demonstraram que a aplicação da política baseada em relevância definida na estratégia ETA-PoCon pode ser bastante útil na redução do total de atualizações em DW ativos, uma vez que em todos os casos, o número de transferência dos dados da fonte para o repositório foi diminuído consideravelmente. Porém, o objetivo principal da estratégia proposta é permitir a redução do número de execuções do processo de transferência dos dados e manter um grau satisfatório de atualizações dos dados no repositório. Uma redução considerável nas atualizações não é interessante se houverem atrasos na transferência de informações importantes às análises efetuadas sobre o DW.

A estratégia ETA-PoCon deve então ser capaz de, além de diminuir as atualizações, transferir prioritariamente os dados considerados sensíveis. Para verificação desse requisito foram elaborados alguns experimentos que se diferenciam dos anteriores à medida que analisam não somente o total de atualizações executadas, mas também a semântica contida nos dados transferidos. O objetivo desses experimentos foi verificar se a aplicação da política por relevância não gerou atrasos na transferência de dados sensíveis decorrentes da diminuição na frequência de atualização do DW.

Foi utilizada a mesma metodologia aplicada nos experimentos anteriores, ou seja, inicialmente tanto a base fonte (SIVAT) quanto o DW possuem os mesmos dados, e os esquemas utilizados são apresentados na Figura 4.1. Para as análises sobre os dados transferidos será utilizado um relatório, também extraído do sistema SIVAT, que tem como objetivo sumarizar as 10 empresas com maior número de ocorrência de acidentes do trabalho.

Na Tabela 4.12, é apresentado um exemplo desse relatório extraído da base de dados utilizada nos experimentos. A base contém um total de 22693 acidentes e 10733 empresas, a terceira coluna da tabela representa o percentual que o total de acidentes de cada empresa representa em relação ao total de acidentes. Portanto, a empresa com maior número de acidente é a Empresa A com um total de 725 acidentes que representam 3,195% do total registrado na base de dados.

Tabela 4.12 - Relatório das 10 empresas com maior número de acidentes

Empresa	Total de Acidentes	Percentual
Empresa A	725	3,195%
Empresa B	365	1,608%
Empresa C	325	1,432%
Empresa D	268	1,181%
Empresa E	259	1,141%
Empresa F	221	0,974%
Empresa G	180	0,793%
Empresa H	171	0,754%
Empresa I	166	0,732%
Empresa J	158	0,701%

As empresas que constam no relatório apresentado constituem o foco dos trabalhos de vigilância executados pelos órgãos públicos responsáveis pela saúde do trabalhador. Uma alteração efetuada em qualquer uma dessas empresas deve ser transferida ao repositório o mais rápido possível. Portanto, no ambiente utilizado para os testes, os dados dessas empresas são considerados sensíveis, ou seja, alteram diretamente os resultados extraídos do repositório.

Nos experimentos descritos a seguir, a cada transferência dos dados, é realizada uma análise sobre os dados transferidos com intuito de verificar se algum dado de uma das 10 principais empresas foi alterado. Assim, é possível verificar se esses dados, que são de fato relevantes, estão sendo transferidos com prioridade e sem atrasos.

Na Tabela 4.13 são apresentados os parâmetros utilizados em todos os experimentos e na Tabela 4.14 as configurações das bases de dados e das transações executadas. Os quatro experimentos contaram com essa mesma configuração e as alterações se resumiram ao conjunto de dados alterados nas transações. A seguir são apresentados os resultados obtidos em cada um dos experimentos.

Tabela 4.13 - Parâmetros utilizados nos quatro experimentos

<i>Mapeamento(i)</i>	<i>Ti</i>	<i>R(Δi)</i>
2	10 segundos	0,5%

Tabela 4.14 – Configurações utilizadas nos quatro experimentos

Tuplas na tabela Ficha	Tuplas na tabela Empresa	Total de transações	Tempo total do experimento	Transações por Segundo
22693	10733	1000	10 Minutos	1,66

4.5.1. Experimento I

No experimento I o DW foi atualizado 12 vezes, enquanto sem a aplicação da estratégia seriam executadas 48 atualizações. Portanto, o percentual de redução no total de atualizações foi de 75%. Na Figura 4.13 é apresentado o gráfico do $R(\Delta)$ durante a execução do experimento.

A análise dos dados migrados a cada transferência permitiu identificar que entre as 12 atualizações executadas durante o experimento apenas em uma delas tiveram dados alterados das 10 principais empresas. Essa alteração ocorreu na fonte de dados entre os instantes 19 e 20 e foi transferida no instante 20. Na Figura 4.14 é apresentado o gráfico do $R(\Delta)$ com destaque para a migração de informações da “Empresa E”. Foram efetuadas 11 atualizações que não afetaram diretamente alguma das 10 empresas, isso porque a relevância calculada é referente a todo o delta.

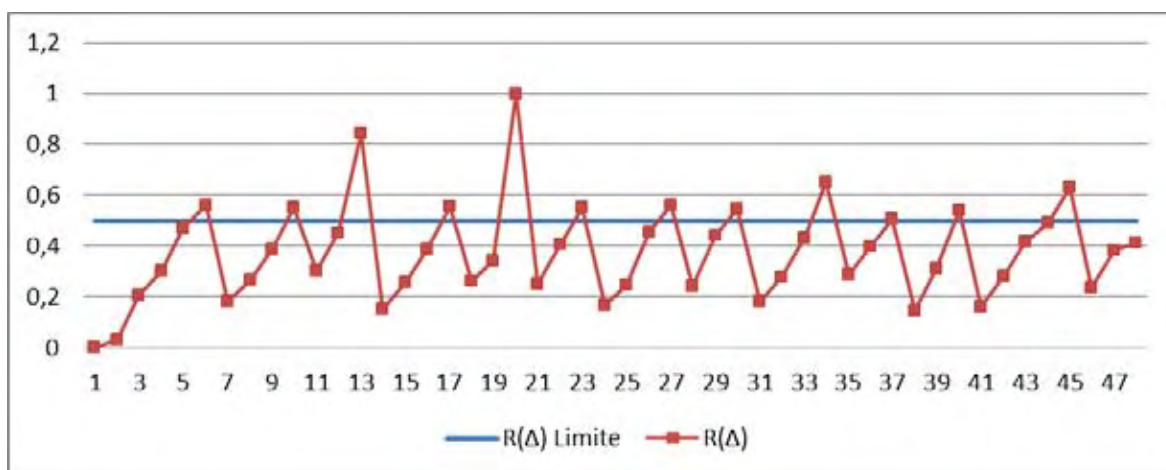


Figura 4.13 – Gráfico da evolução do $R(\Delta)$ durante o experimento I

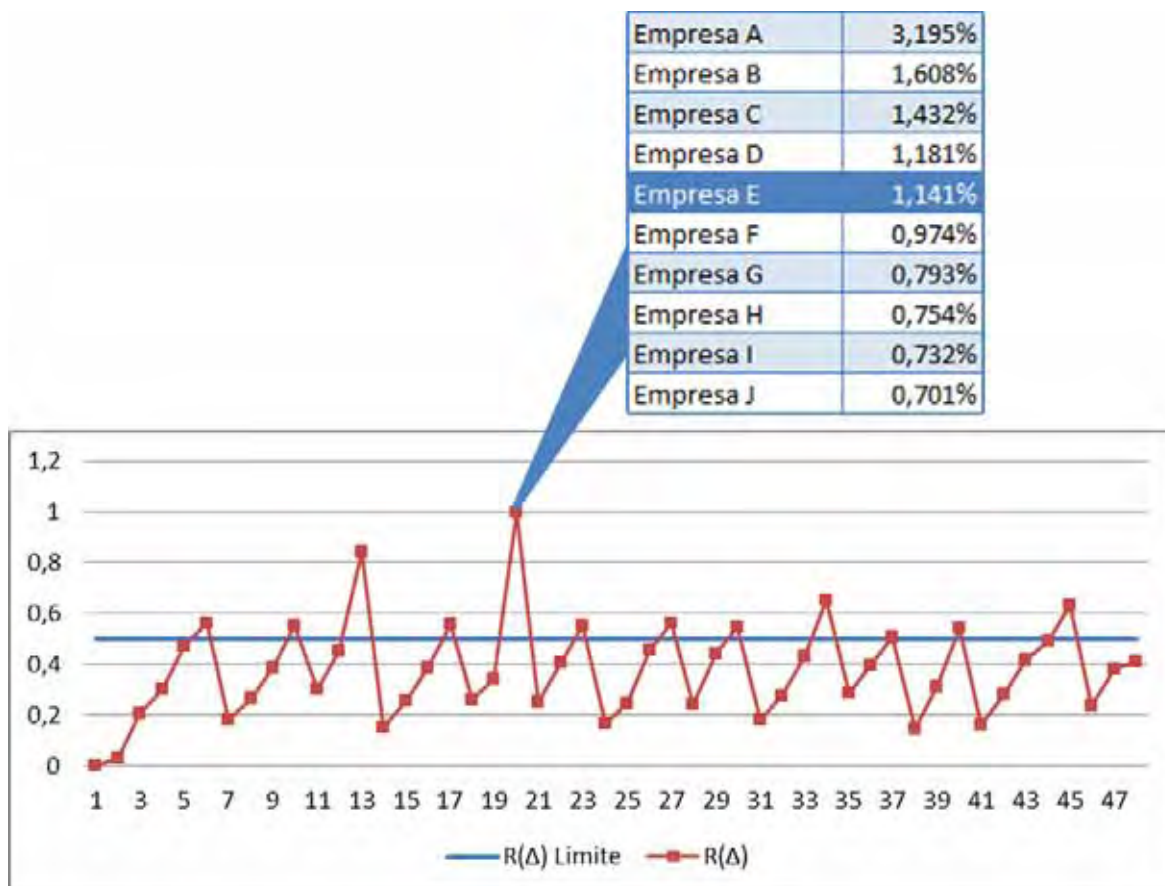


Figura 4.14 – Gráfico da evolução do $R(\Delta)$ durante o experimento I com destaque para os dados sensíveis transferidos no instante 20

4.5.2. *Experimento II*

Na Figura 4.15 é apresentado o gráfico da evolução do $R(\Delta)$ e destacam-se os dados transferidos nos instantes 22, 34 e 40. Com a estratégia, o total de atualizações foi reduzido em 62,7% uma vez que foram efetuadas 15 atualizações. Com a não utilização da estratégia, seriam efetuadas 48.

Nesse experimento, das 15 atualizações efetuadas, 3 dessas afetaram diretamente as 10 principais empresas, como destacado na Figura 4.15. Nos 3 casos a atualização da empresa alterada ocorreu em intervalos menores que um ciclo do mecanismo controlador, ou seja, em nenhum dos casos houve um atraso maior que 10 segundos para que um dado sensível fosse transferido ao repositório.

Nos instantes 20 e 40 é possível verificar que a relevância alcançada se aproximou de 1,5%. Isso se deve ao fato das transações simuladas nas fontes de dados afetarem empresas que representam 1,181% (Empresa D) e 1,432% (Empresa C) respectivamente.

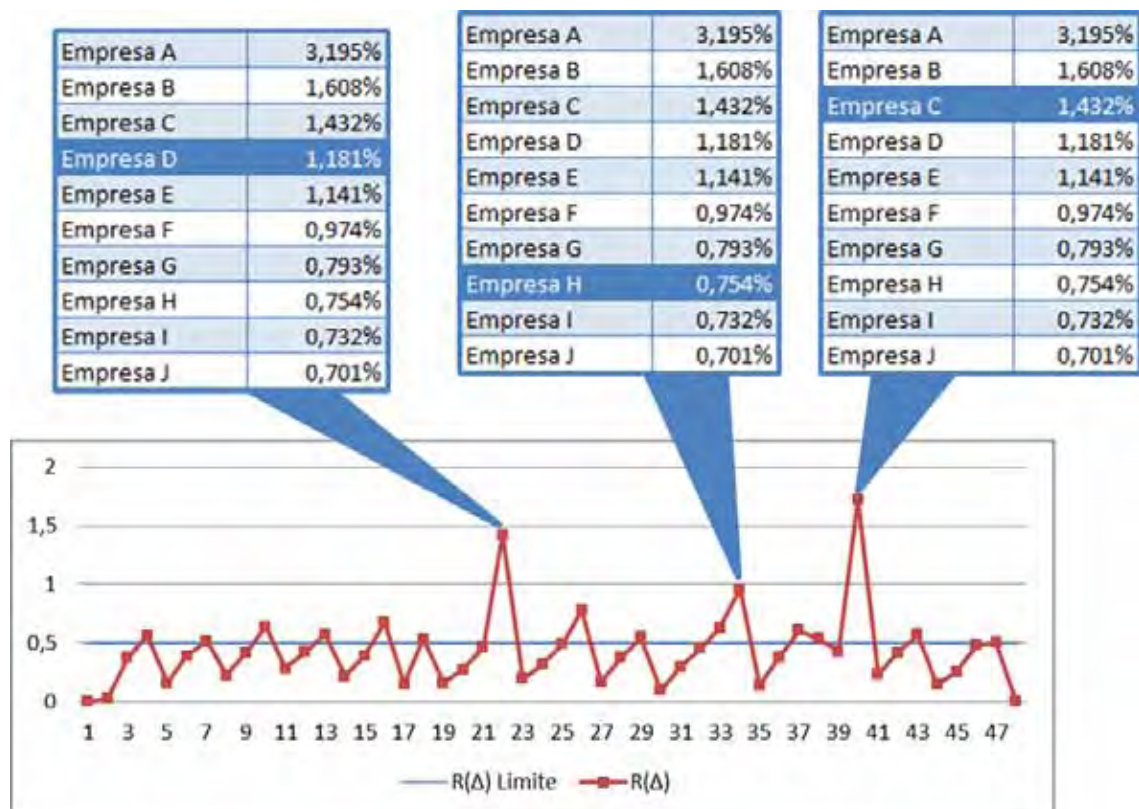


Figura 4.15 - Gráfico da evolução do $R(\Delta)$ durante o experimento II com destaque para os dados transferidos na instante 22, 34 e 40.

4.5.3. Experimento III

O gráfico da evolução do $R(\Delta)$ e o destaque da transferência de dados sensíveis são apresentados na Figura 4.16. Foram executadas 14 atualizações do DW que representam uma diminuição de 70% em relação ao total de 48 atualizações que seriam executadas sem a aplicação da estratégia. Durante o experimento houve apenas uma alteração sobre as 10 principais empresas e ocorreu entre os instantes 26 e 27 e foi transferido no instante 27.

Assim como nos outros experimentos, a transferência dos dados sensíveis não ultrapassou os 10 segundos referentes ao ciclo do mecanismo controlador. É possível observar que no instante 26 a relevância medida era pouco menor que 0,5%. Já no instante 27 a relevância teve um salto para aproximadamente 1,4%, esse crescimento da relevância decorre de operações efetuadas sobre dados da “Empresa I” que representa 0,732% do total de acidentes.

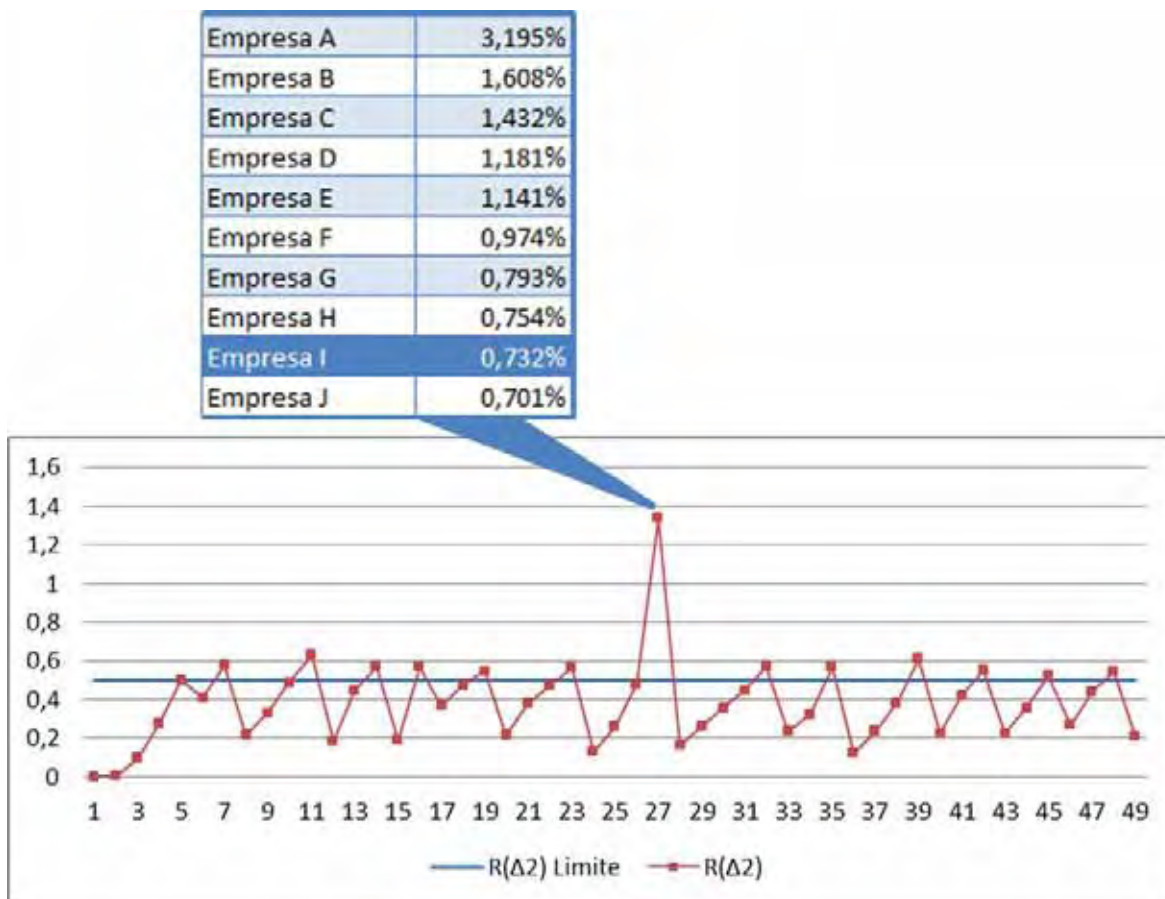


Figura 4.16- Gráfico da evolução do $R(\Delta)$ durante o experimento III com destaque para os dados transferidos na instante 27.

4.5.4. *Experimento IV*

Na Figura 4.17, é apresentado o gráfico da evolução do $R(\Delta)$ durante o experimento IV. Nesse último experimento, o total de atualizações do DW foi de 16, o que representou uma redução de 66% se comparada com as 48 atualizações que seriam executadas sem a aplicação da estratégia. No experimento, houve transferências de dados sensíveis nos instantes 13 e 20 e, assim como nos anteriores, o atraso entre a transação sobre as informações das empresas e a transferência desses para o repositório não ultrapassou os 10 segundos.

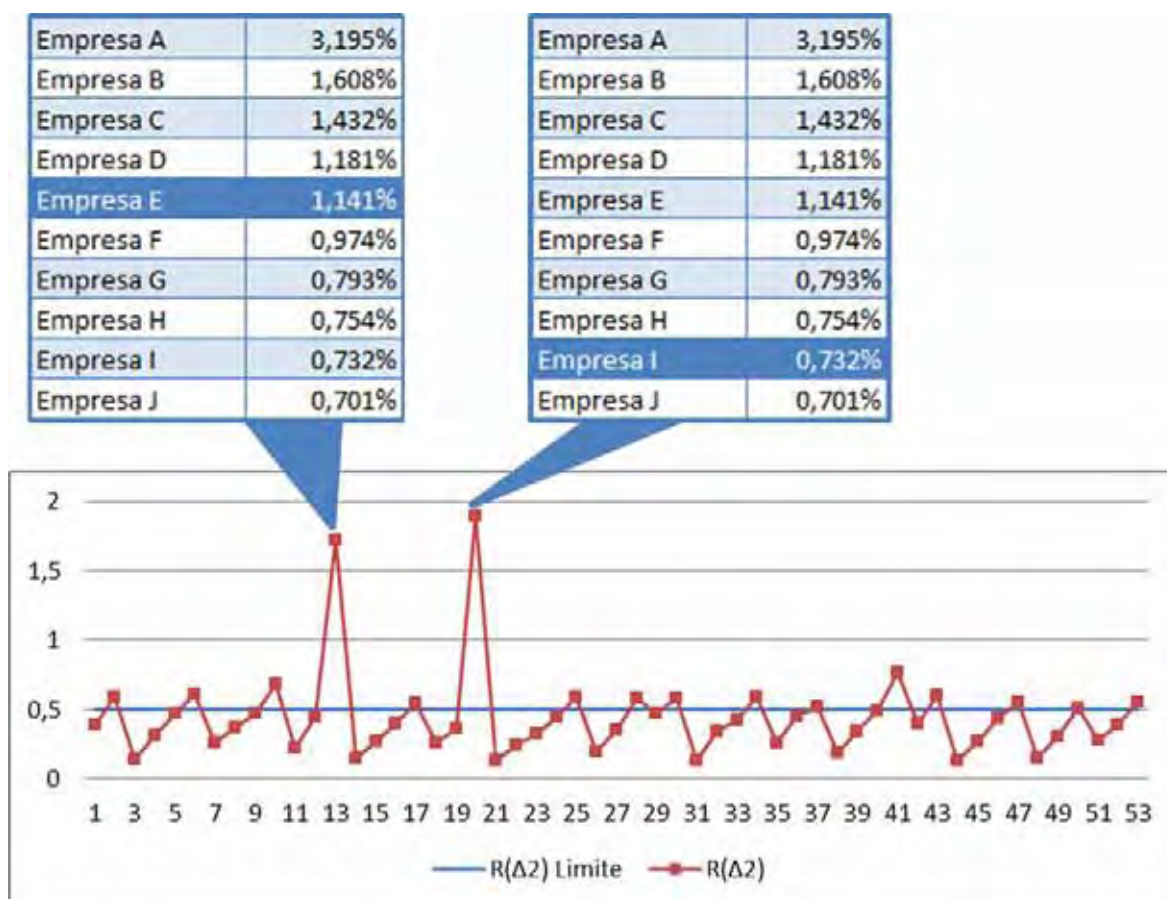


Figura 4.17 - Gráfico da evolução do $R(\Delta)$ durante o experimento IV com destaque para os dados transferidos nos instantes 13 e 20.

4.5.5. Discussão dos resultados

Nos quatro experimentos apresentados houve uma redução considerável no total de atualizações quando comparado com o total que seria atingido sem a utilização do parâmetro $R(\Delta)$, como pode ser verificado no gráfico apresentado na Figura 4.18. Os experimentos mostraram também que, mesmo com a redução no número de atualizações, os dados considerados sensíveis foram priorizados e não resultaram em atrasos.

As alterações de uma operação executada sobre uma das 10 principais empresas foi transferida ao repositório em no máximo 10 segundos, intervalo de tempo referente ao ciclo do mecanismo controlador e definido pelo parâmetro T .

Em alguns casos é possível verificar que nos instantes em que houve a transferência de dado sensível, a curva do gráfico do $R(\Delta)$ apresenta um pico maior que os demais, ou seja, a relevância alcançada com a alteração de uma das 10 principais empresas é relativamente maior que a relevância dos demais deltas. Essa característica demonstra que o valor de $R(\Delta)$ poderia ser aumentado mantendo o grau de atualização satisfatório.

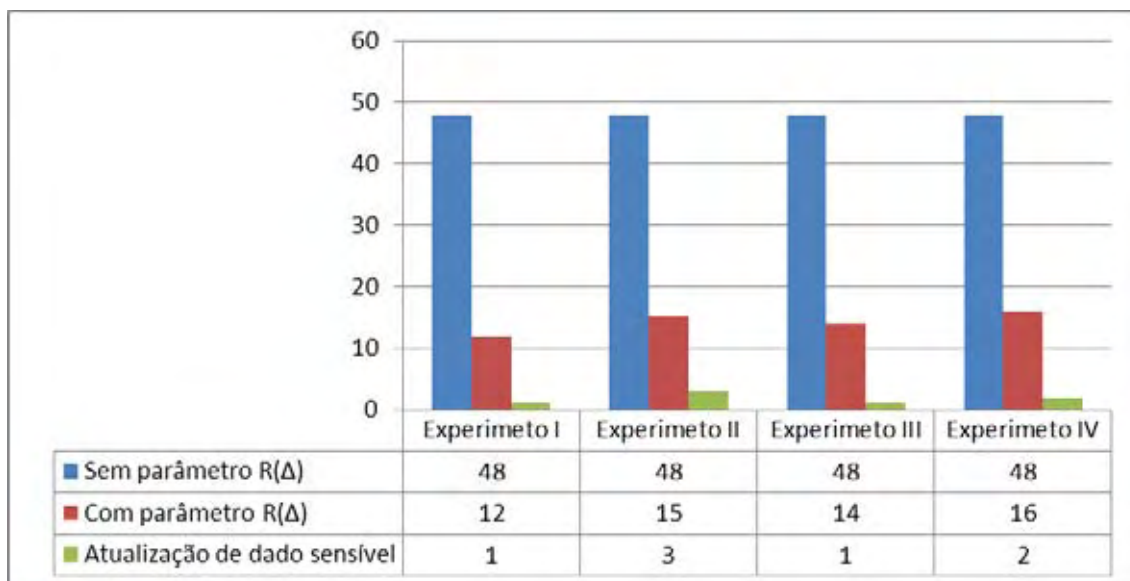


Figura 4.18 - Totais de atualizações executadas nos experimentos I, II, III e IV

Por outro lado, a escolha de um $R(\Delta)$ maior que 0,732% poderia ocasionar atrasos na transferência de informações referentes à “Empresa J”. Isso porque a relevância escolhida seria superior ao percentual representado por essa empresa, e, caso o delta seja composto somente por essa empresa, o $R(\Delta)$ não seria atingido e as informações não seriam transferidas. Vale ressaltar que a escolha do $R(\Delta)$ é feita pelo administrador do DW, que deve efetuar análises sobre as necessidades do negocio e os relatórios extraídos do repositório.

4.6. Considerações finais

Nesse capítulo foram apresentados os experimentos executados com intuito de verificar que a estratégia ETA-PoCon permite uma diminuição no total de atualizações e prioriza os dados que afetam de maneira direta os resultados extraídos no DW. O ambiente utilizado nos testes permitiu a simulação de um ambiente de DW ativo real, o que facilitou a validação da estratégia. Os resultados apresentados demonstraram que as políticas configuráveis podem ser aplicadas como auxílio na redução da probabilidade de sobrecarga sobre os sistemas envolvidos. Foram mostrados através dos testes que, uma vez configurada corretamente, a estratégia permite uma redução considerável no total de atualizações do DW e não ocasiona atrasos na migração de dados sensíveis.

Capítulo 5 Conclusões

O objetivo do trabalho foi a elaboração de uma estratégia para execução do processo ETA que permita uma redução na frequência de atualização de ambientes de DWA com intuito de reduzir a sobrecarga sobre os sistemas envolvidos. A diminuição no número de atualizações é baseada em políticas configuráveis que visam priorizar dados sensíveis.

A fim de fundamentar o trabalho, inicialmente foi apresentado um panorama geral dos trabalhos desenvolvidos relacionados à construção de DWA, o que evidenciou a presença do problema relacionado à sobrecarga dos sistemas envolvidos e a falta de propostas para o tratamento desse problema.

A estratégia ETA-PoCon foi descrita em detalhes e a ideia central do trabalho foi definir as políticas de propagação de dados. Para tanto, foi escolhida a utilização do volume e da relevância do delta além da definição de que cada mapeamento entre as fontes e o repositório deve ser tratado de forma diferente com frequência de atualização específica.

Para definição do grau de relevância do delta, foi definida uma metodologia baseada no impacto causado por cada tupla de uma fonte de dados ao ser transferida ao repositório. No cálculo da relevância são efetuadas medições à estrutura do banco de dados com intuito de verificar o percentual de tuplas que referenciam o delta. Já a política baseada em volume necessita apenas da execução de uma operação de contagem ao total de tuplas que constituem o delta.

O trabalho contou também com a construção da ferramenta FETA que implementa os requisitos definidos na estratégia. A arquitetura da ferramenta é constituída por módulos que permitem o mapeamento entre as fontes e o repositório além do controle sobre os parâmetros T , $V(\Delta)$ e $R(\Delta)$ de cada um dos mapeamentos. Desse modo, foi possível construir experimentos para verificação da estratégia proposta.

Os experimentos elaborados geraram resultados que permitiram verificar que as políticas propostas oferecem uma redução considerável no total de atualizações de DWA sem acarretar em atrasos na migração de dados sensíveis. Desse modo, é possível afirmar que o objetivo do trabalho foi atingido. A seguir, são descritas algumas conclusões importantes sobre a estratégia ETA-PoCon:

- A política baseada na análise do parâmetro $V(\Delta)$ se mostrou bastante promissora quanto à redução de atualizações. Porém, sua utilização deve contar com uma forte análise sobre os requisitos do negócio ao qual é aplicada. A análise deve focar principalmente na frequência de transações executadas nas fontes;
- Devido à forte ligação entre o $V(\Delta)$ e as transações executadas, a política por volume gera melhores resultados quando aplicada em ambientes em que não há grandes oscilações na taxa de inserção de dados;
- A política baseada em relevância também se mostrou interessante à redução das atualizações. Como $R(\Delta)$ é relativo ao repositório, a definição desse parâmetro pode contar com análises sobre os próprios relatórios extraídos do DW;
- O $R(\Delta)$ não tem relação direta com o $V(\Delta)$. Dessa forma, o uso de política por relevância independe da taxa de transações executadas sobre as fontes de dados;
- O cálculo do $R(\Delta)$ é realizado por meio de consultas aos meta-dados e aos dados do repositório. Dessa forma, a estratégia pode ser aplicada a diferentes contextos independentemente da semântica contida nas tuplas.
- Apesar do trabalho ter sido validado por meio da aplicação da ferramenta FETA, a estratégia definida pode ser implementada em outras ferramentas ETA, uma vez que as políticas definidas se baseiam em consultas ao delta, aos metadados e dados do repositório;

Portanto, o trabalho desenvolvido contribui significativamente à construção de ferramentas ETA para atualizações de DWA, uma vez que oferece uma alternativa ao tratamento do problema da sobrecarga dos sistemas envolvidos. Vale ressaltar que esse problema é citado em vários dos trabalhos correlatos apresentados, porém, somente o

mecanismo definido por Che [CHE_10] descreve uma possível solução. Na Tabela 5.1 é apresentado um comparativo entre a estratégia desenvolvida e o trabalho descrito por Che.

Tabela 5.1 – Comparação entre o trabalho desenvolvido e o mecanismo descrito por Che [Che_10]

<i>Característica</i>	<i>Che [CHE_10]</i>	<i>ETA-PoCon</i>
Análise do impacto da transferência dos dados	Sim	Sim
Política baseada em tempos distintos a cada mapeamento	Não	Sim
Análise de Volume	Não	Sim
Análise da Relevância	Baseada no conteúdo	Baseada na estrutura
Análise das consultas executadas no DW	Sim	Não

Dentre as diferenças entre os trabalhos, a que mais se destaca é a diferença na análise sobre a relevância. Che faz uso de um mecanismo que analisa o impacto causado pelo delta baseado no conteúdo dos dados, ou seja, a relevância é calculada com base nas próprias informações das tuplas, o que dificulta a aplicação do mecanismo em diferentes contextos. Como já mencionado, a estratégia proposta permite sua aplicação em diferentes ambientes, uma vez que a relevância é calculada por meio dos metadados e da estrutura do repositório. Assim a comparação com esse trabalho reforça a contribuição oferecida pelo trabalho desenvolvido.

5.1. Trabalhos Futuros

Com base na análise sobre o trabalho desenvolvido, são descritas algumas sugestões de trabalhos futuros:

- Os parâmetros $V(\Delta)$ e $R(\Delta)$ são definidos pelo usuário. A elaboração de uma metodologia para cálculo automático desses parâmetros traria uma contribuição significativa;
- A política por relevância analisa todas as tabelas que referenciam a tabela destino (Tref). A fim de melhorar os resultados obtidos poderiam ser atribuídos pesos distintos para cada tabela do Tref. Dessa forma, o administrador do DW teria mais

uma opção para configurar a estratégia de cada mapeamento e poderia assim melhor a priorização de dados sensíveis;

- Permitir combinações entre os parâmetros $V(\Delta)$ e $R(\Delta)$. Na estratégia, a combinação entre os dois parâmetros é feita por meio do operador lógico “OU”. O trabalho poderia ser estendido e permitir a utilização de outros operadores, com intuito de aumentar ainda mais o suporte à redução no total de atualizações do DW;
- As políticas de tempo e relevância foram baseadas em necessidades do mundo corporativo, a aplicação da estratégia ETA-PoCon em bases de dados científicas poderia gerar novos requisitos e consequentemente novas políticas de propagação;
- O trabalho utilizado no quadro comparativo da Tabela 5.1 descreve uma política baseada na análise na frequência das consultas sobre o DW. A combinação dessa análise com os parâmetros definidos na estratégia pode gerar resultados interessantes.

Referências Bibliográficas

- [ALA_09] Alalwan, N. et al. 2009. Generating OWL Ontology for Database Integration. *2009 Third International Conference on Advances in Semantic Processing*. (Oct. 2009), 22-31.
- [BOR_11] Bornea, M.A. et al. 2011. Semi-Streamed Index Join for near-real time execution of ETL transformations. *2011 IEEE 27th International Conference on Data Engineering* (Apr. 2011), 159-170.
- [BRO_02] Brobst, S. 2002. Active data warehousing: a new breed of decision support. *Proceedings. 13th International Workshop on Database and Expert Systems Applications* (2002), 769.
- [CHE_10] Chen, L. et al. 2010. Towards Near Real-Time Data Warehousing. *2010 24th IEEE International Conference on Advanced Information Networking and Applications* (2010), 1150-1157.
- [FAN_12] Fan, Y. 2012. The Research of Active Data Warehouse Based on Multi-Agent. *2012 Spring Congress on Engineering and Technology* (May. 2012), 1-4.
- [GUE_11] Guerra, J. et al. 2011. Creating a Real Time Data Warehouse. *Time*. (2011).
- [HAL_06] Halevy, A. and Ordille, J. 2006. Data Integration: The Teenage Years. *VLDB '06 Proceedings of the 32nd international conference on Very large data bases*. (2006), 9-16.
- [JAR_03] Jarke, M. et al. 2003. *Review by: Fundamentals of Data Warehouses*. Wiley.
- [JAV_10] Javed, M.Y. and Nawaz, A. 2010. Data Load Distribution by Semi Real Time Data Warehouse. *2010 Second International Conference on Computer and Network Technology* (2010), 556-560.
- [KIM_04] Kimball, R. and Caserta, J. 2004. *The Data Warehouse ETL Toolkit: Practical Techniques for Extracting, Cleaning, Conforming, and Delivering Data*. John Wiley & Sons.

- [LUJ_04] Luján-Mora, S. et al. 2004. Data mapping diagrams for data warehouse design with UML. *Conceptual Modeling–ER 2004*. (2004), 191–204.
- [NGU_06] Nguyen, T.M. and Tjoa, A.M. Zero-latency data warehousing (ZLDWH): the state-of-the-art and experimental implementation approaches. *2006 International Conference on Research, Innovation and Vision for the Future* 167-176.
- [QIA_09] Qian, Z. and Li-jun, S. 2009. The Architecture and Design Strategy for Data Warehouse of Highway Management. *2009 Second International Conference on Intelligent Computation Technology and Automation*. (2009), 459-462.
- [RIZ_06] Rizzi, S. et al. 2006. Research in data warehouse modeling and design. *Proceedings of the 9th ACM international workshop on Data warehousing and OLAP - DOLAP '06* (New York, New York, USA, 2006), 3.
- [SAL_08] Salhi, N. et al. 2008. An Ontology and a Description Schema Base for Relational Database Integration. *2008 International Workshop on Advanced Information Systems for Enterprises* (Apr. 2008), 3-10.
- [SAN_08] Santos, V. R.. Ferramenta de apoio a integração entre base de dados: Gerenciador de conflitos e transações. 2008. 48 f. Monografia (Bacharelado) - Curso de Ciência da Computação, Departamento Ciências de Computação e Estatística, Universidade Estadual Paulista “Julio de Mesquita Filho”, São José do Rio Preto, Brasil, 2008.
- [SCA_09] Scarpelini Neto, P. Ferramenta de apoio à integração de dados utilizando ontologias. 2009. 70 f. Monografia (Bacharelado) - Curso de Ciência da Computação, Departamento Ciências de Computação e Estatística, Universidade Estadual Paulista “Julio de Mesquita Filho”, São José do Rio Preto, Brasil, 2009.
- [SHI_09] Shi, J. et al. 2009. Priority-Based Balance Scheduling in Real-Time Data Warehouse. *2009 Ninth International Conference on Hybrid Intelligent Systems*. (2009), 301–306.
- [SIM_10] Simitsis, A. et al. 2010. Partitioning real-time ETL workflows. *2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010)* (2010), 159–162.

- [SON_10] Song, J. et al. 2010. A Triggering and Scheduling Approach for ETL in a Real-time Data Warehouse. *2010 10th IEEE International Conference on Computer and Information Technology* (Jun. 2010), 91–98.
- [SUN_12] Sun, K. and Lan, Y. 2012. SETL: A scalable and high performance ETL system. *2012 3rd International Conference on System Science, Engineering Design and Manufacturing Informatization*. (Oct. 2012), 6–9.
- [THO_10] Thomas Jörg and Stefan Dessloch 2010. Near real-time data warehousing using state-of-the-art ETL tools. *Enabling Real-Time Business Intelligence*. 41, (2010), 100-117.
- [VAS_09] Vassiliadis, P. 2009. Near real time etl. *New Trends in Data Warehousing and Data*. 3, (2009).
- [VIQ_11] Viqarunnisa, P. et al. 2011. Generic data model pattern for data warehouse. *Proceedings of the 2011 International Conference on Electrical Engineering and Informatics* (Jul. 2011), 1–8.
- [XIA_10] Xiaoli, W. and Yuan, Y. 2010. XML-based heterogeneous database integration system design and implementation. *2010 3rd International Conference on Computer Science and Information Technology* (Jul. 2010), 547-550.
- [XU_11a] Xu, Q. and Sun, Q. 2011. The Research of Information Sharing Platform Based on Data Warehouse in Fossil Power Plant. *2011 International Conference on Computational and Information Sciences*. (Oct. 2011), 227–229.
- [XU_11b] Xu, L. et al. 2011. A PaaS based metadata-driven ETL framework. *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*. (Sep. 2011), 477–481.
- [YUN_10] Yunpeng, L. and Meiyun, X. 2010. Research of heterogeneous database integration based on XML. *2010 International Conference on Mechanical and Electrical Technology* (Set. 2010), 793–796.
- [ZHE_09] Zhenyou, Z. et al. 2009. Research of Heterogeneous Database Integration Based on XML and JAVA Technology. *2009 International Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Government* (2009), 275-278.

[ZHU_08] Zhu, Y. et al. 2008. Data Updating and Query in Real-Time Data Warehouse System. *2008 International Conference on Computer Science and Software Engineering* (2008), 1295-1297.

Autorizo a reprodução xerográfica para fins de pesquisa.

São José do Rio Preto, ____/____/____

Assinatura