Um Método para Elaborar Bancos de Dados Analítico-Temporais para o Armazenamento de Históricos de Dados

Alex Sandro Romeo de Souza Poletto[1], Jorge Rady de Almeida Júnior[2]

[1]Centro de Pesquisas em Informática - Instituto Municipal de Ensino Superior de Assis
Fundação Educacional do Município de Assis (FEMA)
Av. Getúlio Vargas, 1200 - CEP 19.807-634 – Assis - SP – Brasil

[2]Departamento Sistemas Digitais – Escola Politécnica – Universidade São Paulo (USP)
Av. Prof. Luciano Gualberto, 158. CEP: 05508-900 - São Paulo, SP - Brasil

apoletto@femanet.com.br, jorge.almeida@poli.usp.br

Abstract*. This work describes a method whose main objective is to store historic data, resulting in the Analytic-Temporal Databases. The origins of the data are the Operational Databases. This model can aid in the elaboration of the Analytic-Temporal Databases that constitutes a very adequate foundation to help in the decision taking process, audits and data recovery. This work contains two stages. The first stage aims to manually help the modeling of Analytical-Temporal Database, based on models of data from Operational Databases. The second stage aims to provide automatic mechanisms, used in Database Management Systems, providing generation, transport and storage of analytical-temporal data, using triggers and stored procedures.*

Resumo. *O presente trabalho descreve um método para, a partir de Bancos de Dados Operacionais, elaborar Bancos de Dados Analítico-Temporais, cujo principal objetivo é o de armazenar históricos de dados para auxiliar no processo de tomada de decisões, auditorias e recuperação de dados. O trabalho contém duas etapas. A primeira etapa tem por objetivo auxiliar, manualmente, a modelagem do Banco de Dados Analítico-Temporal, com base nos modelos de dados dos Bancos de Dados Operacionais. Na segunda etapa objetivou-se disponibilizar mecanismos automáticos, explorados nos próprios Sistemas Gerenciadores de Banco de Dados, que possibilitem a geração, o transporte e o armazenamento dos dados Analítico-Temporais, usando gatilhos e procedimentos armazenados.*

1. Introdução

O objetivo deste trabalho é propor um método para elaboração de Bancos de Dados Analítico-Temporais (BDAT), os quais, por sua vez, visam servir de alicerce aos processos de tomada de decisões, auditorias e recuperação de dados. Os dados a serem inseridos nesses bancos de dados serão originados exclusivamente dos Bancos de Dados Operacionais.

Normalmente, a finalidade dos Bancos de Dados Operacionais é armazenar somente um único valor dos dados que, por sua vez, deve ser o mais recente. A necessidade de possuir históricos de dados levou ao surgimento dos chamados Bancos de Dados Analítico-Temporais, que representam uma opção a ser utilizada para o armazenamento do estado evolutivo dos dados, no sentido também, de evitar sobrecarga

nas operações operacionais, quando da necessidade de recuperar históricos de dados [Cordeiro; Santos; Edelweiss e Galante 2004].

Recentemente surgiu a tecnologia de Oracle Flashback, que tem a capacidade de consultar dados históricos, realizar análises, alterar e realizar a auto-reparação do serviço para recuperar corrupções lógicas, para monitoramento de alterações, bem como recuperação e retorno em períodos de tempo das exclusões de dados, mas todo esse processo é realizado diretamente nos Bancos de Dados Operacionais [Bryla e Loney 2008].

A criação de um BDAT é importante, já que além de oferecer um armazenamento mais completo e rico (especializado) em relação aos Bancos de Dados Operacionais, tem como propósito específico prestar auxílio aos responsáveis pelas tomadas de decisões, auditorias e recuperação de dados. Para se ter um melhor rendimento nessas atividades, e no sentido de não atrapalhar as operações operacionais, é necessário que esse banco de dados seja separado logicamente dos Bancos de Dados Operacionais [Sprague e Watson 1991].

Uma das formas disponíveis para a criação de Bancos de Dados Analítico-Temporais dá-se por meio do processo de *Data Warehousing*. No entanto, deve-se considerar a grande complexidade dos mecanismos de extração, transformação e carga de dados envolvidos na implementação de um *Data Warehouse*, chegando a representar até 80% do esforço total de desenvolvimento [Vassiliadis; Simitsis e Skiadopoulos 2002].

Todo esse panorama motivou o surgimento deste trabalho, cujo objetivo é possibilitar o desenvolvimento de mecanismos e técnicas que auxiliem na modelagem, geração e armazenamento de dados, resultando, assim, em um BDAT.

Este artigo está organizado da seguinte forma: na Seção 2, é descrito o método proposto; na Seção 3, é apresentado um estudo de caso; na Seção 4, são apresentados alguns trabalhos correlatos e, finalmente, na última seção são relatadas as considerações finais.

2. Descrição do Método Proposto

Para a realização do método proposto, dividiu-se o mesmo em duas etapas, conforme ilustrado na Figura 1.

A Etapa A diz respeito à realização de uma análise do Modelo de Dados Operacional no sentido de gerar um Modelo de Dados Analítico-Temporal para a construção do BDAT. Nessa etapa, há preocupações quanto a selecionar os conjuntos de entidades e atributos que comporão o BDAT, bem como quanto a definir os conjuntos de entidades a serem criados para o armazenamento dos dados analítico-temporais. Essa etapa constitui as atividades responsáveis pela modelagem do BDAT. É importante salientar que os Modelos de Dados Operacionais em funcionamento, não sofrerão modificações, e permanecerão implementados em seu formato original.

A Etapa B refere-se à especificação genérica de gatilhos e procedimentos armazenados, os quais serão responsáveis por enviar os dados dos Bancos de Dados Operacionais para o BDAT, devendo ter por base tanto os modelos de dados operacionais, quanto o modelo de dados analítico-temporal.

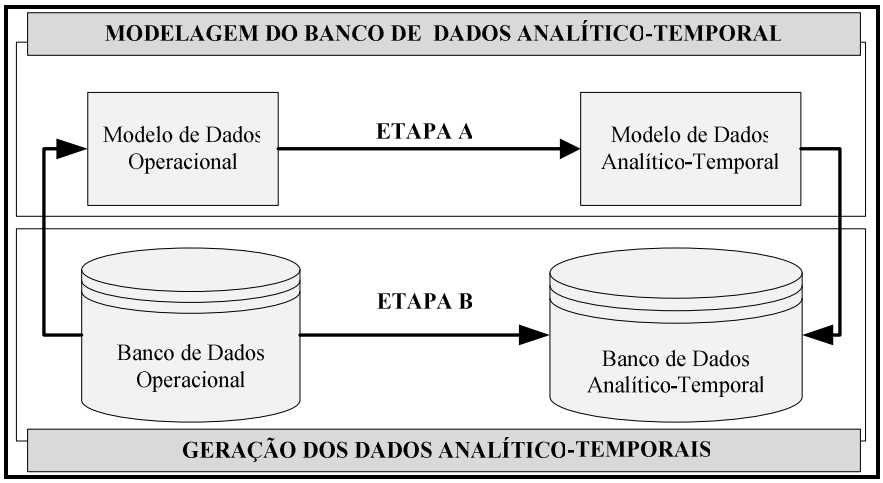

Figura 1. Diagrama geral do Método Proposto.

O controle do estado evolutivo dos dados será realizado com a inclusão de "elementos de tempo" (*timestamp*) nos conjuntos de entidades a serem criados no BDAT. Esses "elementos de tempo" têm a função de representar as épocas exatas das modificações dos dados ao longo do tempo, envolvendo tanto o instante inicial de validade quanto o instante em que tal dado sofrer alterações, que por sua vez será gerado automaticamente pelo Sistema Gerenciador de Banco de Dados.

2.1. Descrição da Etapa A: Modelagem do BDAT

Em se tratando de modelar dados analítico-temporais, um modelo de dados deve apresentar as características básicas de Bancos de Dados Convencionais (Operacionais), acrescentando a possibilidade de representar dados que se alterem ao longo do tempo. Em termos gerais, bancos de dados que mantêm dados correntes, bem como dados passados, são designados Banco de Dados Analítico-Temporais [Tansel 1997].

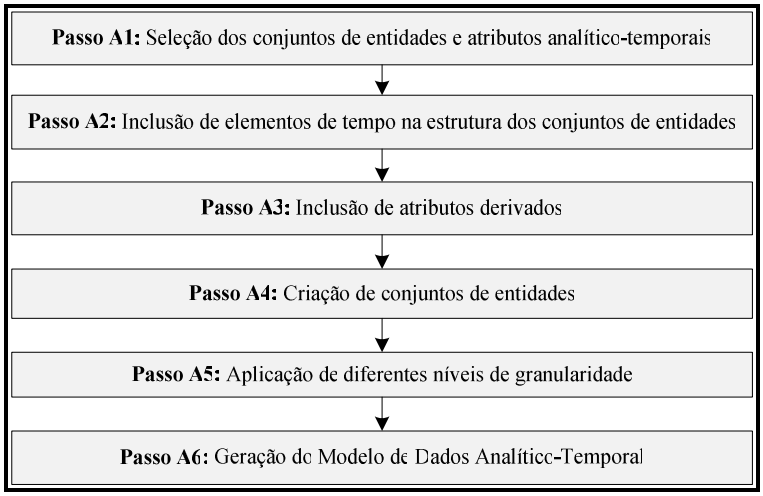

Figura 2. Passos da Etapa A.

De maneira geral, esta etapa visa a preparação manual do Modelo de Dados Analítico-Temporal com base no Modelo de Dados Operacional. O objetivo deste último é criar um esquema que ofereça condições para o armazenamento de dados analítico-temporais. Os objetivos desta etapa são: selecionar os conjuntos de entidades e

atributos essenciais para a geração de dados analítico-temporais; adicionar atributos derivados; transformar relacionamentos entre dados em conjuntos de entidades; acomodar diferentes níveis de granularidade; e controlar os períodos nos quais os dados operacionais ficaram válidos, bem como registrar atributos pré-calculados, no sentido de evitar sobrecargas de processamento no BDAT, quando da necessidade de recuperar informações que envolvam cálculos. A seguir, apresenta-se uma descrição dos passos propostos para esta etapa.

Passo A1: Seleção dos conjuntos de entidades e atributos analítico-temporais

Este passo tem a função de identificar, no Modelo de Dados Operacional, quais são os conjuntos de entidades e atributos essenciais para a geração dos dados analítico-temporais, a fim de que os mesmos possam ser incluídos na elaboração do Modelo de Dados Analítico-Temporal. Além do que, será possível verificar quais conjuntos de entidades são puramente operacionais, uma vez que dados puramente operacionais não são incluídos no processo de geração de dados analítico-temporais. Na realidade, tanto a seleção dos conjuntos de entidades quanto a seleção dos atributos devem ocorrer de maneira simultânea, já que quando um atributo é selecionado, determinará que o conjunto de entidades também o seja.

Passo A2: Inclusão de "elementos tempo" na estrutura dos conjuntos de entidades

É primordial, em alguns casos, a inclusão de atributos que controlarão os períodos (intervalo de datas) nos quais os valores contidos nos conjuntos de entidades permanecerão válidos. Ou seja, em um período, determinado valor foi considerado o mais recente, ou válido. Para que isso seja possível, é necessária a inclusão de novos atributos "timestamp" para o controle dos períodos. Desta forma, a função deste passo é incluir, no BDAT, um ou mais atributos "timestamp", na estrutura dos conjuntos de entidades "selecionados", a fim de possibilitar o armazenamento exato dos dados analítico-temporais, podendo ou não, compor a chave primária existente. Pode ser necessária também, a adição de um novo conjunto de entidades, para que haja o armazenamento mais detalhado de todas as ocorrências de alteração de valores.

Passo A3: Inclusão de atributos derivados

Em alguns casos, é preciso tomar decisões em um curto espaço de tempo. Pode ser necessário realizar consultas muito complexas que impliquem acessar um grande volume de tuplas, com inúmeros cálculos. Isso poderá atrasar a recuperação das informações, daí a necessidade da adição de atributos derivados, já que tal procedimento pode eliminar a necessidade de realizar cálculos durante o processo de recuperação de valores. Portanto, a função deste passo é adicionar "atributos totalizadores", derivados de outros atributos, cujo objetivo é acelerar a recuperação de consultas que envolvam cálculos.

Passo A4: Criação de conjuntos de entidades

Este passo é justificado pelo fato de que, em alguns casos, pode ocorrer a necessidade de armazenar "atributos multivalorados", ou seja, atributos ou conjuntos de atributos que precisem armazenar muitos valores para uma mesma entidade proprietária. Portanto, a função deste passo é oferecer um meio para a representação dos vários relacionamentos entre os dados analítico-temporais.

Passo A5: Aplicação dos diferentes níveis de granularidade

A função deste passo é estabelecer, junto aos usuários, qual será o nível necessário de detalhe ou de sumarização para os dados a serem transportados ao BDAT, para que, futuramente, seja possível responder às diversas solicitações de consultas dos usuários.

Passo A6: Geração do Modelo de Dados Analítico-Temporal

A função deste passo é apresentar o Modelo de Dados Analítico-Temporal resultante desta etapa, e que culminará na criação do BDAT. No método aqui proposto, há uma tabela chamada Analítico-Temporal, que deverá conter os seguintes atributos: uma chave primária, atributos descritivos, atributos numéricos, atributos temporais e atributos derivados, conforme ilustrado na Figura 3.

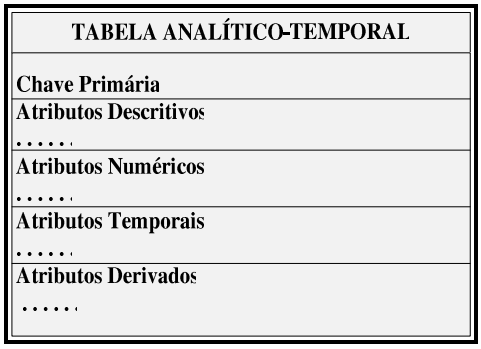

Figura 3. Tabela Analítico-Temporal.

A intenção é que essa Tabela Analítico-Temporal seja utilizada como um padrão para a criação de toda a estrutura de dados do BDAT.

2.2. Descrição da Etapa B: Geração dos Dados Analítico-Temporais

Esta etapa tem por objetivo a geração, o transporte e o armazenamento de dados analítico-temporais, de forma dinâmica e automática. Tais dados são originados dos Bancos de Dados Operacionais para o BDAT, por intermédio de gatilhos e procedimentos armazenados instalados nos Bancos de Dados Operacionais e de procedimentos armazenados instalados no BDAT.

Para a realização dinâmica e automática dessa geração, é necessário algum controle sobre o Banco de Dados Operacional, que permita, a qualquer momento, a aquisição e o transporte dos dados para o BDAT. Por essa razão é que são utilizados gatilhos e procedimentos armazenados, visto que são os recursos dos Sistemas Gerenciadores de Bancos de Dados que permitem acompanhar as atualizações na base operacional e enviá-las para a base analítico-temporal [Italiano e Ferreira 2006].

Para auxiliar esse processo, são oferecidas, nesta etapa, duas especificações genéricas, uma de um gatilho e outra de um procedimento armazenado, que deverão ser utilizados como *templates* para as codificações dos gatilhos e procedimentos armazenados necessários para a geração, transporte e armazenamento dos dados junto ao BDAT [Poletto e Almeida Jr. 2007], conforme descrito a seguir:

2.2.1. Especificação Genérica do Gatilho

A finalidade dessa especificação é oferecer um *template* genérico de um gatilho, que deverá servir de modelo para a codificação dos gatilhos junto ao Banco de Dados

Operacional, que serão os responsáveis pela geração, transporte e armazenamento dos dados operacionais ao BDAT. Esses gatilhos deverão ser executados automaticamente pelo Sistema Gerenciador de Banco de Dados imediatamente após ocorrer alguma operação de inclusão ou alteração nos conjuntos de entidades e/ou atributos do Banco de Dados Operacional, selecionados para a geração dos dados analítico-temporais.

```
CREATE OR REPLACE TRIGGER TRIGGER_TEMPLATE
BEFORE/AFTER INSERT OR UPDATE OF BDO_Atributo_Y ON BDO_Tabela
FOR EACH ROW
[DECLARE
     Variavel_1 BDO_Tabela_X.BDO_Atributo_X%TYPE;
     Variavel_2 BDO_Tabela_X.BDO_Atributo_X%TYPE; ...;]
BEGIN
  IF INSERTING THEN
    [SELECT BDO_Atributo_1, BDO_Atributo_2, . . . INTO Variavel_1, Variavel_2, . . .
        FROM BDO_Tabela_X WHERE BDO_Chave_X = :NEW.BDO_Chave_2;]
    INSERT INTO BDAT_Tabela (BDAT_Chave_1, BDAT_Chave_2, . . .,
       BDAT_Atributo_1, BDAT_Atributo_2, BDAT_Atributo_3, BDAT_Atributo_4,. . .,
       BDAT_Atributo_Pre_1, BDAT_Atributo_Pre_2, BDAT_Atributo_Pre_3,
       BDAT_Atributo_Pre_4, BDAT_Atributo_Data_1)
       VALUES(:NEW.BDO_Chave_1, NEW.BDO_Chave_2, . . .,
       :NEW.BDO_Atributo_1, :NEW.BDO_Atributo_2, . . ., [Variavel_1, Variavel_2], . . .,
       :NEW.BDO_Atributo_Calc_1*:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1+:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1/:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1-:NEW.BDO_Atributo_Calc_2, Data_Dia);
    ELSE
    [IF :NEW.BDO_Chave_2 <> :OLD.BDO_Chave_2  then
       SELECT BDO_Atributo_1, BDO_Atributo_2, . . . INTO Variavel_1, Variavel_2, . . .
          FROM BDO_Tabela_X WHERE BDO_Chave_X = :NEW.BDO_Chave_2;
     ELSE
       SELECT BDO_Atributo_1, BDO_Atributo_2, . . . INTO Variavel_1, Variavel_2, . . .
          FROM BDO_Tabela_X WHERE BDO_Chave_X = :OLD.BDO_Chave_2;
       ENDIF;]
      UPDATE BDAT_Tabela SET BDAT_Atributo_Data_2=Data_Dia
         WHERE BDAT_Tabela.BDAT_Atributo_X=:OLD. BDAT_Atributo_X AND
         BDAT_Tabela.BDAT_Chave_1=:OLD.BDO_Chave_1 AND
         BDAT_Tabela.BDAT_Chave_2=:OLD.BDO_Chave_2 AND . . . .
         AND BDAT_Tabela.BDAT_Atributo_Data_2 is NULL;
    INSERT INTO BDAT_Tabela (BDAT_Chave_1, BDAT_Chave_2, . . .,
       BDAT_Atributo_1, BDAT_Atributo_2, BDAT_Atributo_3, BDAT_Atributo_4,. . .,
       BDAT_Atributo_Pre_1, BDAT_Atributo_Pre_2, BDAT_Atributo_Pre_3,
       BDAT_Atributo_Pre_4, BDAT_Atributo_Data_1)
       VALUES(:OLD.BDO_Chave_1, OLD.BDO_Chave_2, . . .,
       :NEW.BDO_Atributo_1, :NEW.BDO_Atributo_2, . . ., [Variavel_1, Variavel_2], . . .,
       :NEW.BDO_Atributo_Calc_1*:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1+:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1/:NEW.BDO_Atributo_Calc_2,
       :NEW.BDO_Atributo_Calc_1-:NEW.BDO_Atributo_Calc_2, Data_Dia);
    END IF;
 END.
END.
```

Figura 4. Especificação Genérica do Gatilho.

Para o armazenamento dos “elementos de tempo”, ou seja, dos períodos exatos em que o dado ficou válido no Banco de Dados Operacional, os atributos “BDAT_Atributo_Data_1” e “BDAT_Atributo_Data_2”, especificados no gatilho genérico devem ser utilizados, no sentido de controlar as épocas e/ou períodos em que ocorreram as mudanças nos valores dos dados operacionais. Vale destacar que durante as operações de inclusões (insert) dos dados no BDAT, dos atributos que controlarão os períodos, somente o atributo “BDAT_Atributo_Data_1” será gerado e gravado. O atributo “BDAT_Atributo_Data_2” somente será gerado e gravado quando ocorrerem operações de alteração (*update*) , no sentido de finalizar a validade do dado anteriormente incluído.

Para o armazenamento de valores pré-calculados, isto é, “atributos derivados”, por intermédio das operações de adição, multiplicação, subtração, divisão, e/ou combinações dessas operações matemáticas, provindos do Banco de Dados Operacional para o BDAT, devem ser utilizados os atributos “BDAT_Atributo_Pre_1”, “BDAT_Atributo_Pre_2”, “BDAT_Atributo_Pre_3” e “BDAT_Atributo_Pre_4”. Ver, na Figura 4, a especificação genérica do gatilho proposto como *template*.

2.2.2. Especificação Genérica do Procedimento

A finalidade dessa especificação é oferecer um *template* genérico de um procedimento armazenado que possibilitará a definição de “níveis de granularidade”, no sentido de totalizar dados operacionais e enviá-los ao BDAT, bem como gerar níveis maiores de agrupamento diretamente no BDAT, no sentido de diminuir o volume de dados com o passar do tempo. O procedimento genérico proposto para este passo está especificado na Figura 5.

```
CREATE OR REPLACE PROCEDURE PROCEDURE_TEMPLATE
    (Variavel_Filtro_1 IN BDAT_Tabela.BDAT_Atributo_1%TYPE,
    Variavel_Filtro_2 IN BDAT_Tabela.BDAT_Atributo_2%TYPE, . . . . )
 IS
   Variavel_1 BDO_Tabela.BDO_Atributo_1%TYPE;
   Variavel_2 BDO_Tabela.BDO_Atributo_2%TYPE;  . . . .;
   CURSOR BDO_Tabela_CURSOR IS
     SELECT BDO_Chave_1, BDO_Chave_2, . . . . , BDO_Atributo_1, BDO_Atributo_2, . . . . ,
        Sum(BDO_Atributo_Calc_1* BDO_Atributo_Calc_2), Sum(BDO_Atributo_Calc_3, . . . .)
        INTO Variável_1,Variável_2, . . . .  FROM BDO_Tabela
        WHERE BDO_Chave_1=Variavel_Filtro_1
        OR/AND . . . .  OR BDO_Atributo_1=Variavel_Filtro_2 OR/AND . . . .
        GROUP BY BDO_Chave_1, BDO_Chave_2, . . . . , BDO_Atributo_1, BDO_Atributo_2, . . .
BEGIN
   OPEN BDO_Tabela_CURSOR;
   LOOP
     FETCH BDO_Tabela_CURSOR INTO Variavel_1, Variavel_2, . . . . ;
     EXIT WHEN BDO_Tabela_CURSOR%NOTFOUND;
     DELETE FROM BDAT_Tabela  WHERE BDAT_Chave_1=Variavel_Filtro_1 OR/AND . . . .
         OR BDAT_Atributo_1=Variavel_Filtro_2  OR/AND . . . .
     INSERT INTO BDAT_Tabela (BDAT_Chave_1, BDAT_Chave_2, BDAT_Atributo_1,
         BDAT_Atributo_2, BDAT_Atributo_3, . . . . ) VALUES(Variavel_1,Variavel_2, . . . . );
   END LOOP;
   COMMIT;
 END.
```

Figura 5. Especificação Genérica do Procedimento Armazenado.

Em suma, neste item foram especificados alguns subprogramas a fim de oferecer um padrão genérico para a realização prática da proposta.

3. Estudo de Caso

O Banco de Dados Operacional utilizado foi o do Sistema de Folha de Pagamento, do setor de Recursos Humanos da Fundação Educacional do Município de Assis, já que esse setor encontra muitas dificuldades quando lhe são solicitadas, pela direção executiva, históricos para tomada de decisões e/ou auditorias, bem como uma simples recuperação de dados anteriormente modificados. O nome de serviço do banco de dados da aplicação da Folha de Pagamento é “BDO_FOLHA”.

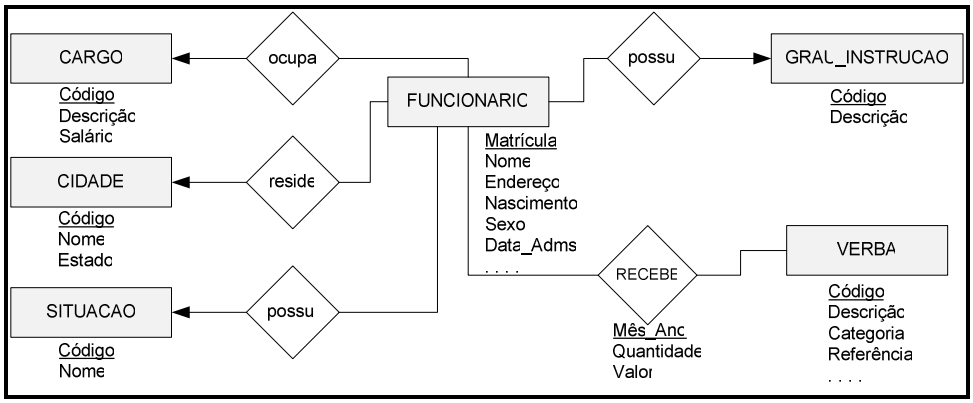

Figura 6. Parte do Modelo de Dados do Sistema de Folha de Pagamento (BDO_FOLHA)

3.2. Resultado da Aplicação da Etapa A

Após a aplicação dos passos da Etapa A, pôde-se chegar ao modelo de dados necessário para que se possa ter um controle razoável das modificações efetuadas junto ao Banco e Dados Operacional "BDO_FOLHA", conforme apresentado na Figura 7.

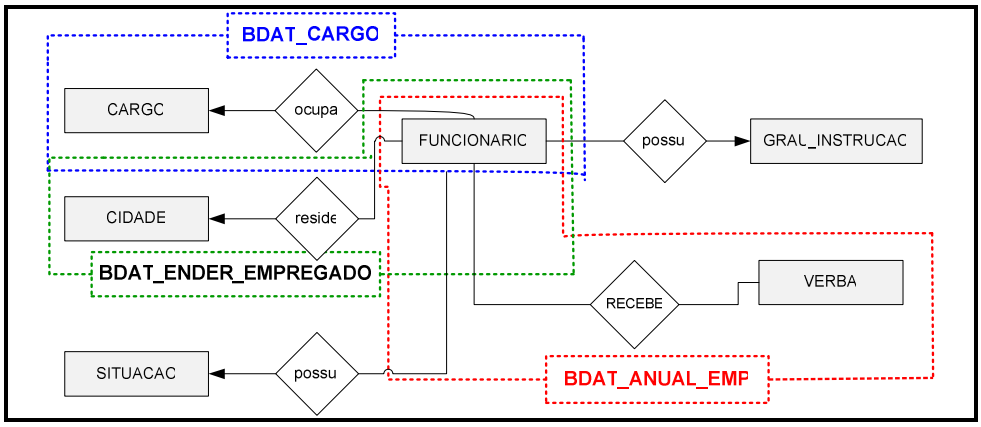

Figura 7. Modelo de Dados Analítico-Temporal

Os conjuntos de entidades (tabelas) BDAT_CARGO, BDAT_ANUAL_EMP e BDAT_ENDER_EMPREGADO são resultantes da modelagem, da aplicação da Etapa A. Essas são as tabelas que formarão o Banco de Dados Analítico-Temporal.

3.3. Aplicação da Etapa B

Neste item, são apresentados dois exemplos, que foram testados e validados no Sistema Gerenciador de Banco de Dados Oracle 11g [Price 2009]. Os construtores utilizados para ilustrar os modelos de dados apresentados na figuras foram retirados de [Silberschatz, Korth e Sudarschan 2006].

Exemplo 01: Aplicando o gatilho genérico.

A Figura 8 apresenta parte do Modelo de Dados Operacional, representado pelos conjuntos de entidades, FUNCIONARIO e CARGO, e parte do Modelo de Dados Analítico-Temporal, representado pelo conjunto de entidades BDTA_CARGO, obtido quando da aplicação da Etapa A.

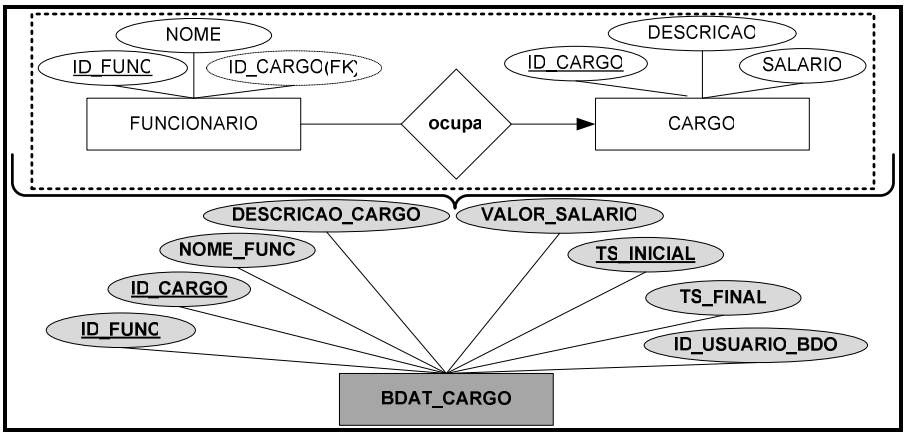

Figura 8. Conjuntos de Entidades utilizados para a criação de um Gatilho.

```
CREATE OR REPLACE TRIGGER GAT_BDAT_CARGO
BEFORE INSERT OR UPDATE OF ID_CARGO ON FUNCIONARIO
FOR EACH ROW
DECLARE
  V_DESCRICAO CARGO.DESCRICAO%TYPE;
  V_SALARIO CARGO.SALARIO%TYPE;
  V_USER_NAME USER_USERS.USERNAME%TYPE;
BEGIN
  SELECT DESCRICAO, SALARIO INTO V_DESCRICAO, V_SALARIO
    FROM CARGO WHERE ID_CARGO=:NEW.ID_CARGO;
  SELECT USERNAME INTO V_USER_NAME
    FROM USER_USERS;
  IF INSERTING THEN
     INSERT INTO BDAT_CARGO@BDAT_LINK(ID_FUNC, NOME_FUNC, ID_CARGO,
       DESCRICAO_CARGO, VAL_SALARIO, TS_INICIAL, ID_USUARIO_BDO)
       VALUES(:NEW.ID_FUNC, :NEW.NOME, :NEW.ID_CARGO, V_DESCRICAO,
       V_SALARIO, SYSDATE, V_USER_NAME);
  ELSE
     UPDATE BDAT_CARGO@BDAT_LINK SET TS_FINAL=SYSDATE
       WHERE BDAT_CARGO.ID_CARGO=:OLD.ID_CARGO AND
       BDAT_CARGO.ID_FUNC=:OLD.ID_FUNC AND BDAT_CARGO.TS_FINAL IS NULL;
     INSERT INTO BDAT_CARGO@BDAT_LINK (ID_FUNC, NOME_FUNC, ID_CARGO,
       DESCRICAO_CARGO, VAL_SALARIO, TS_INICIAL,  ID_USUARIO_BDO,)
       VALUES(:OLD.ID_FUNC, :OLD.NOME, :NEW.ID_CARGO, V_DESCRICAO,
       V_SALARIO,  SYSDATE, V_USER_NAME);
  END IF;
END;
```

OBS: **As siglas :NEW e :OLD possibilitam obter os valores posteriores/anteriores de um atributo.**

Figura 9. Ilustração Prática do Gatilho Genérico (GAT_BDAT_CARGO)

Tendo como base esses conjuntos de entidades, foi elaborado o gatilho GAT_BDAT_CARGO, com base na especificação genérica da Figura 4, ilustrado na Figura 9, cujo principal objetivo é armazenar informações relacionadas aos cargos que um funcionário ocupou durante toda a sua vida na empresa, considerando-se também todos os períodos nos quais ocorreram essas mudanças. Os atributos TS_INICIAL e TS_FINAL destacam a adição dos elementos "tempo" responsáveis pelo armazenamento dos períodos de validade dos dados. O atributo ID_USUARIO_BDO tem por função, armazenar o usuário do banco de dados operacional que realizou a operação, que poderá ser usado para eventuais auditorias. Os atributos NOME_FUNC, DESCRICAO_CARGO e VALOR_SALARIO, têm por função, retratar os valores exatos que um atributo assumiu ao longo do tempo, já que esses atributos podem sofrer alterações em seus valores operacionais, como por exemplo, a mudança da descrição de um cargo, um empregado que a partir de determinada época sofreu alteração em seu nome pelo fato de ter se casado, e principalmente, as mudanças dos valores salariais que

um funcionário venha a sofrer. O gatilho ficará armazenado no banco de dados operacional.

Exemplo 02: Aplicando o procedimento genérico.

A Figura 10 apresenta parte do Modelo de Dados Operacional, representado pelos conjuntos de entidades FUNCIONARIO e VERBA, bem como pelo conjunto de relacionamentos RECEBE, e parte do Modelo de Dados Analítico-Temporal, representado pelo conjunto de entidades BDAT_ANUAL_FUNC, ao qual demonstra a aplicação da granularidade, já que os recebimentos são agrupados e totalizados por funcionário, verba e ano. O atributo TOTAL se refere a soma do atributo VALOR, e o atributo QTDE a soma do atributo QUANTIDADE.

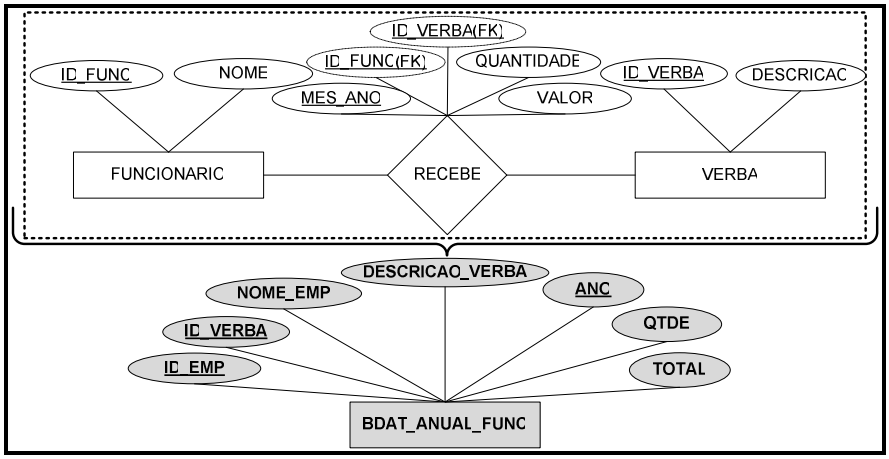

Figura 10. Conjuntos de Entidades utilizados para a criação de um Procedimento.

```
CREATE OR REPLACE PROCEDURE PROC_BDAT_ANUAL_FUNC
  (I_ANO IN CHAR)
IS
  V_ID_FUNC RECEBE.MATRICULA%TYPE;
  V_ID_VERBA RECEBE.VERBA%TYPE;
  V_NOME_FUNC FUNCIONARIO.NOME%TYPE;
  V_DESCRICAO_VERBA VERBA.DESCRICAO%TYPE;
  V_ANO CHAR(4);
  V_VALOR RECEBE.VALOR%TYPE;
  V_QTDE RECEBE.QUANTIDADE%TYPE;
  CURSOR BDO_ANUAL_FUNC_CURSOR IS
  SELECT FUNCIONARIO.MATRICULA,FUNCIONARIO.NOME,VERBA.CODIGO,
     VERBA.DESCRICAO,SUBSTR(TO_CHAR(MES_ANO),3,4),SUM(VALOR),
     SUM(QUANTIDADE) INTO V_ID_FUNC,V_NOME_FUNC,V_ID_VERBA,
     V_DESCRICAO_VERBA, V_ANO,V_VALOR,V_QTDE
     FROM FUNCIONARIO,VERBA,RECEBE
     WHERE FUNCIONARIO.MATRICULA=RECEBE.MATRICULA AND
     VERBA.CODIGO=RECEBE.VERBA  AND
     SUBSTR(TO_CHAR(MES_ANO),3,4)=I_ANO
     GROUP BY FUNCIONARIO.MATRICULA,FUNCIONARIO.NOME,VERBA.CODIGO,
     VERBA.DESCRICAO, SUBSTR(TO_CHAR(MES_ANO),3,4);
BEGIN
  DELETE FROM BDAT_ANUAL_FUNC@BDAT_LINK WHERE ANO=I_ANO;
  OPEN BDO_ANUAL_FUNC_CURSOR;
  LOOP
    FETCH BDO_ANUAL_FUNC_CURSOR INTO V_ID_FUNC,V_NOME_FUNC,
        V_ID_VERBA,V_DESCRICAO_VERBA, V_ANO,V_VALOR,V_QTDE;
    EXIT WHEN BDO_ANUAL_FUNC_CURSOR%NOTFOUND;
    INSERT INTO BDAT_ANUAL_FUNC@BDAT_LINK
       VALUES(V_ID_FUNC,V_ID_VERBA,V_NOME_FUNC,V_DESCRICAO_VERBA,
       V_ANO,V_VALOR,V_QTDE);
  END LOOP;
  COMMIT;
END;
/
```

Figura 11. Especificação Prática do Procedimento Genérico

Tendo como base esses conjuntos de entidades e relacionamentos, foi elaborado o procedimento armazenado ilustrado na Figura 11. Assim, o procedimento armazenado PROC_BDAT_ANUAL_FUNC é um exemplo prático de como armazenar dados sumarizados, aplicando-se níveis de granularidade.

Os exemplos apresentados foram elaborados com base em um banco de dados que armazena informações sobre uma Folha de Pagamento, porém, esse método pode ser aplicado em outras áreas, tais como, telefonia, bancária, acadêmica, dentre outras.

4. Trabalhos Correlatos

Há os trabalhos que se intitulam “Sistemas de Bancos de Dados Temporais”, com características e funcionalidades para o tratamento da temporalidade, que focalizam mais a questão do controle do tempo das transações, validade dos dados, versionamento de esquemas, conversões entre modelos entidade-relacionamento e multidimensionais. Não se tem, nesses trabalhos, a preocupação de oferecer um método para a modelagem um Banco de Dados Analítico-Temporal.

Um trabalho de proposta similar ao aqui apresentado é o intitulado “Implementação de Bancos de Dados Temporais com Versionamento: um estudo comparativo”, de [Santos 2003], cujo principal objetivo era avaliar as diferentes técnicas de implementação e gerenciamento de versionamento de esquemas em bancos de dados temporais. Outros trabalhos de proposta similar são: “Conceptual Design of Data Warehouses from E/R Schemes” de [Moody e Kortnik 2000] e o “From enterprise models to dimensional models: a methodology for data warehouse and data mart design” de [Golfarelli et al. 1998], têm por finalidade encontrar formas de se converter um modelo entidade-relacionamento em um modelo multidimensional, bem como efetuar a consolidação e a agregação dos dados.

5. Considerações Finais

Este trabalho possibilita a modelagem e a implementação de um Banco de Dados Analítico-Temporal, de grande valia para o processo de geração de históricos de dados, sem a necessidade de se adquirir onerosas ferramentas de *software*, já que as rotinas necessárias são implementadas por meio de gatilhos e procedimentos armazenados, recursos oferecidos pelos próprios Sistemas Gerenciadores de Bancos de Dados. Acredita-se que essa é uma solução simples e de menor custo, que poderá vir a suprir as necessidades quanto à geração e ao armazenamento de dados analítico-temporais, para apoiar tomadas de decisões, auditorias e recuperação de dados.

Vale reforçar, que a estrutura de dados dos Bancos de Dados Operacionais em funcionamento, não é afetada, sendo necessária apenas a inserção de gatilhos e procedimentos armazenados. Nessa mesma linha, não são necessárias alterações nos programas já implementados e utilizados pelos usuários finais.

Um resultado bastante interessante está ligado aos períodos de atualizações, visto que a maioria das atualizações deve ser propagada ao BDAT, logo após uma ação de modificação ter sido submetida ao Banco de Dados Operacional, obtendo-se assim, um sincronismo quase que imediato, entre o BDAT e os Bancos de Dados Operacionais.

No que se refere à sobrecarga de processamento no ambiente operacional, considerando-se que a quantidade de registros a serem inseridos no BDAT não será

grande, a cada disparo dos gatilhos inseridos no ambiente operacional, entende-se que a sobrecarga imposta não prejudique sensivelmente o seu desempenho. Além do mais, os procedimentos armazenados, serão acionados em períodos agendados.

Referências

BRYLA, Bob; LONEY, Kevin. “Oracle Database 11g Manual do DBA”, 2008.

CORDEIRO, Robson Leonardo Ferreira; SANTOS, Clesio Saraiva dos; EDELWEISS, Nina; GALANTE, Renata de Matos. “Classificação de restrições de integridade em bancos de dados temporais de versões”. In: Brazilian Symposium on Databases, 19. Anais/Proceedings SBBD’2004. Brasilia, p. 336-337, 2004.

GOLFARELLI, Mateo; MAIO, Dario; RIZZI, Stefano. “Conceptual Design of Data Warehouses from E/R Schemes”. In: Hawaii International Hierarquias Conference on Systems Sciences, 1998, Hawaii. Proceedings. Hawaii, 1998. 10 p.

ITALIANO, Isabel C.; FERREIRA, João Eduardo. “Synchronization Options for Data Warehouse Designs”, Publicado na IEEE Computer Magazine, Revista de IEEE Computer Society, 2006.

MOODY, Daniel L.; KORTINK, Mark A. R. “From enterprise models to dimensional models: a methodology for data warehouse and data mart design”. In: International Workshop on Design and Management of Data Warehouse, Stockholm, 28., p.2, 2000, p. 1-12.

POLETTO, Alex S. R. de S.; ALMEIDA JUNIOR, Jorge Rady de. “Modeling of an Analytical Database System”, 9ª International Conference on Enterprise Information Systems - ICEIS’2007, Ilha da Madeira, Portugal, Funchal, 13 – 16 de Jun., 2007.

PRICE, Jason. “Oracle Database 11g SQL: Domine SQL e PL/SQL no banco de dados Oracle”. Aborda as versões 11g, 10g, 9i e 8i. Porto Alegre. Editora Bookman, 2009.

SANTOS, Constância da Silva. “Implementação de bancos de dados temporais com versionamento de esquemas: um estudo comparativo”. 2003. 68 p. Dissertação - Instituto de Informática, Univ. Fed. Rio Grande Sul, Porto Alegre, 2003.

SILBERSCHATZ, Abraham; KORTH, Henry F.; SUDARSCHAN, S. “Sistemas de Bancos de Dados”. 5ª edição - Rio de Janeiro. Editora: Elsevier, 2006.

SPRAGUE, Ralph H. Jr.; WATSON, Hugh J. “Sistema de apoio à decisão: colocando a teoria em prática”, 2ª edição – Rio de Janeiro. Editora: Campus, 1991.

TANSEL, Abdullah Uz. “Temporal relational data model”. Revista IEEE Computer Society (IEEE Transactions on Knowledge e Data Engineering), v.9, n.3, may/june, p. 464-479, 1997.

VASSILIADIS, Panos; SIMITSIS, Alkis; SKIADOPOULOS, Spiros. “Conceptual modeling for ETL processes”. In: Workshop on Data Warehousing and OLAP. Proceedings of the 5th ACM international workshop on Data Warehousing and OLAP. McLean, Virginia, USA, p. 14-21, 2002.