PTV – PACOTE DE TEMPO DE VALIDADE PARA O SGBD ORACLE

Sandro Favin Pinheiro[1,2]
sandro@fetagrs.com.br

Nina Edelweiss[2]
nina@inf.ufrgs.br

[1]Av. João XXIII, 113 Apto. 234
Cep. 91060-100 – Porto Alegre/RS

[2]UFRGS - Universidade Federal do Rio Grande do Sul

RESUMO

Embora há muito tempo já se tenha conhecimento da importância da recuperação de informações temporais, não existe um SGBD temporal que tenha sido desenvolvido unicamente para aplicações comerciais, que supra todas as necessidades e abranja todos os aspectos temporais necessários a estas aplicações. O SGBD Oracle, a partir da versão 8i, possibilita a inserção de características temporais similares às de tempo de transação no BD, através de um pacote de tempo denominado *Time Series Cartridge*. Entretanto, em muitos casos, este não é suficiente para que se possa implementar por completo tudo o que foi especificado na modelagem do sistema. A modelagem completa da realidade só é alcançada se forem utilizadas em conjunto as características de tempo de transação e de tempo de validade no banco de dados. Neste artigo são sugeridos e implementados mecanismos para a inserção e o gerenciamento do tempo de validade no SGBD Oracle, através da implementação de um pacote denominado PTV. No PTV, o tempo de validade é administrado através da execução de funções, procedimentos, gatilhos e objetos que possibilitam a inserção de informações temporais, bem como, mantêm a consistência temporal do banco de dados.
Palavras-chave: Banco de Dados, Banco de Dados Temporal, Séries de Tempo.

ABSTRACT

Although the importance of the temporal information retrieval is known from a long time, a temporal SGBD for commercial applications, supporting all the necessities and that embodies all the temporal aspects essential to these aplications has not get been developed. The Oracle DBMS introduced in his version 8 a temporal package – the T*ime Series Cartridge*. The use of this cartridge allows the insertion of temporal properties similar to the temporal databases transaction time. However, in many practical applications this temporal information is not enough to implement all the temporal features identified in the system's modeling. Actions to insert and management the validity time using the Oracle DBMS are proposed and implemented in this work. Validity time is managed through functions, procedures, triggers and objetcs execution. These processes were organized in a package, *Valid Time Package*, whitch can be used together with the Time Series Package.

1 Introdução

Informações temporais estão presentes em um grande número de sistemas que modelam o mundo real. No entanto, em muitos casos, no processo de construção de sistemas, as informações temporais são especificadas indutivamente e modeladas sem visar uma dimensão temporal, adequando-se aos recursos do banco de dados.

A versão 8i do Sistema Gerenciados de Banco de Dados (SGBD) Oracle oferece alguns recursos para tratamento de informações temporais, através da instalação de um cartucho de tempo, chamado *Time Series Cartridge.* O *Time Series Cartridge* (TSC) é um cartucho composto de um conjunto de objetos, procedimentos e funções, que torna possível situar um dado ou objeto no tempo (ORACLE, 2001).

Apesar do Banco de Dados (BD) Oracle fornecer alguns mecanismos para tratamento de informações temporais, estes recursos são muito superficiais e ficam muito aquém da necessidade dos sistemas que modelam o mundo real.

O objetivo deste artigo é buscar mecanismos para que aplicações possam ser implementadas em SGBDs comerciais mantendo ao máximo as características temporais especificadas no projeto do sistema. Para alcançar este objetivo, no artigo são sugeridos e implementados recursos para a inserção, a manutenção e o gerenciamento do tempo de validade em BD objeto-relacionais.

2 Banco de Dados Temporais

Diferentemente de BDs convencionais onde a realidade é representada apenas pelo estado presente de um objeto, os bancos de dados temporais permitem armazenar e recuperar todos os estados do objeto ao longo do tempo (ÖZSOYOGLU e SNODGRASS, 1995)(EDELWEISS e OLIVEIRA, 1994).

O processo de modelagem de dados na construção de sistemas nada mais é do que o mapeamento de informações do mundo real para um ambiente computacional. Normalmente em

aplicações do mundo real estão presentes informações temporais relevantes à especificação dos sistemas, informações estas que tiveram, têm ou terão seus valores alterados.

Informações temporais são representadas em BDs através de datas, períodos, intervalos temporais e duração da validade de informações. A associação entre as informações temporais e os dados, ou simplesmente a associação de tempo a um dado, é possível através da inserção de uma dimensão temporal no banco de dados. Com esta dimensão é possível identificar quando uma informação foi definida e qual o tempo em que ela é válida (EDELWEISS, 1998). Esta dimensão temporal associa um tempo a um atributo de forma que, se o valor do atributo for alterado, o valor anterior não é perdido.

3 Inserção e Gerência do Tempo em SGBDs Relacionais

Um sistema gerenciador de banco de dados temporais (SGBDT) é responsável por manter a consistência do Banco de Dados Temporal (BDT), fornecer mecanismos de armazenamento e recursos para operações de inserção, exclusão e atualização de forma rápida e segura. Estas operações sobre dados temporais demandam muito tempo de processamento, o que pode comprometer o desempenho e a confiabilidade dos BDTs. Este processo pode piorar se os recursos forem administrados através de gatilhos, funções e procedimentos.

Em (HÜBLER, 2000) são analisados procedimentos para inserir informações temporais sobre BD convencionais, utilizando os conceitos aplicados a BDTs. Não é medido o tempo de processamento para garantir a consistência e o tempo de resposta na execução de operações sobre os dados. Presume-se que a utilização dos conceitos de BDTs como um todo, sobre BD convencionais, utilizando recursos como gatilhos e funções, degradariam sensivelmente o desempenho de acesso às informações do BD.

4 Pacote de Tempo de Validade (PTV)

O PTV é um pacote para o SGBD Oracle que fornecem recursos para a inserção e a administração do tempo de validade, bem como recursos para facilitar a recuperação de informações temporais. Através do PTV, o tempo de validade é administrado através de um intervalo temporal, identificando a validade inicial e final da informação. Desta forma, foram

definidos dois atributos para cada informação, o atributo *TvalidadeI* que representa o tempo de validade inicial e o *TvalidadeF* representando o tempo de validade final.

Após a execução do PTV, as tabelas temporais passam a conter uma chave composta, aplicada sobre o atributo de identificação (*ID*) e o atributo de tempo de validade inicial (*TvalidadeI*). Desta forma é evitado que haja informações redundantes ou até mesmo inconsistentes.

O objetivo principal do pacote é controlar a consistência temporal do BDT. O objetivo é alcançado através da criação de objetos, gatilhos, procedimentos e funções que são executados durante operações de inserção, alteração e remoção de informações temporais.

Através da execução de procedimentos do PTV, é possível inserir o rótulo temporal de tempo de validade, bem como acionar o mecanismo de controle de consistência. Este mecanismo é ativado através da execução de gatilhos. Para cada tabela temporal no BDT existira um gatilho associado. Os gatilhos implementam regras de BDT, ou seja, formas para manter o BD consistente da visão temporal.

Basicamente o PTV é formado de:

Procedimentos:

insereAtributoTV – insere o rótulo do tempo de validade;

ativaTV – ativa o mecanismo de controle de consistência (cria o gatilho);

de*sativaTV* – desativa o mecanismo de controle de consistência (exclui o gatilho);

verConsistenciaTV – verifica se um BD já populado esta temporalmente consistente;

Gatilhos:

TV+(nome da tabela) – controla a consistência do BD. Existe um gatilho para cada tabela temporal no BDT. São implementadas as regras de consistência de tempo de validade;

Função:

periodo() – auxilia na elaboração de consulta para acesso a informações temporais;

5 Regras de Consistência de Dados de Tempo de Validade

Esta seção sugere mecanismos de inserção e administração de informações temporais, levando em consideração a performance do acesso e da manipulação dos dados no BD. As características de BDTs são preservadas. Entretanto, algumas funcionalidades não serão permitidas por imposição das regras de consistência, integridade e gerenciamento implementadas, bem como das limitações impostas pelo SGBD Oracle.

A seguir são apresentadas as regras implementadas para gerenciar informações de tempo de validade, onde:

tstamp	- representa o rótulo temporal;
tvi	- representa o tempo de validade inicial;
tvf	- representa o tempo de validade final;
new	- representa o novo valor de tempo de validade da tupla;
old	- representa o valor já inserido do tempo de validade da tupla;
maxOld	- representa o maior valor de tempo de validade existente no BD;
minOld	- representa o menor valor de tempo de validade existente no BD;
tvfOldNext	- representa o tempo de validade final da próxima tupla ;
tviOldPrior	- representa o tempo de validade inicial da tupla anterior;
UT	- Unidade de tempo com granularidade de 1 dia;
123	- (fonte vazada) representa os dados inconsistentes no BD;
123	- (fonte preta) representa os valores que sofreram alterações.

Regra 1 : **tvi <> NULL**

- O tempo de validade inicial de uma informação não pode ser nulo. Se este valor não for fornecido pelo usuário será assumido o valor do rótulo temporal do tempo de transação, se possível.

Regra 2 : **tvi < tvf**

- O tempo de validade inicial de uma informação nunca poderá ser maior que o tempo de validade final.

Regra 3 : **tvi.NEW > tvi.MaxOLD**

➢ Esta regra aplica-se quando o valor de *tvi* da tupla a ser inserida no BDT (nova tupla) é maior que o maior valor de *tvi* de uma tupla que já tenha sido inserida no BDT (tupla armazenada).

Regra 3.1: SE (**tvi.NEW > tvi.MaxOLD**) E ((tvi.new <= tvf.old) OU (tvi.new = NULL))

ENTÃO tvf.old := tvi.new – UT

➢ Se o valor de *tvi* da nova tupla for menor ou igual do que o valor de *tvf* da tupla armazenada, ou for igual a nulo, uma inconsistência no BDT é verificada, como mostra a tabela 1, onde o BDT está inconsistente. Nesta ocasião uma operação de consulta requisitando o salário da funcionária Maria na data 07/10/2001, resultaria dois valores.

Id	tvi	tvf	nome	Salário	
1011	10/10/2000	10/10/2001	Maria	800,00	
1011	05/10/2001	10/10/2002	Maria	1000,00	Nova

Tabela 1: Regra 3.1 *tvi.new* < *tvf.old* (BDT inconsistente)

A solução encontrada para manter a consistência no BDT foi alterar automaticamente o valor do *tvf* da tupla armazenada pelo valor de *tvi* da nova tupla, diminuído de uma unidade de tempo. A tabela 2 mostra um BDT consistente a partir da aplicação da regra 3.1 sobre o exemplo da tabela 1.

Id	Tvi	tvf	nome	Salário	
1011	10/10/2000	04/10/2001	Maria	800,00	
1011	05/10/2001	10/10/2002	Maria	1000,00	Nova

Tabela 2: Regra 3.1 *tvi.new* < *tvf.old* (BDT consistente)

REGRA 4 : **tvi.NEW < tvi.MinOLD**

➢ Esta regra aplica-se quando o valor de *tvi* da tupla a ser inserida no BDT é menor que o menor valor de *tvi* de uma tupla que já tenha sido inserida no BDT.

Regra 4.1 : SE (**TVI.NEW < TVI.MinOLD**) E (tvf.new >= tvi.old) E ((tvf.old + UT) > tvi.old)

ENTÃO tvi.old := tvf.new + UT

SENÃO exceção

➢ Se o valor de *tvf* da nova tupla for maior ou igual ao valor de *tvi* da tupla armazenada, uma inconsistência é verificada, como mostra a tabela 3. Neste caso, mais de uma tupla teria intervalos de tempo em comum ou a mesma unidade temporal para informações diferentes. A solução para esta situação é substituir o valor de *tvi* da tupla já armazenada pelo valor de *tvf* da nova tupla, acrescentado de uma unidade de tempo. A tabela 4 mostra esta situação.

Id	Tvi	tvf	Nome	salario	
1011	13/01/2001	NULL	Maria	1000,00	
1011	10/01/2001	15/01/2001	Maria	800,00	Nova

Tabela 3: Regra 4.1 *tvf.new* > *tvi.old* (BDT inconsistente).

Id	Tvi	Tvf	Nome	salario	
1011	16/01/2001	NULL	Maria	1000,00	
1011	10/01/2001	15/01/2001	Maria	800,00	Nova

Tabela 4: Regra 4.1 *tvf.new* > *tvi.old* (BDT consistente).

Valor nulo para o *tvf* da nova tupla implica na invalidade do aspecto temporal das demais tuplas armazenadas no BDT. Deste modo, presume-se que a pretensão do usuário é a exclusão de todas as tuplas subseqüentes. Desta maneira não é permitida esta operação, de forma que uma exceção é executada.

REGRA 5 : **tvi.NEW = tvi.OLD (não extremo)**

➢ Esta regra aplica-se quando o valor de *tvi* da tupla a ser inserida no BDT é igual ao valor de *tvi* de qualquer outra tupla que já tenha sido inserida no BDT, desde que este valor não seja o maior ou o menor valor de *tvi* da tupla de mesmo "ID" existente na tabela, ou seja, não esteja localizada nas extremidades da tabela. As extremidades são o maior ou o menor valor de *tvi* armazenado.

Regra 5.1 : SE (**tvi.NEW = tvi.OLD (não extremo)**) E ((tvi.new = tvi.old) E (tvf.new = tvf.old))

OU (tvf.new >= tvf.old) OU (tvf.new < tvf.old) OU (tvf.new = NULL)

ENTÃO exceção

➢ Se o valor de *tvi* da nova tupla já existir para outra tupla no BDT e o valor de *tvf* da nova tupla coincidir com o valor de *tvf* da tupla armazenada, ou o valor de *tvf* da nova tupla for menor, igual ou maior que o valor de *tvf* da tupla armazenada, ou até mesmo nulo, duas soluções são encontradas: a atualização dos dados ; ou a execução de uma exceção proibindo a operação.

Em BDTs não é permitida a atualização física dos dados. Bem como para BDTs, optou-se pela implementação da proibição da operação de atualização. Se a intenção do usuário é a de atualização das informações no BDT por não serem verídicas, por exemplo, a alternativa é realizar a exclusão física da informação (através do DBA) e realizar uma nova inserção. A tabela 5 mostra o BD inconsistente baseado nesta situação.

Id	Tvi	tvf	Nome	salario	
1011	10/01/2001	15/05/2001	Maria	800,00	
1011	16/05/2001	25/12/2001	Maria	1000,00	
1011	26/12/2001	29/06/2002	Maria	1200,00	
1011	16/05/2001	25/12/2001	Maria	1200,00	Nova

Tabela 5: Regra 5.1 (*tvi.new* = *tvi.old*) E (*tvf.new* = *tvf.old*) – (Inconsistente)

REGRA 6 **: tvi.NEW <> TVI.OLD (não extremo)**

➢ Esta regra aplica-se quando o valor de *tvi* de uma nova tupla não existe no BDT, e este valor não é um valor de extremidades, ou seja, o valor de *tvi* não é o maior ou o menor valor de *tvi* existente no BD.

Regra 6.1 : SE (**tvi.NEW <> tvi.OLD (não extremo))** E ((tvf.new >= tvi.oldNext) E (tvf.new +1 <= tvf.oldNext)) ENTÃO tvi.oldNext:= tvf.new +UT
SENÃO exceção

➢ A solução encontrada para a situação onde o valor de *tvf* da nova tupla é maior ou igual ao valor de *tvi* da próxima tupla, é a atualização do valor de *tvi* da tupla subseqüente, utilizando o valor de *tvf* da nova tupla acrescentado de uma unidade temporal. Entretanto, este procedimento somente poderá ser realizado se o valor de *tvf* da nova tupla não exceder o valor de

tvf da tupla já armazenada. Tal procedimento necessitaria da atualização de um número de tuplas indeterminadas. Deste modo, optou-se pela execução de uma exceção. A tabela 6 mostra esta situação para um BDT inconsistente.

Id	Tvi	tvf	Nome	Salário
1011	10/01/2001	15/02/2001	Maria	800,00
1011	25/02/2001	15/03/2001	Maria	1100,00
1011	16/03/2001	15/04/2001	Maria	1150,00
1011	16/02/2001	30/02/2001	Maria	1000,00

Nova

Tabela 6: Regra 6.1 *tvf.new* > *tvi.oldNext* (BDT inconsistente).

Regra 6.2 : SE (**tvi.NEW <> tvi.OLD (não extremo)**) E ((tvi.new <= tvf.oldPrior) E (tvi.new -1 > tvi.oldPrior)) ENTÃO tvf.oldPrior:= tvi.new –UT
SENÃO exceção

- Para a situação onde o valor de *tvi* da nova tupla é menor ou igual ao valor de *tvf* da tupla que a antecede, a solução encontrada foi diminuir de uma unidade temporal o valor de *tvf* da tupla antecessora. Para garantir a consistência é necessário que o valor de *tvf* não seja inferior ao valor de *tvi* da mesma tupla. Se isso ocorrer, a operação é abortada e uma exceção é executada. A tabela 7 mostra esta situação para um BDT inconsistente.

Id	Tvi	tvf	Nome	Salário
1011	10/01/2001	15/02/2001	Maria	800,00
1011	25/02/2001	15/03/2001	Maria	1100,00
1011	16/03/2001	15/04/2001	Maria	1150,00
1011	13/02/2001	30/02/2001	Maria	1000,00

Nova

Tabela 7: Regra 6.2 *tvi.new* <= *tvf.oldPrior* (BDT inconsistente).

REGRA 7 : **tvi.NEW = tvi.MaxOLD**

- Esta regra aplica-se quando o valor de *tvi* da nova tupla coincide com o valor de *tvi* da tupla de maior extremidade, ou seja, quando o valor de *tvi* de uma nova tupla é a igual ao maior valor de *tvi* já inserido no BDT.

Regra 7.1 : SE (**tvi.NEW = tvi.MaxOLD**) E ((tvf.new >= tvf.old) OU (tvf.new < tvf.old) OU

(tvf.new = NULL))
ENTÃO exceção

- Esta situação é a mesma ocorrida para a regra 5.1, onde o valor de *tvf* da nova tupla coincide, é menor ou maior do que o valor de *tvf* da tupla armazenada, ou até mesmo o valor é nulo. Desta maneira as duas soluções encontradas são: atualização dos dados não temporais; ou execução de uma exceção proibindo a operação. Da mesma forma que a regra 5.1, será executada uma exceção proibindo esta operação. Esta situação pode ser verificada na tabela 5.

REGRA 8 : **tvi.NEW = tvi.MinOLD**

- Nesta regra, para todas as circunstâncias envolvidas: (*tvf.new < tvf.old)* ; (*tvf.new > tvf.old) ;(tvf.new = tvf.old) e (tvf.new = NULL)* existem duas soluções: a execução de uma exceção proibindo a operação; ou a atualização dos dados. A descrição dos dados é especificada pelo usuário e varia de aplicação para aplicação, de forma que não há mecanismos pelo qual possam ser identificados. Deste modo não é possível a sua atualização, como foi mencionado na regra 5.1.

6 Implementação

A inserção e a administração do tempo de validade no BD consistem da execução de três procedimentos: a inserção dos atributos de tempo de validade; a ativação do mecanismo de consistência; e a desativação. Todos os procedimentos implementados no PTV são executados sobre o SQL-PLUS do Oracle, podendo ser utilizados em qualquer outro sistema através de chamadas de procedimentos do SQL.

Os atributos necessários para o controle do tempo de validade no BD são inseridos através do procedimento *insereAtributoTV*. Este procedimento requer como parâmetro o nome tabela que conterá os novos atributos.

A ativação do mecanismo de controle do tempo de validade consiste da criação de um gatilho que implementa as regras de consistência apresentadas no capítulo 5. A ativação é executada através do procedimento *ativaTV*. Cada tabela especificada pelo usuário terá um

gatilho a ela associada. Este gatilho é acionado antes de qualquer operação de inserção, remoção e alteração dos dados.

A desativação do mecanismo de controle do tempo de validade consiste na exclusão do gatilho associado à tabela especificada. Este processo é efetivado através da execução do procedimento *desativaTV*. Este procedimento não exclui os atributos de tempo, somente desativa a verificação de consistência de informações temporais durante operações sobre o BD. Os procedimentos para ativação e desativação do PTV podem ser realizados a qualquer instante. Entretando, se o PTV for ativado sobre uma base de dados populada, antes é necessário a utilização do procedimento *verConsistenciaTV* para verificação de inconsistências temporais.

Um recurso foi inserido no PTV com a finalidade de facilitar a recuperação de informações temporais. Este recurso consiste de uma função denominada *PERIODO* que pode ser utilizada juntamente com a expressão *SELECT* do SQL.

A implementação da função *PERIODO* é baseada nas sugestões dos modelos relacionais temporais, mais especificamente na linguagem de consulta temporal estruturada TSQL2 [SNO 95], proposta com o objetivo de consolidar uma linguagem de consulta temporal.

A figura 1 apresenta a utilização da função *PERIODO* para a seguinte a consulta: **"*Selecione o nome dos funcionários que receberam mais de R$500,00 no período de jun-1998 a jun-2000*".**

```
SELECT func.nome
FROM funcionario func
WHERE func.salário > 500 AND
      func.id IN (
      SELECT per.id
      FROM TABLE(CAST(ptv.periodo('folha_pgto',
                        '01/06/1998','01/06/2000') as colecaoTV) per);
```

Figura 1: Consulta utilizando a função PERIODO inserida no PTV

7 Conclusão

A dificuldade de implementação do tempo de validade como um todo em SGBDs relacionais e objeto-relacionais é muito grande. Algumas características de BD de tempo de

validade não são implementadas ou permitidas no pacote de tempo de validade implementado, devido à limitação imposta pelos recursos dos SGBDs relacionais, ou mesmo devido ao tempo de processamento que demandaria executar determinada operação.

Apesar de não ser possível a implementação dos conceitos relativos a BDT como um todo, o objetivo foi alcançado. Através do PTV é possível inserir o tempo de validade e manter a consistência temporal das informações, também foi possível visualizar o limite da implementação de um BD de tempo de validade sobre um SGBD relacional.

8 Bibliografia

EDELWEISS, N.; OLIVEIRA, J. P. M. **Modelagem de aspectos temporais de sistemas de informação**. 1994. Livro texto da Escola de Computação, 9., Universidade Federal de Pernambuco, Recife, 1994.

EDELWEISS, N. **Bancos de Dados Temporais: Teoria e Prática**. In: XVII Jornada de Atualização em Informática, do XVIII Congresso Nacional da Sociedade Brasileira de Computação, v.2. Ed.: H.P. MOURA. Anais... Recife, 1998, p.225-282.

HÜBLER, P.N. **Definição de um Gerenciador para o Modelo de Dados TF-ORM**. Porto Alegre/RS: Universidade Federal do Rio Grande do Sul – CPGCC da UFRGS, 2000. Dissertação de mestrado.

JENSEN, C.S. et al. **The Consensus Glossary of Temporal Database Concepts** - February 1998 Version. Temporal Databases Research and Practice . O. Etzion, S. Jajodia and S. Sripada (eds.) Springer-Verlag. Berlin Heidelberg, 1998. pp. 367-405.

Oracle 8i ***Time Series*** **User's Guide**, Release 8.1.5 A67294-01, Disponível em http://technet.oracle.com/doc/inter.815/a67294/toc.htm. Acessado em 30 de Outubro de 2001.

ÖZSOYOGLU, G,; SNODGRASS, R. T. **Temporal and real-time databases: a survey**. IEEE Transactions on Knowledge and Data Engineering, New York, v.7, n.4, Aug. 1995, p.513.

SNODGRASS, R: **The TSQL2 Temporal Query Language.** Kluwer Publishers, 1995.