

JOEVA CRISTINA DE ANDRADE

BANCOS DE DADOS TEMPORAIS:
UMA COMPARAÇÃO DE DUAS TÉCNICAS DISTINTAS DE
DESENVOLVIMENTO.

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, na disciplina de Projeto Orientado, como parte das exigências do curso Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

Orientadora
Professora Olinda Nogueira Paes Cardoso

LAVRAS
MINAS GERAIS – BRASIL
2002

JOEVA CRISTINA DE ANDRADE

BANCOS DE DADOS TEMPORAIS:
UMA COMPARAÇÃO DE DUAS TÉCNICAS DISTINTAS DE
DESENVOLVIMENTO.

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras, na disciplina de Projeto Orientado, como parte das exigências do curso Ciência da Computação, para a obtenção do título de Bacharel em Ciência da Computação.

APROVADA em 16 de Dezembro de 2002.

Prof. Ana Cristina Roullier
UFLA

Prof. Olinda Nogueira Paes Cardoso
UFLA
(Orientadora)

LAVRAS
MINAS GERAIS - BRASIL

Sumário

1 – Introdução	4
2 – Bancos de Dados Temporais	8
2.1 – Principais Conceitos de Representação Temporal	10
2.1.1 – O Tempo	10
2.1.2 – Dimensão Temporal	11
2.1.3 – Ordem no Tempo	11
2.1.4 – Tempo de Transação e Tempo de Validade	12
2.1.5 – Tempo Absoluto e Tempo Relativo	12
2.1.6 – Variação Temporal	13
2.1.7 – Granularidade Temporal	13
2.1.8 – Elementos Primitivos de Representação Temporal	13
2.2 – Modelagem de Dados	16
2.2.1 – Extensões do Modelo Relacional	17
2.2.2 – Extensões do Modelo Entidade Relacionamento	18
2.2.2.1 – O Modelo TempER	19
2.3 – Consultas Temporais	21
2.3.1 – Problemas no Processamento de Consultas Temporais	22
2.4 – <i>Time Series Cartridge</i>	22
2.4.1 – Termos para <i>Oracle8i Time Series</i>	23
2.4.2 – Modelos no <i>Oracle8i Time Series</i>	24
2.4.2.1 – Calendários	24
2.4.2.2 – Modelos de Série de Tempo	28
3 – As Implementações	30
3.1 – A Aplicação	34
3.2 – As Técnicas	36
3.2.1 – Implementação de um BDT Utilizando SGBD Relacional	36
3.2.2 – Implementação de um BDT Utilizando um SGBD Temporal	44
3.2.2.1 – <i>Time Series</i> e Tempo de Validade	48
4 – Estudo Comparativo	50
4.1 – Tratamento do Tempo de Validade	50
4.2 – Gerenciamento de Informações	52

4.3 – Armazenamento de Dados	52
4.4 – Consistência em Banco de Dados	53
4.5 – Validação de Rótulos Temporais	54
4.6 – Facilidade de Uso	55
4.7 – Orientação à Objeto x Modelo Relacional	55
5 – Conclusões	57
6 – Referência Bibliográfica	59

Lista de Figuras

Figura 2.1 – Criação de um calendário no <i>Time Series</i>	27
Figura 2.2 – Descrição do tipo de dado <i>ORDTCalendar</i> .	28
Figura 3.1 – Arquitetura simplificada de um SGBD tradicional [Tansel 93]	33
Figura 3.2 – Modelagem para BDT utilizando um banco de dados relacional e modelagem ER.	37
Figura 3.3 – Representação da seqüência de disparos após criação da entidade evento	41
Figura 3.4 – Modelagem do Banco de Dados Temporal utilizando modelo TempER	45
Figura 3.5 – Calendário para Promotora de Eventos com granularidade de dia.	48
Figura 3.6 – Calendário para Promotora de Eventos com granularidade de hora.	48

Lista de Tabelas

Tabela 2.1 – Frequências suportadas pelo <i>Oracle Time Series</i> e seus respectivos códigos.	25
Tabela 3.1 – Dicionário de Dados para Modelagem ER.	37
Tabela 3.2 – Dicionário de Dados para Modelagem TempER.	45
Tabela 4.1 – Tipo de tabela para o atributo <i>nome</i> da entidade <i>CLIENTE</i> .	56

Resumo

A necessidade de se armazenar e recuperar os vários estados de uma informação gerou um novo elemento importante dentro dos bancos de dados convencionais: o tempo. Muitas são as aplicações que associam às suas informações elementos temporais (em diversos aspectos) tais como o tempo em que a informação ocorreu no banco de dados, o tempo em que esta informação terá validade para a aplicação, a situação desta informação a um ano atrás, etc. No entanto, a implementação de um sistema envolvendo o conceito de tempo pode ser realizada de várias formas. Neste trabalho estão presentes duas maneiras distintas de desenvolvimento de um sistema como este. A primeira consiste no desenvolvimento de um Banco de Dados Temporal sobre um SGBD relacional cujo tratamento de informações temporais é realizado pela aplicação. A segunda técnica consiste no desenvolvimento de um Banco de Dados Temporal sobre um SGBDT capaz de manipular as informações temporais retirando da aplicação esta função. A partir do desenvolvimento dessas duas técnicas, pretende-se compará-las e relacionar quais as vantagens e desvantagens de cada uma, desde a modelagem do sistema até seu pleno funcionamento. Este trabalho destina-se a pessoas que atuam na área ou que possuem conhecimento básico de Banco de Dados.

Palavras-chave:

Técnicas de desenvolvimento, Bancos de Dados Temporais.

Capítulo 1

Introdução

Os Bancos de Dados relacionais ou convencionais (BD) são utilizados para armazenar informações de uma parte do mundo real representada por uma aplicação. Essas informações são relacionáveis e acessíveis e seu conteúdo reflete um instante da realidade. Portanto, a medida em que o tempo passa, um banco de dados relacional se modifica conforme as mudanças do mundo real, ou seja, suas informações vão se alterando a cada instante de acordo com o novo estado do mundo. Dessa maneira, todas as informações armazenadas anteriormente a uma mudança são perdidas.

Para algumas aplicações, talvez a maior parte delas, tanto o estado passado das informações como o estado presente são de extrema importância. Por exemplo, para um sistema bancário, é necessário que informações sobre todas as transações efetuadas em uma conta sejam armazenadas. Conforme as necessidades do cliente ou do próprio banco, essas informações poderão ser obtidas retratando o estado da conta em qualquer instante de tempo desejado ou

todos os eventos ocorridos durante um determinado período. Esse fato então limita o uso de BD relacionais uma vez que estes não possuem recursos que permitam acessar os estados passados da conta.

Nesse contexto, surge então um novo elemento na área de Banco de Dados: o tempo relacionado às informações. Esse novo elemento deu origem aos Bancos de Dados Temporais (ou BDT) que são bancos de dados que permitem armazenar e recuperar todos os estados de um objeto, registrando sua evolução ao longo do tempo [Edelweiss 94, Etzion 98, Tansel 93]. Informações temporais são associadas implicitamente aos dados, correspondendo ao tempo de validade (tempo que a informação será válida no banco de dados) e/ou ao tempo de transação (tempo que a informação foi inserida no banco de dados). Novos conceitos relativos aos BDT são introduzidos na área: dimensão temporal, tempo de transação, tempo de validade, granularidade temporal, etc. Todos esses conceitos serão tratados posteriormente.

Podemos citar mais algumas aplicações cujas características temporais são claras:

- para um médico, os dados passados de um paciente são essenciais para conhecê-lo, fazer novos diagnósticos, acompanhar a evolução de uma doença que o paciente possui, determinar novos tratamentos, etc;
- na área comercial, dados estatísticos de venda de um produto específico podem ser importantes para determinar a frequência de compra deste produto, assim como ajudar na tomada de decisão em vários aspectos;
- controle de tráfego aéreo onde se controla os horários de decolagem e pouso de aviões;
- meio acadêmico onde informações curriculares sobre os alunos precisam ser armazenadas e acessadas posteriormente.

A utilização de BDT tornou-se freqüente em consequência da necessidade de grande parte das aplicações atuais. Portanto, os estudos na área se intensificaram com o intuito de melhor definir conceitos e estratégias de tratamento de informações históricas [Edelweiss 98]. Os resultados destes estudos podem ser encontrados em diversas bibliografias dentre as quais podemos citar [Bolour 82, McKenzie 86, Stam 88, Soo 91, Kline 93, Tsotras 96, Wu 97].

Apesar de todos os estudos e dos novos conceitos para tratamento de dados temporais, BDT são geralmente implementados utilizando-se SGBD (Sistema Gerenciador de Banco de Dados) tradicionais. Isto se deve ao fato de que SGBD que dominam o mercado atualmente (como *Oracle*, *Sybase* e *Ingres*), pouco oferecem em termos de funcionalidades para o tratamento adequado de informações com características temporais. Mas isto não significa que os sistemas de informação deixem de tratá-las. O que tem ocorrido é que, sendo necessário, a modelagem de uma aplicação representa essas características de maneira empírica já no nível conceitual.

Este trabalho tem como ponto de partida a questão mencionada acima. Ele consiste da implementação de um BDT para uma aplicação específica, utilizando se para isto duas técnicas distintas. A primeira realiza a implementação de um BD convencional (utilizando o SGBD *Oracle*) e trata as informações temporais de forma explícita, ou seja, o valor temporal relacionado aos dados está estruturado no BD de forma tradicional sendo um atributo comum de uma tabela. Todas as informações temporais necessárias são manipuladas pela aplicação e obtidas pelo usuário da mesma. Na segunda técnica, o valor do tempo é fornecido automaticamente pelo SGBD (sob forma de um rótulo temporal) sendo esta operação transparente ao usuário e à aplicação. Para tal, foi utilizada a ferramenta *Oracle8i Time Series*.

O objetivo deste projeto é comparar estas duas técnicas de implementação de Bancos de Dados Temporais e, com base nos resultados, relatar quais são as vantagens e desvantagens de cada técnica utilizada e, se possível, sugerir qual seria a melhor. Para a comparação foram definidos alguns critérios de avaliação durante e após o desenvolvimento da aplicação. Inicialmente, alguns aspectos foram considerados: dificuldade de implementação da aplicação, desempenho após seu término, consistência e integridade dos dados durante sua utilização e facilidade de uso.

O desenvolvimento deste projeto foi motivado pelo elevado número existente de aplicações que utilizam uma base de dados temporal sendo estes, em sua grande maioria, implementados de maneira tradicional (a primeira técnica utilizada). Isto decorre da não existência de um SGBDT comercial, apesar da substancial atividade de pesquisa e dos requisitos temporais das aplicações.

O trabalho está organizado da seguinte maneira: o Capítulo 2 trata os principais conceitos relacionados à BDT e à representação temporal; o Capítulo 3 descreve o desenvolvimento da implementação das duas técnicas mencionadas: utilizando um BD tradicional e utilizando um BDT; no Capítulo 4 são feitas as comparações entre as implementações e o Capítulo 5 finaliza com as conclusões.

Capítulo 2

Bancos de Dados Temporais

Um banco de dados temporal armazena vários estados dos dados, assim como os instantes em que estes diferentes estados são válidos [Silva 92, Pissiniou 94, Snodgrass 95, Tansel 90].

Segundo [Edelweiss 98], conforme a forma utilizada para armazenar valores temporais, os bancos de dados podem ser classificados em quatro tipos diferentes:

- ***Bancos de Dados Instantâneos***: corresponde aos bancos de dados tradicionais, onde são armazenados os valores presentes. A cada modificação no valor de uma propriedade, o valor anteriormente armazenado é destruído e somente o último valor está disponível. A manutenção de informações temporais neste tipo de banco de dados somente pode ser realizada explicitamente, pela inclusão de atributos

definidos sobre o domínio tempo, e pela sua manipulação dos programas de aplicação;

- ***Bancos de Dados de Tempo de Transação:*** as informações temporais são associadas ao seu tempo de transação sob forma de rótulo temporal (*timestamps*). Este tempo é fornecido automaticamente pelo SGBD, sendo esta operação transparente ao usuário. A alteração do valor de uma informação não destrói o valor anteriormente definido, ficando todos os valores armazenados no banco de dados. O estado atual do BD é composto pelos últimos valores definidos para cada uma das informações.
- ***Bancos de Dados de Tempo de Validade:*** associa a cada informação somente o tempo de sua validade no tempo real. Este pode representar o início de sua validade (ponto no tempo, variação por degraus), a validade somente naquele ponto no tempo (variação discreta), ou seu intervalo de validade. O tempo de validade deve sempre ser fornecido pelo usuário. Em bancos de dados de tempo de validade não se tem acesso ao tempo em que a informação foi definida, sendo armazenado somente o tempo em que a mesma é válida. Este tipo de banco de dados permite que sejam corrigidas informações do passado – se alguma das informações tiver sido registrada incorretamente, é feita uma nova definição com a data de validade correspondente, sendo que somente a versão atual dos dados é a disponível.
- ***Bancos de Dados Bitemporais:*** é a forma mais completa de armazenar informações temporais. Ambos os tempos, de transação e de validade, são associados a cada informação. Toda a história do banco de dados fica

armazenada. É possível ter acesso a todos os estados passados do banco de dados - tanto a história das transações realizadas, como a história da validade dos dados; P estado atual do banco de dados é constituído pelos valores atualmente válidos. Valores futuros podem ser definidos através do tempo de validade, sendo possível recuperar o momento em que estes valores foram definidos para eventuais alterações.

2.1- Principais Conceitos de Representação Temporal

Nesta Seção, serão definidos os principais conceitos relativos à representação de aspectos temporais em bancos de dados para melhor compreensão do leitor. Algumas das definições que se seguem podem ser encontradas em [Jensen 94], num glossário consensual de termos relativos à Bancos de Dados Temporais elaborado pela comunidade desta área através de uma discussão por correio eletrônico e em [Edelweiss 98].

2.1.1 - O Tempo

O tempo, segundo [Tansel 90], é uma seqüência de “pontos de tempo” equidistantes, ou seja, 0, 1, 2, ..., **NOW**. Onde 0 representa o tempo relativo inicial, e **NOW** é o tempo presente.

Entre dois pontos consecutivos de pontos t_i e $t_i + 1$ existe um período de tempo que é equivalente a uma unidade de tempo. Esse período é invisível, a menos que uma unidade menor de tempo seja usada. Um intervalo de tempo é um conjunto consecutivo de “pontos de tempo”, e representa um período de tempo. A representação de um intervalo é através do limite inferior (l), e do limite superior (u). Este intervalo inclui l e todos os pontos entre l e u , mas não inclui o u . O intervalo que inclui o **NOW**, como seu limite superior, é um caso especial, onde ele é fechado em ambos os limites. Os intervalos que incluem

NOW, expandem-se com o passar do tempo, sendo assim diferente dos outros tipos de intervalos que são fixados na duração.

Um conjunto temporal é um conjunto de pontos de tempo $\{t_j, t_k, \dots, t_n\}$. Um conjunto temporal pode ter qualquer número de pontos de tempo que não estejam na forma de um intervalo.

2.1.2 - Dimensão Temporal

Os modelos de dados tradicionais apresentam duas dimensões, as linhas (instâncias dos dados) e as colunas (atributos) de uma tabela. Cada atributo de uma instância apresenta um só valor. Se for feita uma alteração deste valor, o anterior é perdido. Por exemplo, se um atributo representa o saldo da conta bancária de um cliente, o banco de dados somente armazena o último valor.

Os modelos temporais acrescentam mais uma dimensão aos modelos tradicionais – a dimensão temporal. Esta dimensão associa alguma informação temporal a cada valor. Caso o valor de um atributo seja alterado, o valor anterior não é removido do banco de dados – o novo valor é acrescentado, associado a alguma informação que define, por exemplo, seu tempo inicial de validade. Por exemplo, um atributo que representa o saldo da conta bancária de um cliente, todos os valores ficam armazenados, cada valor associado ao seu tempo de validade. Deste modo, é possível acessar toda a história dos atributos, sendo possível analisar sua evolução temporal.

2.1.3 - Ordem no Tempo

A dimensão temporal é composta por uma seqüência de pontos consecutivos no tempo, que recebe o nome de *eixo temporal*. A definição de uma ordem a ser seguida no tempo é fundamental quando utilizada alguma representação temporal. O mais comum é que se assuma que o tempo flui linearmente. Isto implica em total ordenação entre quaisquer dois pontos no tempo. Em alguns

casos pode ser considerado tempo ramificado. Em outros, tempo circular. Não entraremos em detalhes sobre os dois últimos, uma vez que somente o primeiro é o adotado neste trabalho.

2.1.4 - Tempo de Transação e Tempo de Validade

Segundo [Snodgrass, 85], três diferentes conceitos temporais podem ser identificados em aplicações de banco de dados:

- (1) *tempo de transação*: tempo no qual o fato é registrado no banco de dados;
- (2) *tempo de validade*: tempo em que o valor é válido na realidade modelada;
- (3) *tempo definido pelo usuário*: propriedades temporais definidas explicitamente pelos usuários em um domínio temporal e manipuladas pelos programas da aplicação.

O tempo de transação é suprido automaticamente pelo SGBDT, enquanto o tempo de validade é fornecido pelo usuário. O tempo de validade pode ser representado de formas distintas, dependendo do elemento temporal básico utilizado no modelo.

2.1.5 - Tempo Absoluto e Tempo Relativo

Tempo absoluto é uma informação temporal que define um tempo específico, com uma granularidade determinada, associada a um fato. Exemplo: João depositou R\$ 20,00 dia 03/05/2002. Tempo relativo tem sua validade relacionada à validade de outro fato, ou ao momento atual. Exemplo: a poupança de João renderá 4% após um ano da abertura da conta.

2.1.6 - Variação Temporal

Duas maneiras distintas de variação temporal podem ser consideradas: *tempo contínuo e tempo discreto*. Supõe-se que o tempo é contínuo por natureza. Entretanto, sem grande perda de generalidade, o tempo pode ser considerado como discreto. Esta segunda forma de representação simplifica consideravelmente a implementação de modelos de dados.

Modelos de dados que suportam uma noção discreta de variação temporal são baseados em uma linha de tempo composta de uma seqüência de intervalos temporais consecutivos, que não podem ser decompostos, de idêntica duração. Estes intervalos são denominados *chronons*. A duração particular de um *chronon* não é necessariamente fixada no modelo de dados, podendo ser definida em implementações particulares do modelo de dados.

2.1.7 - Granularidade Temporal

A granularidade temporal de um sistema consiste na duração de um *chronon*. Entretanto, dependendo da aplicação, às vezes é necessário considerar simultaneamente diferentes granularidades (minutos, dias, anos) para permitir uma melhor representação da realidade. Por exemplo, em um determinado segmento modelado, a granularidade pode ser diária, enquanto que em outro pode ser semestral. Embora o *chronon* do sistema seja único, é possível manipular estas diferentes granularidades através de funções de operações disponíveis nos SGBD que implementam o modelo.

2.1.8 - Elementos Primitivos de Representação Temporal

Instante de Tempo – o conceito de instante, representando um ponto particular no tempo, depende da forma de variação considerada. Quando é considerado tempo contínuo, um instante é um ponto no tempo de duração infinita. Neste caso os instantes são isomórficos com os números reais, o que significa que

entre dois pontos do tempo sempre existe um outro ponto no tempo. Quando, no entanto, é considerada a variação temporal discreta, um instante é representado por um dos *chronons* da linha de tempo suportada pelo modelo. Na variação discreta, os instantes são isomórficos aos números inteiros ou a um subconjunto destes. Assim, entre dois pontos do tempo consecutivos não existe outro ponto do tempo.

Diz-se que um evento ocorre no tempo t se ocorre em qualquer tempo durante o *chronon* representado por t .

Considerando a ordem de variação temporal linear, temos a existência de um instante especial, correspondendo ao *instante atual (now)*, o qual se move constantemente ao longo do eixo do tempo. Este ponto define o que é considerado como passado (qualquer ponto anterior a este) e o que é futuro (qualquer ponto posterior a ele).

Intervalo Temporal – um intervalo temporal é caracterizado pelo tempo decorrido entre dois instantes – um subconjunto de pontos do eixo temporal. Depende também da forma de representação temporal definida no modelo. Quando é considerado tempo contínuo, o intervalo consiste de infinitos instantes de tempo. Na variação discreta um intervalo é representado por um conjunto finito de *chronons* consecutivos.

É representado pelos dois instantes que o delimitam. Dependendo da pertinência ou não dos instantes limites ao intervalo este pode ser aberto (os limites não pertencem ao intervalo), semi-aberto (um dos limites pertence ao intervalo) ou fechado (os dois limites pertencem ao intervalo). Quando um dos limites é representado pelo instante atual (*now*) temos a representação de um intervalo particular cujo o tamanho varia com a passagem do tempo.

Elemento Temporal – é uma união finita de intervalos de tempo. Em termos de modelagem, o elemento temporal se mostra superior ao uso da primitiva intervalo de tempo pois, quando os intervalos são usados como rótulos temporais, os objetos são fragmentados em várias tuplas, uma para cada instante. Outro aspecto importante desta primitiva é que possibilita a representação de “reencarnação” de objetos com facilidade. Um exemplo da necessidade desse aspecto seria uma pessoa ser cliente de um banco durante o intervalo [1994,1996], tendo fechado a conta e adquirido uma nova três anos depois [1999]. A validade da existência deste cliente no banco seria a união dos intervalos [1994,1996] U [1999, »].

Duração Temporal – durações temporais podem ser basicamente de dois tipos, dependendo do contexto que são definidas: fixas e variáveis. Uma duração fixa independe do contexto de sua definição. Um exemplo típico de duração fixa é uma hora que tem sempre, independentemente do contexto de sua utilização, a duração de 60 minutos. Já a duração variável depende do contexto, sendo um exemplo típico a duração de um mês que pode ser de 28, 29, 30 ou 31 dias.

Representação Temporal Implícita e Explícita – a definição de tempo pode ser feita de forma explícita, através por exemplo, da associação de um valor temporal a uma informação na forma de um rótulo temporal (*timestamp*), ou de forma implícita através da utilização de uma linguagem de lógica temporal. A associação explícita de tempo às informações consiste em associar a cada valor de um atributo, o valor que corresponde à sua primitiva temporal. A representação temporal implícita é feita através da manipulação de conhecimento sobre a ocorrência de eventos ou do relacionamento de intervalos de tempo como, por exemplo: a aula de lógica temporal ocorreu ontem.

2.2 – Modelagem de Dados

A modelagem de uma aplicação temporal é uma forma de representação dos aspectos temporais de grande importância pelos seguintes motivos:

- o sistema pode apresentar informações temporais a serem introduzidas no banco de dados que o representa, sob forma de informação propriamente dita;
- processos a serem executados podem apresentar interações temporais, interações estas que devem também ser representadas;
- determinadas tarefas podem apresentar pré-condições à sua execução, as quais podem ser representadas através de restrições temporais;
- condições de integridade temporal do banco de dados podem ser necessárias.

Deve-se então, buscar um modelo de dados adequado para que todas as requisições acima sejam atendidas.

Um modelo de dados deve apresentar [Edelweiss 98] uma estrutura de objetos que podem ser manipulados por esta linguagem, uma linguagem para atualizar estes objetos (*update*), uma linguagem de consulta, e algum mecanismo para expressar restrições de integridade. Os modelos de dados temporais também devem apresentar estas características, acrescentado a possibilidade de representar informações temporais, efetuar consultas temporais, e permitir a definição de restrições de integridade temporal.

Alguns aspectos devem ser considerados ao analisarmos um modelo de dados, tais como identificar o tipo de rótulo temporal utilizado pelo modelo (ponto no tempo, intervalo temporal, elemento temporal, duração), analisar a forma de variação temporal dos atributos (podem ou não variar com o tempo, todos ou alguns), verificar se os rótulos temporais são implícitos ou explícitos,

homogeneidade temporal e apresentação e funcionalidade da linguagem de consulta.

Ao invés de se criarem novos modelos de dados para tratamento de aspectos temporais, diversos modelos tradicionais foram estendidos para possibilitar a representação de aspectos temporais.

2.2.1 – Extensões do Modelo Relacional

A representação de relacionamentos entre elementos é a base dos modelos relacionais. Ao ser utilizado um modelo temporal, estes relacionamentos devem ser representados ao longo do tempo. Algumas informações adicionais devem ser acrescentadas:

- tempo de início do relacionamento;
- variação do relacionamento com o tempo;
- término do relacionamento;
- reencarnação de relacionamentos;
- restrições de integridade referencial com respeito à dimensão temporal.

Um banco de dados relacional apresenta um conjunto de relações, sendo que cada relação é composta por um conjunto de tuplas. Uma instância deste banco de dados é definida pelo conjunto de relações e de todas suas tuplas.

Quando uma extensão temporal para um banco de dados relacional é feita, três formas podem ser utilizadas para representar a temporização, dependendo do nível ao qual o tempo é associado:

- 1) ao banco de dados como um todo – neste caso, cada estado do banco de dados é armazenado por completo, com seu rótulo temporal. Alterações elementares do BD criam um novo estado;
- 2) às relações – cada relação é temporizada. Para cada estado, todas as tuplas desta relação devem ser armazenadas, com o rótulo temporal correspondente;
- 3) às tuplas – cada tupla é temporizada. Uma alteração elementar de valores de uma relação definem uma nova tupla, e somente esta precisa ser armazenada.

Alguns exemplos de extensão do modelo relacional podem ser citados como os modelos HDRM (*Historical Relational Data Model*) [Clifford 87, 93], IXRM (*Interval-extended Relational Model*) [Lorentzos 93] e TRM (*Temporal Relational Model*) [Navathe 88, 93].

2.2.2 - Extensões do Modelo Entidade Relacionamento

Alguns requisitos foram identificados como necessários a um modelo Entidade-Relacionamento (ER) temporal [Antunes 97]:

- a dimensão temporal deve estar “embutida” no modelo. Desta forma, enquanto que no modelo ER convencional os conjuntos de entidades apresentam apenas duas dimensões, a das tuplas e a dos atributos, no modelo ER temporal passam a apresentar três: a das tuplas, a dos atributos e a do tempo;
- deve oferecer uma notação especial para diferenciar entidades temporizadas (que estão associadas ao tempo) de entidades não temporizadas (que não estão associadas ao tempo);

- deve permitir que uma entidade temporizada se associe com uma entidade não temporizada;
- deve permitir que um relacionamento entre entidades possa ser definido como temporizado ou como não temporizado, não importando qual seja a classificação temporal destas entidades;
- deve permitir que em uma mesma entidade possam conviver atributos temporizados e atributos não temporizados;
- a restrição de cardinalidade que define o grau de participação de uma entidade em um conjunto de relacionamentos temporizados deve considerar os pontos no tempo. Por outro lado, em se tratando de conjunto de relacionamentos não temporizados, a cardinalidade não deve levar em conta os pontos do tempo, mantendo a mesma semântica do modelo ER convencional.

Várias extensões à abordagem entidade relacionamento original têm sido propostas com o objetivo de incorporar a possibilidade de modelar propriedades temporais, entre as quais se destacam: a abordagem ERT (*Entity Relationship Time Model*) [Loucopoulos 91], a abordagem TER (*Temporal Entity Relationship Model*) [Touzovich 91], a abordagem TEER (*Temporal Enhanced Entity Relationship Model*) [Elmasri 93], e a abordagem TempER [Antunes 97].

2.2.2.1 - O Modelo TempER

O objetivo que levou ao desenvolvimento deste modelo é atender a todos os requisitos citados na Seção anterior. O modelo foi concebido com base, principalmente, no modelo ERT. As principais diferenças entre as abordagens situam-se na simbologia e na primitiva temporal adotada – o elemento temporal

no lugar do intervalo de tempo. Em uma visão geral, as principais características do modelo TempER são as seguintes:

- oferece uma simbologia que diferencia elementos temporizados de elementos não temporizados, semelhante à do modelo ERT;
- permite que se associe em um mesmo diagrama entidades temporalizadas com não temporalizadas. As entidades não temporalizadas passam a ser denominadas de “perenes”, sendo assumido que estas também apresentam uma dimensão temporal implícita, igual a todo o conjunto de pontos do eixo temporal. As entidades temporalizadas passam a ser denominadas “transitórias”;
- qualquer que seja a classificação das entidades em relação ao tempo, sejam elas perenes ou transitórias, ortogonalmente sempre apresentam duas perspectivas: uma não temporal e uma temporal. Quando se focaliza os conjuntos de entidades pela perspectiva não temporal estes apresentam apenas duas dimensões (tuplas x atributos não temporais). Por outro lado, quando se focaliza estes mesmos conjuntos pela perspectiva temporal, eles apresentam três dimensões (tuplas x atributos temporais x eixo temporal);
- no tocante aos relacionamentos, ou as entidades se associam entre si na perspectiva temporal (relacionamentos temporais) ou na perspectiva não temporal (relacionamentos não temporais);
- possibilita que as restrições de cardinalidade levem em consideração os momentos do tempo de validade de um relacionamento temporal;
- faz uso de um dicionário de dados para descrever os atributos, evitando que estes sejam explicitados graficamente. Isto contribui para tornar os diagramas mais administráveis visualmente.

O modelo TempER é o modelo escolhido para a modelagem do banco de dados temporal desenvolvido neste trabalho.

2.3 – Consultas Temporais

As consultas podem apresentar diferentes formas quando bancos de dados temporais são utilizados. Uma linguagem de consulta temporal faz-se necessária a fim de explorar ao mesmo tempo os rótulos temporais de um BDT. Nos últimos anos várias linguagens de consulta temporal foram propostas [Moreira, 1999].

A maioria dos modelos de dados temporais apresentam linguagens de consulta textuais, geralmente derivadas do SQL. Dentre elas, a mais conhecida é TSQL2.

O fator tempo pode estar envolvido de formas diferentes em consultas temporais [Oliveira, 1994]:

- Recuperar valores de propriedades cujo domínio é temporal. Ex.: selecione o valor da propriedade que armazena a data de nascimento de uma pessoa;
- Referir-se a um determinado instante ou a um intervalo temporal. Ex.: selecione as notas no 2º semestre de 2000;
- Recuperar valores com base em restrições temporais. Ex.: recuperar todos os valores da nota do aluno antes do dia 28/06/2001;
- Fornecer informações temporais (datas, intervalos).

2.3.1 - Problemas no Processamento de Consultas Temporais

Vários problemas são encontrados no processamento de consultas comuns. Quando se trata do processamento de consultas temporais, além dos citados, este apresenta especificamente os seguintes problemas:

- (1). necessidade de novos métodos de indexação (estrutura e algoritmos de busca) devido ao grande volume de dados armazenados em BDT. Métodos tradicionais de indexação só podem ser utilizados para valores com algum tipo de ordenação completa e com estruturas de acesso a intervalos;
- (2). manipulação de informações incompletas, devido a valores incompletos ou inexistentes. Podem ser devido à incerteza quanto a existência de objetos em certos pontos no tempo ou à indeterminação temporal causada por eventos cujo tempo de ocorrência não é conhecido.

2.4 – *Time Series Cartridge*

Time Series Cartridge ou Cartucho de Séries de Tempo é uma extensão do SGBD *Oracle*, que foi inserido a partir da versão 8, para possibilitar o armazenamento e a recuperação de dados temporais [Oracle Corp., 2001]. Uma série de tempo é o montante de dados de entrada de rótulos temporais (*timestamps*). Cada atributo ou coluna em uma tabela tem associado um rótulo temporal, ou seja, uma marca que situa o dado ou objeto no tempo.

O *Time Series* utiliza conceitos próprios para tratar informações temporais, como a utilização ou não de calendários. Aplicações de bancos de dados temporais, basicamente são implementadas de duas formas, com uso de calendários ou sem calendários. Calendários são objetos do *Time Series* que permitem vincular as informações a datas ou a períodos de tempo definidos

[Oracle Corp., 2001]. Existem dois modelos para tratar dados de séries de tempo: o modelo de séries de tempo regular e o irregular. No modelo irregular os dados da série de tempo não estão necessariamente associados ao uso de calendários, enquanto que no modelo regular obrigatoriamente devem estar associados.

A definição da utilização de um ou outro modelo de série de tempo varia com a especificação da modelagem temporal. Geralmente, calendários não são necessários quando o rótulo temporal não tiver um padrão ou quando tiver um padrão, mas este não estiver correto.

2.4.1 – Termos para *Oracle 8i Time Series*

Data âncora: data a ser usada para estabelecer o início de um padrão e (baseado no padrão) quais rótulos temporais serão incluídos ou excluídos de um calendário. Por exemplo, um padrão de ‘0,1,1,1,1,1,0’ sobre uma frequência de dia definindo um calendário sobre os dias da semana. Se uma data âncora de 03-Jun-2002 é especificada, então o 7º dia começa em cada Domingo; e Domingos e Sábados (0) são excluídos do calendário, enquanto Segunda a Sexta-Feira (1), são incluídos no calendário.

Exceções: um rótulo temporal que não está em conformidade com um calendário padrão, mas que é significativo para o calendário. Há dois tipos de exceções: *off_exceções* e *on_exceções*.

Off_exceções: uma exceção para os bits 1 (um) no padrão, é um rótulo temporal a ser excluído do calendário, para garantir que Terça-Feira, 01-Maio-2002 (feriado), será excluída do calendário quando Terças-Feiras são normalmente incluídas, define tal data como uma *off_exceção*.

On_exceções: uma exceção para os bits 0 (zero) no padrão, é um rótulo temporal a ser incluído no calendário. Por exemplo, para garantir que Sábado,03-Fev-2002, será incluído no calendário quando normalmente Sábados são excluídos, define a data com uma *on_exeção*.

Frequência: granularidade de um calendário. As frequências suportadas são segundo, minuto, hora, dia, semana, 10-dias, quinzenal, mensal, trimestral, semestral e anual.

Padrão: repetição de frequências e uma data âncora que identifica um rótulo temporal válido para o primeiro elemento no padrão. Por exemplo, se a frequência é dia, o padrão pode ser definido como os dias da semana incluídos no calendários.

Precisão: o grau de exatidão do qual um rótulo temporal precisa ser especificado. Cada frequência tem uma precisão associada. Um rótulo temporal que não está consistente com a frequência é dito ser impreciso.

Time series: um conjunto de rótulos temporais como dados de entrada. Cada *time series* consiste de um identificador (tal como RG do cliente) e múltiplos valores temporais (datas em que dados do cliente foram alterados).

2.4.2 – Modelos no Oracle8i Time Series

2.4.2.1 – Calendários

Calendário é uma estrutura de dados que mapeia o significado de tempo humano para representação de tempo no computador. A definição de um calendário

inclui a frequência, padrão e, opcionalmente, exceções e datas limites (inferior e superior).

Os componente de um calendário são:

Frequência: uma frequência especifica a granularidade da representação de um calendário. As frequências suportadas são: segundo, minuto, hora, dia, semana, 10-dias, quinzenal, mensal, trimestral, semestral e anual. Cada frequência tem um código inteiro associado (Tabela 2.1) que é usado nas chamadas de funções.

<i>Frequência</i>	<i>Código</i>
Segundo	1
Minuto	2
Hora	3
Dia	4
Semana	5
Mês	6
Trimestre	7
Ano	8
10-dias	10
15-dias	16
Semestre	18

Tabela 2.1 - Frequências suportadas pelo Oracle Time Series e seus respectivos códigos.

Algumas frequências são flexíveis quanto à definição de data âncora padrão, outras frequências são mais restritivas.

Cada frequência tem uma precisão associada. As funções do *Oracle 8i Time Series* requerem que os rótulos temporais de entrada estejam precisos

quanto à frequência definida no calendário. A função *SetPrecision* é uma exceção, esta função pega um calendário e um rótulo temporal e retorna um rótulo temporal que esteja em conformidade com a frequência do calendário associado. Um rótulo temporal de 09-Set-1997 está impreciso se a frequência é mês ou ano e a data âncora padrão definida no calendário não é o 9º dia de algum mês. Quando se define um calendário, todos os rótulos temporais (a data âncora e todas as *on* e *off exceções*) devem estar precisos com a respectiva frequência. Um calendário não estará válido se a frequência especificada for dia e uma data âncora de 01-Jun-2000 13:00:00.

Padrão: um padrão especifica o modelo de repetição de frequências e uma data âncora que identifica um rótulo temporal válido para o primeiro elemento no padrão.

Um padrão em um calendário é especificado com um ou mais zeros e/ou inteiros. Para cada padrão representado por zeros e/ou uns, cada '1' representa um rótulo temporal válido e cada '0', um rótulo temporal inválido. Por exemplo:

- Um calendário com uma frequência de dia e um único '1' como padrão (*ORDSYS.ORDTPatternBits(1)*), define rótulos temporais para cada dia. *ORDTPatternBits* é um tipo de dado.
- Um calendário com uma frequência de dia, uma data âncora para Domingo e um padrão de um único '1' e seis '0's (*ORDSYS.ORDTPatternBits (1,0,0,0,0,0,0)*), é o mesmo que um calendário semanal onde todos os Domingos são incluídos e todos os outros dias da semana são excluídos.

Para padrões contendo um ou mais inteiros maior que 1, cada inteiro representa um intervalo que é um múltiplo da frequência. Por exemplo, um

calendário com uma frequência de dia, uma data âncora de Domingo, e um padrão de '7' (*ORDSYS.ORDTPatternBits (7)*) é o mesmo que um calendário semanal onde domingos são incluídos e todos os outros dias da semana são excluídos.

Exceções: especificam rótulos temporais que não estão em conformidade com o padrão do calendário, mas que são importantes na sua definição.

Toda série de tempo regular [Pinheiro,2002] é associada ao uso de um calendário. Utilizam – se calendários quando for conhecido o momento da inserção dos dados, através de intervalos de tempo pré-definidos associados a um rótulo temporal padrão. Os calendários são utilizados também para validar a inserção dos dados, definindo períodos, datas limites e datas não válidas para a inserção, como feriados por exemplo. A Figura 2.1 demonstra a criação de um calendário:

```
CREATE TABLE calendario_comercial of ORDSYS.ORDTCalendar;  
  
1. INSERT INTO calendario_comercial VALUES (  
2.     ORDSYS.ORDTCalendar ( 0, 'DiasRealizações', 4,  
3.     ORDSYS.ORDTPattern (  
4.         ORDSYS.ORDTPatternBits (0,1,1,1,1,1,0),  
5.         TO_DATE ('01/01/2002', 'DD/MM/YYYY')),  
6.     NULL, NULL, NULL, NULL));
```

Figura 2.1 – Criação de um calendário no *Time Series*.

Para definir um calendário, deve-se criar uma tabela que armazenará as definições do calendário e então armazenar as linhas para o calendário definido. O exemplo da Figura 2.2 cria uma tabela chamada *calendario_comercial* e define um calendário chamado *DiasRealizações*.

A tabela *calendario_comercial* tem linhas do tipo de dados *ORDTCalendar*, que é um tipo de dados já definido pelo sistema time series, como descrito:

```
CREATE TYPE ORDSYS.ORDTCalendar AS OBJECT (  
  caltype      INTEGER,           -- tipo calendário (0 = padrão)  
  name         VARCHAR2 (256),    -- nome do calendário  
  frequency    INTEGER,           -- frequência para o dia  
  pattern      ORDSYS.ORDTPattern, -- definição do padrão  
  minDate      DATE,              -- data inicial  
  maxDate      DATE,              -- data final  
  offExceptions ORDSYS.ORDTExceptions,  
  onExceptions ORDSYS.ORDTExceptions);
```

Figura 2.2 – Descrição do tipo de dado *ORDTCalendar*.

Os valores *minDate* e *maxDate* podem ser nulos. Fazer esta especificação simplifica a manutenção do calendário, mas não permitirá a validação dos rótulos temporais de entrada quanto ao intervalo de validação do calendário.

2.4.2.2 – Modelos de Série de Tempo

A utilização ou não de calendários é definida quanto a existência ou não de um padrão. São poucos os modelos do *Time Series*. Porém, estes modelos são suficientes para enquadrar a maioria dos usuários.

Existem dois modelos básicos [Oliveira, 2001]:

Nenhuma necessidade de calendário: muitos usuários *Oracle8i Time Series* não necessitam usar calendários com seus rótulos temporais. Situações onde calendários não são necessários incluem as seguintes:

- os rótulos temporais não têm um padrão. Exemplo: transações em um caixa eletrônico;
- os rótulos temporais têm um padrão, mas não é necessário checá-los (deve-se assumir que todos rótulos temporais estão corretos). Uma variação de ter um padrão, mas não necessitar usá-lo ocorre quando os rótulos temporais contêm elementos não-definidos. Por exemplo: um utilitário elétrico pode querer coletar periodicamente a demanda de energia elétrica consumida por diferentes regiões, mas não é importante se o rótulo temporal é exatamente a hora ou contém minutos e segundos. Usando *SYSDATE* (data corrente do sistema) para criar os rótulos temporais podem resultar em dados para 16:00 sendo armazenados com um rótulo temporal de 15:59:37, 16:00:03, 16:01:30. Um *time series* usado sem um calendário é chamado um *irregular time series*.

Necessidade de calendários: usuários necessitam usar calendários com seus rótulos temporais quando pretendem usar funções específicas do pacote. Entretanto, dependendo destas necessidades, o usuário não precisa especificar certos elementos na definição de um calendário, tais como:

- Limite inferior e superior de datas para o calendário (que permitem usar funções *time series* para garantir que todos rótulos temporais estão contidos em intervalo definido);
- Rótulos temporais de exceções (por exemplo, feriados, greves escolares, etc) .

Capítulo 3

As Implementações

Existem poucos sistemas de Bancos de Dados Temporais realmente utilizáveis apesar das intensas pesquisas desenvolvidas nos últimos anos. Na realidade, o que temos são várias experiências sob forma de protótipos, nos quais se baseiam estudos de problemas encontrados (de armazenamento e recuperação de informações), e mapeamento de modelos temporais para BD tradicionais, nos quais os rótulos temporais são explicitamente representados e manipulados.

Segundo [Edelweiss 94], a implementação do conceito de tempo pode ser realizada de três formas de acordo com o grau de integração crescente do conceito de tempo no SGBD:

- (1). a manipulação dos dados temporais é realizada explicitamente pelo usuário. O SGBD só pode armazenar dados dos tipos tradicionais como inteiros, *strings*, reais, etc. Toda a semântica associada ao tempo está

contida na lógica dos programas de aplicação. Neste nível o usuário deve conhecer a semântica associada ao tempo e assegurar a validade das operações sobre os dados temporais;

- (2). a manipulação dos dados temporais é realizada por meio de ações associadas a propriedades definidas como temporais. Isto corresponde a extensões semânticas de tipos de dados normais. Esta solução pode ser aplicada em SGBD extensíveis pela definição de ações semânticas associadas a tipos de dados temporais. Neste caso todas as aplicações compartilham o código associado aos novos tipos de dados. A grande fraqueza é o isolamento entre as operações e o esquema conceitual. É impossível representar as propriedades temporais no esquema conceitual pois a semântica temporal é definida por modificações na manipulação de dados tradicionais (reais, *strings*). Uma solução deste tipo é apresentada como uma extensão do SGBD *INGRES* [Overmeyer 82];
- (3). as propriedades temporais são tratadas por uma extensão do modelo de dados e da linguagem de manipulação. Neste caso a semântica temporal se torna estrutural, isto é, ela pertence ao modelo de dados, e portanto, não pode ser alterada pelas aplicações. A definição de um esquema conceitual inclui as propriedades temporais. O principal inconveniente consiste na necessidade de ser desenvolvida uma nova versão do SGBD incluindo as extensões.

A utilização de um modelo de dados temporal no nível de modelagem, com grande poder de expressão, não implica que seja necessária a existência de um banco de dados correspondente a este modelo. Podem ser utilizados modelos comerciais, com todas as características de robustez que apresentam, desde que seja possível mapear adequadamente o modelo temporal para o modelo do BD utilizado. No entanto, o conceito de Banco de Dados Temporal supõe que o

conceito de tempo seja implícito como no caso da terceira categoria apresentada acima. Segundo [Tansel 93], acrescentar suporte temporal a um SGBD ocasiona um impacto importante sobre todos os seus componentes. Uma Arquitetura simplificada de um SGBD convencional é apresentada na Figura 3.1. O administrador do BD (DBA) e sua equipe projetam o BD, gerando o esquema conceitual físico. Este esquema, expresso em uma linguagem de definição de dados (DDL), é processado pelo compilador desta linguagem e armazenado no catálogo do sistema. Os usuários, por sua vez, preparam suas consultas e as submetem ao processador de consultas. Cada consulta é, inicialmente, analisada léxica e sintaticamente, com base nas informações que constam do catálogo do sistema, sendo depois otimizada, para que sua execução seja eficiente. Um plano de avaliação da consulta é enviado ao avaliador da consulta. Enquanto a consulta está sendo avaliada, este componente acessa o BD através do gerenciador de dados armazenados, o qual implementa controle de concorrência, gerenciamento de transações, *recovery*, *buffering* e método de acesso a dados.

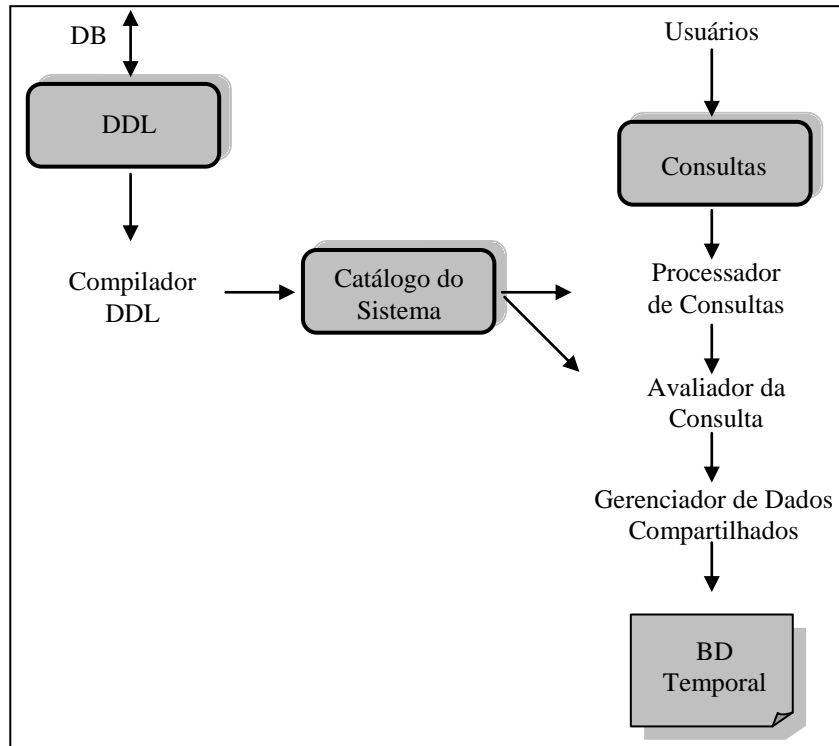


Figura 3.1 – Arquitetura simplificada de um SGBD tradicional [Tansel 93]

A introdução de informações temporais implícitas ao BD influencia cada uma das partes do SGBD, que devem ser adaptadas para permitir consultas eficientes.

As técnicas de implementação de BDT encontradas neste trabalho, seguem as formas (1) e (3) citadas no início deste Capítulo, o qual encontra-se subdividido em três Seções: na primeira serão descritas informações sobre a aplicação escolhida para implementação do BD Temporal; a segunda descreve a primeira técnica de implementação utilizada (1), assim como a modelagem e dicionário de dados para a aplicação seguindo esta técnica; e na terceira, a descrição da segunda técnica utilizada (3) também com sua modelagem e dicionário de dados para a aplicação.

3.1 - A Aplicação

A aplicação escolhida para o desenvolvimento do projeto é uma aplicação para cadastros e solicitação de serviços de uma Produtora de Eventos. Trata-se de uma aplicação para Web, onde pessoas físicas ou empresas poderão contratar a Produtora para realização de congressos, seminários, simpósios e qualquer outro evento cujas necessidades a Produtora possa suprir. Esta aplicação foi projetada de maneira simplificada pois o objetivo é apenas obter características temporais para o desenvolvimento de um BDT utilizando as duas técnicas sugeridas.

A aplicação consiste dos seguintes elementos:

- ***formulário para cadastro de clientes:*** quando um cliente estiver solicitando serviços da Produtora de Eventos pela primeira vez, ele deve, inicialmente, preencher o formulário com seus dados pessoais ou dados da empresa para a qual deseja contratar o serviço. Se o cliente desejar, ele também pode alterar seus dados caso estes tenham sofrido mudanças durante o período em que o cliente não solicitou serviços;
- ***formulário para cadastro do evento:*** neste formulário, o cliente deve fornecer informações sobre o evento a ser realizado como o período, finalidade do evento, etc. Nesta mesma operação, o cliente pode solicitar reservas de hotéis, locais e transporte necessários durante o evento. O sistema oferece ao cliente a opção de escolher qual hotel, local ou transporte devem ser reservados. No entanto, as escolhas poderão ser feitas automaticamente pelo sistema (à partir de cadastros já existentes no banco de dados), bastando para isto o consentimento do cliente;
- ***formulário para transporte:*** quando o cliente precisar de transporte em vários momentos durante o evento, um novo cadastro é requisitado onde todos os horários, dias e capacidade recomendável do veículo são solicitados;

- ***solicitação de eventuais mudanças:*** existe uma opção na aplicação para que o cliente altere informações relacionadas ao evento. Porém, esta opção estará disponível somente durante o período antecedente a um dia da realização do evento. Modificações após este prazo devem ser solicitadas através de contato direto via e-mail ou telefone.

A aplicação possui uma série de informações de caráter temporal. Por exemplo, dados do cliente como telefone, endereço, e-mail, podem sofrer várias modificações durante um período em que o cliente não utilizar os serviços da produtora. Portanto, estes dados terão um novo valor com o passar do tempo sendo necessária a atualização dos mesmos quando o cliente solicitar novos serviços. Outro exemplo, são as datas relativas ao dia do cadastro, início e término dos eventos, necessárias para vincular as informações aos valores temporais que indicam o instante de tempo em que a informação começou a existir no banco de dados (tempo de transação) e o instante de tempo ou período o qual a informação será correspondente ao mundo real (tempo de validade).

Além das informações, um outro aspecto temporal pode ser notado na aplicação: a existência das reservas. Por exemplo, a existência da reserva de hotel está diretamente ligada a existência do evento para o qual o hotel foi reservado, ou seja, a reserva do hotel terá tempo de validade igual ao tempo de validade do evento. No entanto, a relação de reserva de transporte com o evento difere um pouco. Várias reservas de transporte poderão ser realizadas durante o período de existência do evento. Portanto, uma reserva de transporte terá como tempo de validade um conjunto de intervalos de tempo que indicam todas as vezes que o transporte será utilizado em determinado evento. Isto permite que um mesmo transporte seja reservado para vários eventos que estejam acontecendo ao mesmo tempo, desde que os intervalos de tempo relativos a um evento não coincidam com nenhum dos intervalos de tempo de outros eventos.

De acordo com as duas técnicas de implementação de BDT utilizadas, a aplicação será desenvolvida em duas versões:

- (1). esta versão deverá manipular dados temporais de maneira explícita, onde o usuário deverá fornecer todas as informações de tempo necessárias (tempo de validade e/ou tempo de transação). Esta versão será utilizada para lidar com um BD relacional que possui dados temporais como atributos comuns de suas tabelas. Para manter a consistência do banco de dados, serão associados gatilhos (*triggers*) às ações do usuário como realização de cadastro (inserção), atualização de algum dado ou remoção, isto para garantir que uma ação do usuário não interfira na integridade dos dados;
- (2). esta versão da aplicação foi desenvolvida de maneira simples de modo que todo o controle de tempo no banco de dados seja realizado pelo SGBDT. Apenas os valores de tempo de validade associados aos atributos temporais serão fornecidos pelo usuário.

3.2 - As Técnicas

3.2.1 - Implementação de um BDT Utilizando SGBD Relacional

A primeira técnica utilizada consiste na implementação de um BDT à partir do desenvolvimento de um banco de dados relacional onde as características temporais serão representadas por atributos comuns nas tabelas. Para a aplicação escolhida, a seguinte modelagem foi desenvolvida:

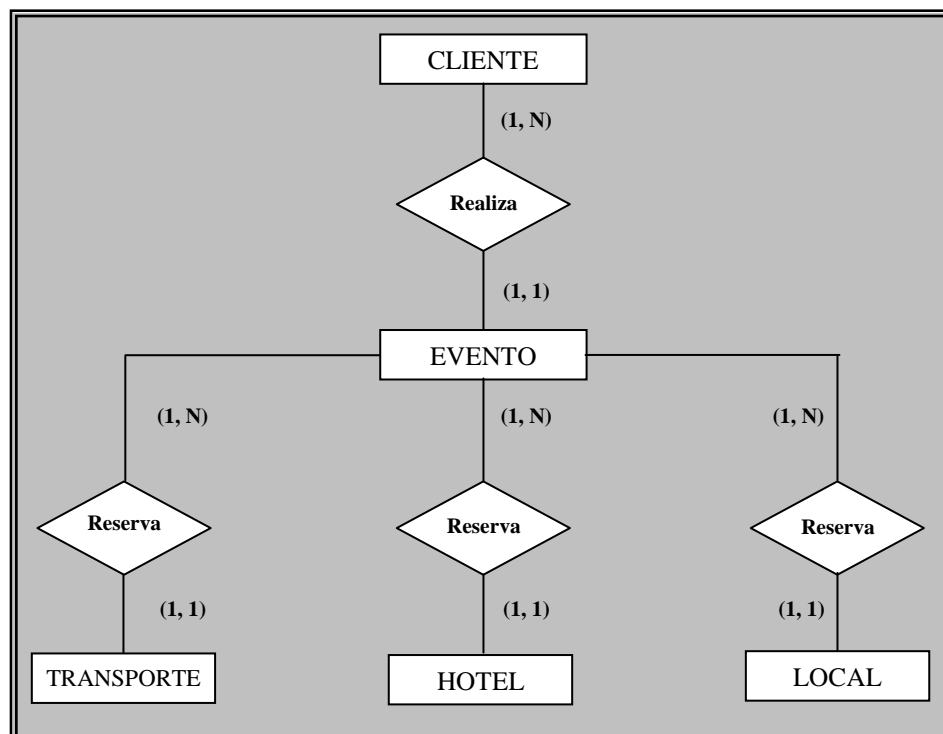


Figura 3.2 – Modelagem para BDT utilizando um banco de dados relacional e modelagem ER.

Na modelagem ER, as características temporais não são explícitas. Entidades e relacionamentos são comuns seguindo os conceitos tradicionais de BD. Informações de tempo que estão relacionadas aos dados do BD, como tempo de validade e tempo de transação, são armazenadas nas tabelas como atributos simples e fornecidos pelo usuário através da aplicação. O dicionário de dados abaixo descreve como são formadas as tabelas para o BD da aplicação:

Entidade: CLIENTE	Atributos:	
	nome_cl	VARCHAR;
	endereco_cl	VARCHAR;
	cidade_cl	VARCHAR;
	estado_cl	CHAR;
	cep_cl	CHAR;
	telefone_cl	CHAR;
email_cl	VARCHAR;	

	rg_cl CHAR; cpf_cl CHAR; profissão_cl VARCHAR; data_nasc_cl DATE; t_validade_inic_cl DATE; t_validade_final_cl DATE. Chave primária: rg.
Entidade: EVENTO	Atributos: codigo_even CHAR; título_even VARCHAR; dia_início_even DATE; dia_termino_even DATE; t_transação_even DATE; t_validade_inic_even DATE; t_validade_final_even DATE. Chave primária: codigo_even.
Entidade: HOTEL	Atributos: codigo_h CHAR; nome_h VARCHAR; endereço_h VARCHAR; cidade_h VARCHAR; estado_h CHAR; cep_h CHAR; telefone_h CHAR; email_h VARCHAR; categoria_h VARCHAR; preço_diária_h CHAR; t_validade_inic_h DATE; t_validade_final_h DATE. Chave primária: codigo_h.
Entidade: LOCAL	Atributos: codigo_lc CHAR; estabelecimento_lc VARCHAR; endereço_lc VARCHAR; cidade_lc VARCHAR; estado_lc CHAR; cep_lc CHAR; telefone_lc CHAR; preço_diária_lc CHAR; t_validade_inic_lc DATE; t_validade_final_lc DATE. Chave primária: codigo_lc.
Entidade: TRANSPORTE	Atributos:

	<p> codigo_tr CHAR; tipo_tr VARCHAR; capacidade_tr CHAR; preço_diária_tr CHAR; t_validade_inic_tr DATE; t_validade_final_tr DATE. </p> <p>Chave primária: codigo_tr</p>
Entidade: REALIZA_EVENTO	<p> Atributos: codigo_rlz_even ref codigo_even; cliente_rlz_even ref rg_cl; t_transação_rlz_even ref t_transação_even; t_validade_inic_rlz_even ref dia_início_even; t_validade_final_rlz_even ref dia_término_even. </p> <p>Chave composta: codigo_rlz_even + cliente_rlz_even .</p>
Entidade: RESERVA_HOTEL	<p> Atributos: codigo_rh ref codigo_even; hreservado_rh ref codigo_h; t_transação_rh ref t_transação_even; t_validade_inic_rh ref dia_início_even; t_validade_final_rh ref dia_término_even. </p> <p>Chave composta: codigo_rh + hreservado_rh.</p>
Entidade: RESERVA_LOCAL	<p> Atributos: codigo_rlc ref codigo_even; lreservado_rlc ref codigo_lc; t_transação_rlc ref t_transação_even; t_validade_inic_rlc ref dia_início_even; t_validade_final_rlc ref dia_término_even. </p> <p>Chave composta: codigo_rlc + lreservado_rlc.</p>
Entidade: RESERVA_TRANSPORTE	<p> Atributos: codigo_rtr ref codigo_even; treservado_rtr ref codigo_tr; t_validade_dia_rtr DATE; t_validade_hor_inicial_rtr TIME; t_validade_hor_final_rtr TIME. </p> <p>Chave composta: codigo_rtr + treservado_rtr.</p>

Tabela 3.1 – Dicionário de Dados para Modelagem ER.

Conforme a Tabela 3.1, as entidades *CLIENTE*, *EVENTO*, *HOTEL*, *LOCAL* e *TRANSPORTE* possuem o atributo *t_validade_inicial*. Este atributo representa o momento em que cada uma das tuplas de uma tabela começa a existir (tempo de transação) e ao mesmo tempo, o instante a partir do qual a tupla começa ser válida como informação (tempo de validade inicial). Isto se deve ao fato de o tempo de transação e tempo de validade inicial possuírem o mesmo valor temporal. O atributo *t_validade_final* é o valor final do tempo de validade das tuplas e será calculado baseando-se no tempo de transação e no tempo em que aquela informação permanecer inalterada.

Os atributos *t_validade_hor_inicial* e *t_validade_hor_final*, presentes na entidade *RESERVA_TRANSPORTE*, indicam o intervalo de tempo no qual a tupla será válida como informação do presente, ou seja, o transporte será utilizado no evento. Este período está associado também ao dia em que o transporte será utilizado. Portanto, o tempo de validade desta entidade é uma associação de três informações.

As entidades *RESERVA_LOCAL* e *RESERVA_HOTEL* não possuem nenhum valor temporal explícito e sim, uma associação à valores temporais de outra entidade (*referências*). Isto se deve ao fato de a existência de uma instância (ou tupla) destas entidades depender da criação de uma instância da entidade *REALIZA_EVENTO*, ou seja, a criação de um objeto é disparada quando outro objeto começar a existir no BD (*gatilho*). Outro exemplo deste caso é quando um cadastro de *EVENTO* é realizado pois, neste momento, é criada a entidade *REALIZA_EVENTO* herdando o tempo de transação da entidade anterior. A figura abaixo representa esta série de disparos para melhor compreensão do leitor:

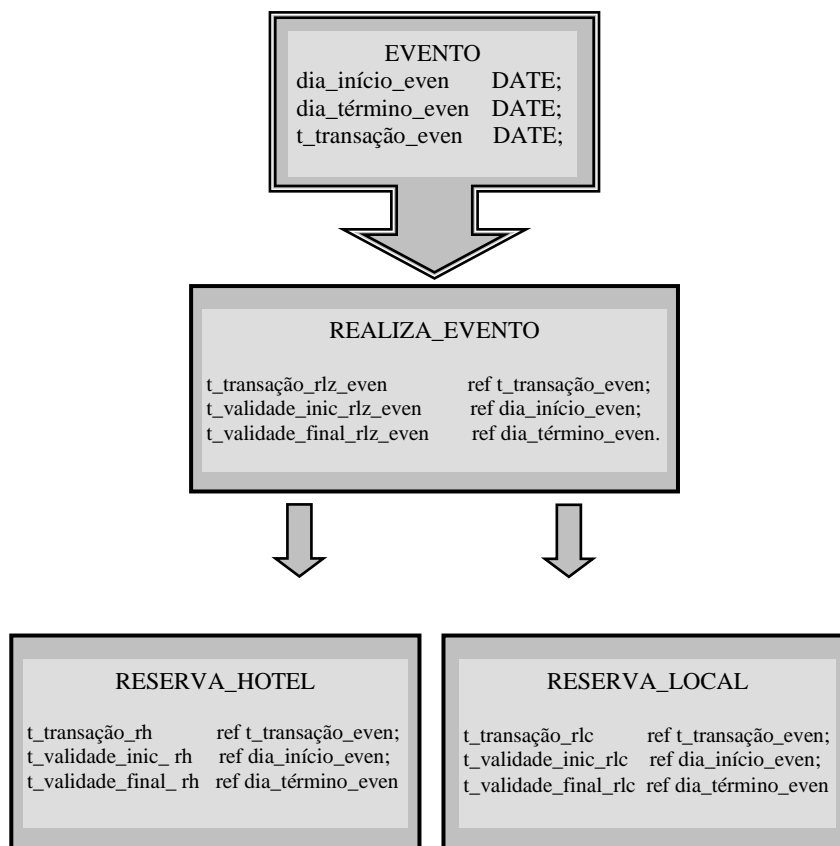


Figura 3.3 – Representação da seqüência de disparos após criação da entidade evento

Uma questão importante que deve ser observada é com relação à possíveis alterações. Uma modificação de um dado em qualquer cadastro, seja de hotel, cliente ou evento, implica na inserção de uma nova tupla na tabela com os valores atuais. Para que os dados anteriores continuem armazenados no BD e a acessibilidade e integridade dos mesmos seja mantida, algumas operações devem ser executadas:

- uma alteração significa inserção de uma nova tupla na tabela com algum dado modificado. Portanto, os dados que permaneceram inalterados serão simplesmente copiados para a nova tupla, incluindo a chave primária. No entanto, duas tuplas em uma tabela nunca podem ter um mesmo valor para chave primária. O que deve ser feito então é o uso de algum recurso para modificar a chave primária da tupla desatualizada, de forma que elas fiquem com chaves primárias semelhantes e não idênticas. A tupla atual deve ficar com o valor original da chave primária para que, quando uma consulta for efetuada à partir da mesma, os dados mais atuais sejam retornados. Por exemplo, um cliente que mudou de endereço três vezes, terá três tuplas diferentes no BD sendo que, a tupla com endereço atual terá associado o valor do seu RG (chave primária) e, as tuplas com os endereços anteriores terão o valor do RG do cliente acrescentado de um valor numérico ou um caractere qualquer. Esta operação de modificação (das chaves primárias de tuplas desatualizadas) é realizada através de gatilhos disparados no momento em que o cliente finaliza a alteração.
- como consequência da operação anterior, uma forma de consulta aos estados passados de um dado, seria utilizando o comando LIKE. Este comando retorna todas as tuplas cujo atributo contém um valor especificado. Por exemplo: para o mesmo cliente acima, cujo RG é mg 9768 534, quando se desejar recuperar seus endereços passados (os quais as chaves primárias poderiam ser, por exemplo, mg 9768 534A e mg 9768 534B, respectivamente), basta realizar a consulta :

```
SELECT nome, endereço, telefone  
FROM CLIENTE  
WHERE rg LIKE "mg 9768 534".
```

Tanto os endereços antigos como os atuais serão disponibilizados.

- no caso representado na Figura 3.3, alterações na entidade EVENTO implicariam não só nas operações acima mas também, na extinção dos objetos disparados pela tupla modificada. Portanto, quando um dado for alterado e uma nova tupla para determinado evento for inserida no BD, os objetos disparados pela tupla anterior serão excluídos e somente a tupla atual possuirá estes objetos. Neste aspecto, dois pontos devem ser observados: a exclusão dos objetos disparados por tuplas modificadas é uma maneira simples encontrada para eliminação de dados inúteis à aplicação (um evento modificado não é mais o evento que será realizado); modificações nos dados de um evento só poderão ocorrer até um dia antes de seu acontecimento (isto é determinado pela aplicação, que exige comunicação direta para alterações após este prazo ou durante a realização do evento).

Todas essas operações citadas acima são implementadas usando gatilhos disparados pelas as atitudes do usuário da aplicação.

Seguindo os conceitos de temporização de informações em BD, todas as tuplas das tabelas têm valores para o tempo de transação e para o tempo de validade, algumas vezes representados simultaneamente pelo tempo de validade inicial. No entanto, através deste modelo de implementação podemos perceber que, mesmo quando um único atributo tiver seu valor alterado, toda a tupla será copiada associada a seus novos tempo de transação e tempo de validade, consequência do fato dessas informações temporais estarem ligados a tupla inteira. Isto nos faz perceber que um grande número de informações “repetidas” será armazenado no BD. Uma outra forma de implementação de BDT em um

SGBD relacional será sugerida no próximo capítulo como melhoria deste problema.

O banco de dados descrito foi implementado no SGBD relacional *Oracle8i*. Orientação à objeto foi usada na criação do atributo composto *ENDEREÇO* (endereço, cidade, estado, cep). As entidades e seus relacionamentos são tabelas comuns, cada qual com seus atributos e restrições.

3.2.2 - Implementação de um BDT Utilizando um SGBD Temporal - *Oracle Time Series Cartridge*

A segunda técnica de implementação utiliza a ferramenta *Oracle Time Series Cartridge 8i*. Esta ferramenta é uma extensão do SGBD *Oracle 8i*, que provê armazenamento e recuperação de dados temporais de tempo de transação [Oracle Corp., 2001]. É um conjunto de procedimentos, funções e objetos que foram desenvolvidos no intuito de atender às exigências de aplicações específicas e tornar mais conveniente e eficaz o tratamento de dados de tempo, do que formas tradicionais e funções definidas por usuários [Pinheiro, 2002]. No *Time Series* o tempo é caracterizado através da inserção de um rótulo temporal. Basicamente qualquer operação sobre qualquer dado que envolva informação temporal é obtida através do rótulo temporal. Este tipo de procedimento caracteriza o *Time Series Cartridge* como uma ferramenta de gerenciamento de informações de tempo de transação.

Para a aplicação Promotora de Eventos, a seguinte modelagem temporal do BDT foi desenvolvida, utilizando a modelagem TempER, extensão da modelagem ER:

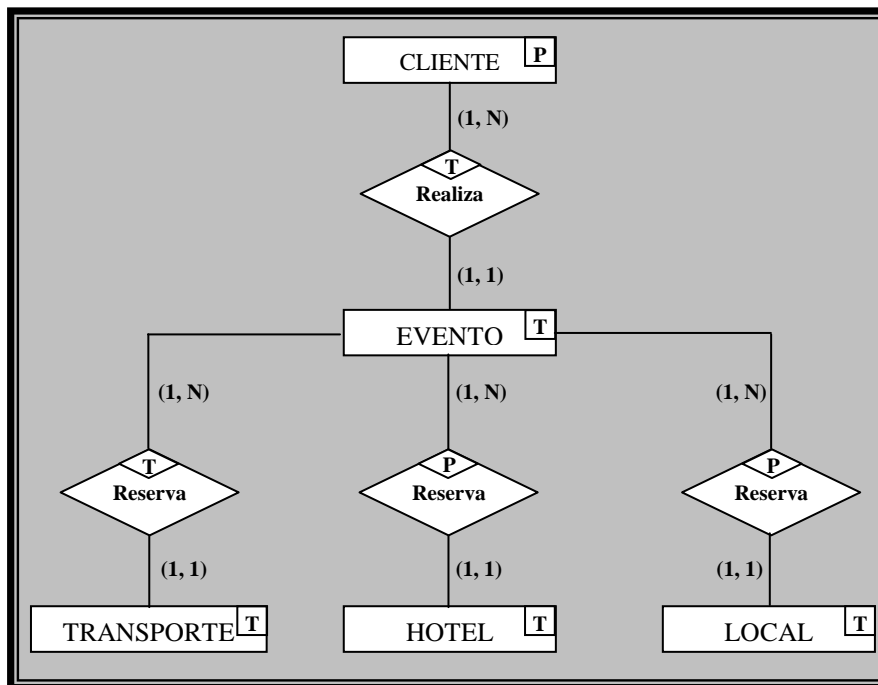


Figura 3.4 – Modelagem do Banco de Dados Temporal utilizando modelo TempER

Na modelagem TempER, no tocante aos relacionamentos, ou as entidades se associam entre si na perspectiva temporal (relacionamentos temporais) ou na perspectiva não temporal (relacionamentos não temporais). A modelagem da Figura 3.4 possui dois relacionamentos perenes, *reserva_hotel* e *reserva_local*. Esta classificação é em decorrência da dependência temporal desses relacionamentos com relação a entidade *EVENTO*. O tempo de transação e o tempo de validade desses relacionamentos dependem diretamente de valores temporais relativos a essa entidade. Abaixo é descrito o dicionário de dados da aplicação para esta modelagem:

Entidade: CLIENTE	Atributos: nome_cl endereço_cl cidade_cl	VARCHAR Temporal; VARCHAR Temporal; VARCHAR Temporal;
-------------------	---	---

	estado_cl CHAR Temporal; cep_cl CHAR Temporal; telefone_cl CHAR Temporal; email_cl VARCHAR Temporal; rg_cl CHAR Intemporal; cpf_cl CHAR Intemporal; profissão_cl VARCHAR Temporal; data_nascimento_cl DATE Intemporal; Chave primária: rg_cl.
Entidade: EVENTO	Atributos: codigo_even CHAR Intemporal; título_even VARCHAR Temporal; dia_início_even DATE Temporal; dia_termino_even DATE Temporal; Chave primária: codigo_even.
Entidade: HOTEL	Atributos: codigo_h CHAR Intemporal; nome_h VARCHAR Temporal; endereço_h VARCHAR Intemporal; cidade_h VARCHAR Intemporal; estado_h CHAR Intemporal; cep_h CHAR Intemporal; telefone_h CHAR Temporal; email_h VARCHAR Temporal; categoria_h VARCHAR Temporal; preço_diária_h REAL Temporal; Chave primária: codigo_h.
Entidade: LOCAL	Atributos: codigo_lc CHAR Intemporal; estabelecimento_lc VARCHAR Temporal; endereço_lc VARCHAR Intemporal; cidade_lc VARCHAR Intemporal; estado_lc CHAR Intemporal; cep_lc CHAR Intemporal; telefone_lc CHAR Temporal; preço_diária_lc REAL Temporal; Chave primária: codigo_lc.
Entidade: TRANSPORTE	Atributos: codigo_tr CHAR Intemporal; tipo_tr VARCHAR Intemporal; capacidade_tr CHAR Intemporal; preço_diária_tr REAL Temporal;

	Chave primária: codigo_tr
Entidade: REALIZA_EVENTO	Atributos: codigo_rlz_even ref codigo_even Intemporal; cliente_rlz_even ref rg_cl Intemporal. Chave composta: codigo_rlz_even + cliente_rlz_even.
Entidade: RESERVA_HOTEL	Atributos: codigo_rh ref codigo_even Intemporal; hreservado_rh ref codigo_h Intemporal. Chave composta: codigo_rh + hreservado_rh.
Entidade: RESERVA_LOCAL	Atributos: codigo_rlc ref codigo_even Intemporal; lreservado_rlc ref codigo_lc Intemporal. Chave composta: codigo_rlc + lreservado_rlc.
Entidade: RESERVA_TRANSPORTE	Atributos: codigo_rtr ref codigo_even Intemporal; treservado_rtr ref codigo_tr Intemporal; t_validade_dia_rtr DATE Temporal; t_validade_hor_inicial_rtr TIME Temporal; t_validade_hor_final_rtr TIME Temporal; Chave composta: codigo_rtr + treservado_rtr.

Tabela 3.2 - Dicionário de Dados para Modelagem TempER.

São necessárias duas granularidades diferentes para a aplicação: dia e hora. Portanto dois calendários serão criados para associar as tabelas. As Figuras 3.5 e 3.6 descrevem a criação destes calendários:

```

CREATE TABLE calendario_eventos of ORDSYS.ORDTCalendar;

1. INSERT INTO calendario_eventos VALUES (
2.     ORDSYS.ORDTCalendar ( 0, 'DiasEvento', 4,
3.     ORDSYS.ORDTPattern (
4.         ORDSYS.ORDTPatternBits (1,1,1,1,1,1,1),
5.         TO_DATE ('01/01/2002', 'DD/MM/YYYY')),
6.     NULL, NULL, NULL, NULL));

```

Figura 3.5 – Calendário para Promotora de Eventos com granularidade de dia.

```

CREATE TABLE calendario_eventos of ORDSYS.ORDTCalendar;

1. INSERT INTO calendario_eventos VALUES (
2.     ORDSYS.ORDTCalendar ( 0, 'HorarioTransporte', 3,
3.     ORDSYS.ORDTPattern (
4.         ORDSYS.ORDTPatternBits (1,1,1,1,1,1,1),
5.         TO_DATE ('01/01/2002', 'DD/MM/YYYY')),
6.     NULL, NULL, NULL, NULL));

```

Figura 3.6 – Calendário para Promotora de Eventos com granularidade de hora.

3.2.2.1 - *Time Series* e Tempo de Validade

A importância da inserção do tempo de validade no processo de modelagem da realidade é inquestionável. Atualmente, não existem SGBD com características temporais ou ferramentas de apoio a inserção de dados de tempo, que tratem este tempo. Portanto, também o *Time Series Cartridge* não fornece suporte ao tempo de validade, mas permite que funções e procedimentos sejam criados ou estendidos.

Em decorrência do não tratamento de tempo de validade pela ferramenta *Time Series Cartridge*, o BDT não pôde ser desenvolvido com sucesso, uma vez que a informação de maior importância para a aplicação é justamente o tempo de validade associado aos dados temporais e todas as operações realizadas sobre o BDT dependem da mesma. Portanto, a conclusão desta técnica de implementação não foi possível.

Capítulo 4

Estudo Comparativo

As comparações apresentadas neste Capítulo, foram realizadas com embasamento no que foi desenvolvido até então e em toda a pesquisa e estudo realizados no decorrer do trabalho. Os critérios de avaliação utilizados foram definidos ao longo do estudo, de acordo com as dificuldades que foram sendo encontradas.

4.1 – Tratamento do Tempo de Validade

Time Series não é um SGBD totalmente temporal pois não trata, por exemplo, de tempo de validade. Quando este tipo de informação temporal é importante para a aplicação e precisa ser utilizado, torna-se necessário a utilização de técnicas convencionais para inserir estes valores no BD. Isto faz parecer que voltamos ao ponto inicial: implementação de BD Temporais utilizando um SGBD relacional.

A associação entre a ferramenta *Time Series* e técnicas convencionais para inserção e manipulação de tempo de validade não significa, necessariamente, mais trabalho para o desenvolvedor. Aplicações SIGs (Sistemas de Informação Geográficas) [Pinheiro, 2002] realizam acompanhamento de fenômenos naturais em tempo real, como o registro automático de condições climáticas em um certo local, tornando o momento de inserção do dado essencial para estudos posteriores e previsões climatológicas. Portanto, o controle de tempo de transação em um SIG implementado sobre o *Time Series* é simples e direto, já que o processo de inserção de um rótulo temporal aos dados é feito automaticamente bastando especificar uma granularidade no calendário. Conseqüentemente, o desenvolvedor encontrará muitas vantagens ao fazer uso dessa associação, uma vez que o ganho obtido com o uso de *Time Series* ultrapassa o esforço gasto na implementação de tempo de validade na forma tradicional.

No caso da aplicação em estudo, o mesmo processo não deve ser adotado. Ao longo do desenvolvimento da aplicação, pudemos constatar que o tempo de validade é a informação temporal de maior relevância para os dados. O tempo de transação é utilizado apenas para calcular o tempo de validade em algumas entidades e relacionamentos. Portanto, a implementação do BDT utilizando a técnica convencional torna-se mais atrativa, uma vez que os recursos do *Time Series* não serão de grande utilidade para a aplicação.

Podemos concluir então que, o grau de importância do tempo de validade e do tempo de transação da aplicação, devem ser avaliados antes de se adotar a ferramenta *Time Series* no desenvolvimento do BDT. Sugerimos portanto, um estudo da aplicação antes do início de seu desenvolvimento.

4.2 - Gerenciamento de Informações

O *Time Series Cartridge* não fornece suporte ao tempo de validade, mas permite que funções e procedimentos sejam criados ou estendidos.

Time Series fornece recursos para o gerenciamento de informações temporais baseando-se no rótulo temporal. Deste modo, o tempo de validade pode ser administrado pelo usuário através de recursos do SGBD. O ponto forte do *Time Series Cartridge* são as funções e procedimentos para o tratamento de informações temporais. Estas funções auxiliam desde a especificação do modelo como a granularidade, padrões e exceções, até a recuperação das informações temporais, possibilitando comparações, agrupamentos, derivações e administração. As funções específicas para tratar o tempo foram desenvolvidas para serem aplicadas sobre atributos que representam o rótulo temporal (*timestamp*). Portanto, inserindo-se o tempo de validade no sistema em forma de um atributo convencional, basta modificar ou estender estes recursos de maneira que possam ser utilizados no tratamento do tempo de validade.

4.3 - Armazenamento de Dados

Todo atributo temporal terá um tempo de transação associado. No *Time Series*, isto acarreta em um grande número de informações a serem armazenadas, as vezes desnecessariamente. No desenvolvimento do BDT sobre um SGBD relacional podemos optar ou não pelo armazenamento dessa informação, a depender de sua importância para a aplicação.

Por outro lado, na implementação de BDT sobre um SGBD relacional, armazena informações repetidas cada vez que um dado temporal se altera (como foi explicado na Seção 3.2.1).

4.4 - Consistência em Banco de Dados

Manter a consistência de um banco de dados é um aspecto muito importante a ser tratado. Quando inserido uma ou mais dimensões temporais aos dados o SGBD deve garantir que estes permaneçam consistentes durante todo o ciclo de permanência da informação no banco de dados. De acordo com [Oracle Corp., 2001], para que uma série de tempo seja consistente é necessário que as seguintes condições sejam verdadeiras:

- todos os rótulos temporais sejam ordenados em ordem crescente;
- não existam rótulos temporais duplicados;
- todos os rótulos temporais tenham a mesma frequência definida no calendário;
- não exista rótulo temporal que não estão compreendidos entre a *mindate* e a *maxdate* definidas no calendário;
- que os dados da série de tempo sejam contíguos, ou seja, entre o maior e o menor valor do rótulo temporal deve haver dados para todos os valores válidos para rótulos temporais, mesmo que sejam nulos.

A consistência dos dados não é garantida pelo *Oracle Time Series*. As regras de consistência podem ser quebradas a partir de ações conscientes ou ações acidentais de usuários. Um exemplo da possibilidade de perda da integridade dos dados em uma série de tempo é uma simples execução de um comando de exclusão de dados no meio de uma série de tempo.

O *Time Series Cartridge* não faz restrição a este tipo de operação, sendo que qualquer outro tipo de controle de integridade pode ser feito através funções de usuários como gatilhos, ou funções próprias do *Oracle*. Para realizar o controle de integridade, uma opção pode ser associar gatilhos nas operações de

inserção, atualização e remoção de tuplas em todas tabelas onde necessário. Desta forma, o controle de integridade é realizado através de programação específica para cada tabela/aplicação, constituindo-se em uma solução correta, mas não genérica.

Portanto, todo o esforço para desenvolver métodos de controle de integridade em BDT é similar nas duas técnicas.

4.5 – Validação de Rótulos Temporais

Muitos usuários *Oracle8i Time Series* necessitam usar calendários para ter a completa vantagem das funções *time series*, identificar um padrão para os rótulos temporais e realizar pelo menos alguma validação dos rótulos temporais. Em um modelo definido por completo, cada *time series* tem um calendário com um padrão, começando e terminando com datas limites, e completa especificação de todas exceções. Ao adotar este modelo completo, usuários podem usar funções do pacote *time series* para determinar se qualquer dos rótulos temporais aos dados de entrada estão válidos. Por exemplo, os seguintes erros podem ser detectados:

- O ano de um rótulo temporal foi incorretamente digitado: existe um rótulo temporal de 01-Mar-2001 quando na verdade deveria ser 01-Mar-2002, e os *times series* estão restritos a dados de 2002;
- Em uma empresa, cadastrar um funcionário em um departamento antes da criação do departamento;
- Em uma escola, cadastrar as notas de um aluno em um semestre, quando na verdade aquele semestre houve greve, tendo definido um calendário com exceções;

- Para a aplicação em estudo, a solicitação de realização de um evento sem o cliente solicitante estar cadastrado.

4.6 – Facilidade de Uso

O conhecimento de conceitos básico de Banco de Dados Temporais permite que facilmente se desenvolva um BDT utilizando a primeira técnica estudada. Porém, utilizar a segunda técnica implicaria no estudo das funções e procedimentos fornecidos pelo pacote *Time Series*.

4.7 - Orientação à Objeto x Modelo Relacional

Na Seção 3.2.1, é descrita a técnica utilizada neste trabalho para desenvolver um BDT sobre um SGBD relacional. Com o decorrer do processo, foi constatado que a técnica adotada não seria a melhor para a resolução do problema.

Como foi discutido anteriormente, em um modelo relacional, quando uma alteração é efetuada no BDT uma nova tupla é inserida na tabela onde está ocorrendo a alteração. Esta operação tem como consequência um grande volume de dados armazenados desnecessariamente, uma vez que todas as informações da tupla anterior a modificação serão copiadas, mesmo as que não foram alteradas. Este fato motivou o estudo de uma outra forma de implementação de BDT: usando orientação à objeto.

Neste modelo, para cada atributo temporal é criada um “tipo de tabela” na qual atributos relativos a tempo serão inseridos. Por exemplo, para o atributo *nome* da entidade *CLIENTE*, poderia ser criado o tipo de tabela abaixo:

<i>Tempo de transação</i>	<i>Tempo de validade inicial</i>	<i>Tempo de validade final</i>	<i>Nome</i>
17/03/1980	17/03/1980	14/08/2002	Eva Ana Dias
15/08/2002	15/08/2002	<<	Eva Ana Dias Andrade

Tabela 4.1 – Tipo de tabela para o atributo *nome* da entidade *CLIENTE*.

Todos atributos temporais de uma tabela serão tipos de tabela. Portanto, toda vez que um atributo temporal sofrer uma alteração, somente no tipo de tabela do atributo que foi modificado será inserida uma nova tupla com o novo valor do atributo e seus respectivos tempo de transação, tempo de validade inicial e tempo de validade final.

Inicialmente, uma tabela estará armazenando um número maior de informações, uma vez que para cada atributo valores temporais terão que ser guardados. Mas com o passar do tempo, a medida em que os dados forem sofrendo alterações, este número vai se tornando insignificante se comparado ao número de informações que seriam armazenadas se o BDT tivesse sido implementado sem o uso dessa técnica. Principalmente se considerarmos aplicações cujos os dados são modificados com muita frequência, diariamente ou a toda hora por exemplo. É o caso de uma aplicação para contas bancárias nas quais a maioria dos dados se modifica várias vezes em curto espaço de tempo.

Outro aspecto importante em que essa técnica oferece vantagens é na consistência do BDT. A inconsistência dos dados diminui consideravelmente com o uso dessa técnica, mas ainda existe. Portanto, assim como no BDT não orientado à objeto, algumas operações (em forma de gatilhos ou não) são necessárias para se realizar o controle de consistência. No entanto, com o uso dessa técnica, o número de operações diminui significativamente.

Capítulo 5

Conclusões

A utilização de Bancos de Dados Temporais é muito freqüente nos dias de hoje, em consequência da necessidade de grande parte das aplicações atuais. Portanto, os estudos na área se intensificaram com o intuito de melhor definir conceitos e estratégias de tratamento de informações históricas. Mas BDTs são geralmente implementados utilizando-se SGBDs tradicionais, mesmo estando disponível no mercado uma ferramenta de tratamento de informações temporais (*Time Series Cartridge*).

Através do estudo realizado neste trabalho, alguns pontos foram comparados entre a implementação de BDTs sobre SGBD relacionais e SGBD estendido para tratamento de informações temporais. Vantagens e desvantagens de cada técnica foram destacadas afim de proporcionar ao leitor um

conhecimento básico que facilite a escolha de um método de implementação de BDTs.

Muito ainda deve ser feito nesta área principalmente com relação ao controle de tempo oferecido pelo próprio SGBD. Aspectos como consistência dos dados e tratamento de tempo de validade são de grande importância para a maioria das aplicações e ainda não oferecidos por esta espécie de SGBD. Isto tem como consequência o mesmo esforço por parte do desenvolvedor que na implementação sobre um SGBDs relacionais.

A evolução destas técnicas é de grande valor em áreas que necessitam do conceito de tempo, tais como SIGs, Sistemas de Suporte à Decisão, *Data Warehousing*, mineração de dados (*Data Mining*), etc.

Referência Bibliográfica

[Antunes 97] ANTUNES, D.C. *Modelagem temporal de sistemas: uma abordagem fundamentada em redes de Petri*. Porto Alegre: CPGCC da UFRGS, 1997. Dissertação de mestrado.

[Bolour 82] BOLOUR, A. et al. The Role of time in information processing: A survey. *SigArt Newsletter*, v.80, p. 28-48, April, 1982.

[Clifford 87] CLIFFORD J.; CROCKER A. The Historical relational data model (HRDM) and algebra based on lifespans. Proceedings of the Third International Conference on Data Engineering, Los Angeles, California, 1987. p. 528-537.

[Clifford 93] CLIFFORD J.; CROCKER A. The Historical relational data model (HRDM) revisited. In: A.U. TANSEL et al. (eds.) *Temporal Databases: Theory, Design, and Implementation*. Redwood City, California: Benjamin /Cummings, 1993. p.6-27.

[Edelweiss 94] EDELWEISS, N. Modelagem de Aspectos Temporais de Sistemas de Informação. In: ESCOLA DE COMPUTAÇÃO, 9.,1994, Recife. Recife: UFPE, 1994. 163p.

[Edelweiss 98] EDELWEISS, N. Bancos de Dados Temporais: Modelagem e Prática. In: INSTITUTO DE INFORMÁTICA, 1998, Rio Grande do Sul. Recife: UFRGS, 1998.

[Elmasri 93] ELMASRI, R.; WUU, G.T.J.; KOURAMAJIAN, V. A temporal model and query language for EER Databases. In: A.U. TANSEL et al. (eds.) *Temporal Databases: Theory, Design, and Implementation*. Rewood City, California: Benjamin /Cummings, 1993. p.212-229.

[Etzion 98] ETZION, O.; JAJODIA, S.; SRIPADA, E. (Eds.) Temporal Databases: Research and Practice. Berlin: Springer-Verlag, 1998. LNCS1399.

[Jensen 94] JENSEN, C.S. et al. A Consensus Glossary of Temporal Database Concepts. *SIGMOD Record*, v. 23, n.1, p. 53-63, Mar 1994.

[Kline 93] KLINE, M. Un update on the temporal database. Bibliography: Temporal databases. *ACM SIGMOD Record*, v.22, n.4, p. 66-80, Dec. 1993.

[Lorentzos 93] LORENTZOS, N.A. The Interval-extended Relational Model and its applications to valid time databases. In: A.U. TANSEL et al. (eds.) *Temporal Databases: Theory, Design, and Implementation*. Rewood City, California: Benjamin /Cummings, 1993. p.67-91.

[Loucopoulos 91] LOUCOPOULOS, P.; THEODOULIDIS, C.; WANGLER, B. *The entity relationship time model and conceptual rule language. Proceedings*

of the tenth Int. Conf. on the Entity Relationship Approach, San Mateo, California, 1991.

[McKenzie 86] MCKENZIE, M. Bibliography: Temporal databases. *ACM SIGMOD Record*, v.15, n.4, p. 40-52, Dec. 1986.

[Moreira, 99] MOREIRA, Viviane P.; EDELWEISS, N. Queries to Temporal Database Supporting Schema Versioning. XIV SBBD Florianópolis, 1999.

[Navathe 88] NAVATHE, S.B; AHMED, R. TSQL: A language interface for history databases. In: ROLLAND, C.; BODART, F.; LEONARD, M. (Eds.) *Temporal Aspects in Information Systems*. Amsterdam: North-Holland, 1998. p. 109-122.

[Navathe 93] NAVATHE, S.B; AHMED, R. Temporal extensions to the relational Model and SQL. In: A.U. TANSEL et al. (eds.) *Temporal Databases: Theory, Design, and Implementation*. Redwood City, California: Benjamin /Cummings, 1993. p.92-109.

[Oliveira, 94] OLIVEIRA, José P M.; EDELWEISS, n. Modelagem de Aspectos Temporais de Sistemas de Informação. IX Escola de Computação, Recife, 1994.

[Oliveira, 2001] OLIVEIRA, Rosiane A. Modelagem e Desenvolvimento de um Banco de Dados Temporal. Monografia de conclusão de curso. Universidade Federal de Lavras, MG, 2001.

[Oracle Corp., 2001] ORACLE Corporation, Oracle 8i Time Series User's Guide. <http://technet.oracle.com/doc/inter.815/a67294/toc.htm>

[Overmeyer 82] OVERMEYER,R.; STONEBRAKER,M. Implementation of a Time Expert in a *DataBase* System. *ACM SIGMOD Record*, v. 11, n.3, p. 51, Apr 82.

[Pinheiro,2001] PINHEIRO, S.F. *Estudo dos Mecanismos de Administração e Processamento de Informações Temporais fornecidos pelo Oracle8i Time Series Cartridge*. Trabalho Individual do curso de Mestrado. Universidade Federal do Rio Grande do Sul. Porto Alegre, 2001.

[Pinheiro, 2002] PINHEIRO, S.F.; FORNARI, M.R *Implementação de um Modelo Conceitual Temporal e Espacial Utilizando o SGBD Oracle* . ULBRA, 2002

[Pissiniou 94] PISSINIOU, N.; SNODGRASS, R.T. Towards na Infrastructure for Temporal Databases: Report of na Invitational ARPA/NSF Workshop. *Sigmod Record*. V.23, n.1, 03/1994.

[Silva 92] SILVA, R.C.; TRAINA JR. C. Armazenagem e Gerenciamento de Informações Temporais em um Modelo de Dados Orientado a Objetos. In: *7º Simpósio Brasileiro de Banco de Dados. Anais...* Porto Alegre: Instituto de Informática da UFRGS, 1992.

[Snodgrass 95] SNODGRASS, R.T. **The TSQL2 Temporal Query Language**. Boston: Kluwer Academic Publishers, 1995.

[Soo 91] SOO, M.D. Bibliography: Temporal databases. *ACM SIGMOD Record*, v.20, n.1, p. 14-23, Mar. 1991.

[Stam 88] STAM, R.; SNODGRSS, R.T. Bibliography on temporal databases. IEEE Database Eng., v.7, n.4, p.231-239, Dec. 1988.

[Tansel 90] TANSEL, A.U. Modelling Temporal Data. *Information and Software Technology*. V. 32, n. 8, p.514-520, 1990.

[Tansel 93] TANSEL, A.U., et al. Temporal Databases: Theory, Design, and Implementation. Bridge Parkway: benjamin/Cummings, 1993. p. 633.

[Touzovich 91] TAUZOVIK, B. Towards temporal extensions to the entity-relationship model. *Proceedings* of the tenth Int. Conf. on the Entity Relationship Approach, San Mateo, California, 1991.

[Tsotras 96] TSOTRAS, V.J.; KUMAR, A. Temporal databases bibliography update. ACM SIGMOD Record, v.25, n.1, p. 41-51, Mar. 1996.

[Wu 97] WU, Y.; JOJODIA, S.; WANG, X.S. Temporal Databases Bibliography Update. <http://isse.gmu.edu/~csis/tdb/bib97/bib97.html>.