

ROSIANE APARECIDA OLIVEIRA

**MODELAGEM E DESENVOLVIMENTO DE UM
BANCO DE DADOS TEMPORAL**

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

Prof^a. Olinda Nogueira Paes Cardoso

LAVRAS
MINAS GERAIS - BRASIL
2001

ROSIANE APARECIDA OLIVEIRA

**MODELAGEM E DESENVOLVIMENTO DE UM
BANCO DE DADOS TEMPORAL**

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 28 de junho de 2001.

Prof. André Luiz Zambalde

Prof. Heitor Augustus Xavier Costa

Profa. Olinda Nogueira Paes Cardoso
(Orientadora)

LAVRAS
MINAS GERAIS – BRASIL

DEDICATÓRIA

A Deus.

AGRADECIMENTOS

Agradeço a minha mãe
por todas as orações nos momentos críticos,
Agradeço aos meus colegas
pelo companheirismo durante todos estes anos,
Agradeço aos meus professores
que ajudaram-me a trilhar um caminho de sabedoria.

SUMÁRIO

| | | |
|-------|---|----|
| 1. | INTRODUÇÃO..... | 1 |
| 2. | PRINCIPAIS CONCEITOS DE REPRESENTAÇÃO TEMPORAL..... | 4 |
| 2.1 | Dimensão Temporal..... | 4 |
| 2.2 | Tempo de Transação e Tempo de Validade | 4 |
| 2.3 | Ordem no Tempo | 5 |
| 2.4 | Tempo Absoluto e Tempo Relativo..... | 5 |
| 2.5 | Variação Temporal | 6 |
| 2.6 | Granularidade Temporal..... | 6 |
| 2.7 | Instante no Tempo | 6 |
| 2.8 | Intervalo Temporal | 7 |
| 2.9 | Elemento Temporal | 7 |
| 2.10 | Duração Temporal | 8 |
| 2.11 | Representação Temporal Explícita e Implícita..... | 8 |
| 3. | MODELAGEM DE DADOS EM BANCO DE DADOS TEMPORAL... | 10 |
| 3.1 | Entidades transitórias..... | 15 |
| 3.2 | Entidades Perenes | 15 |
| 3.3 | O identificador interno de entidades - OID | 16 |
| 3.4 | Relacionamentos Temporais..... | 17 |
| 3.5 | Relacionamentos Intemporais..... | 17 |
| 3.6 | As perspectivas temporal e intemporal das entidades | 17 |
| 3.7 | Atributos | 18 |
| 3.7.1 | Atributos temporais | 19 |
| 3.7.2 | Atributos intemporais | 19 |
| 3.8 | Restrições de Cardinalidade | 20 |
| 3.8.1 | Relacionamentos intemporais..... | 20 |
| 3.8.2 | Relacionamentos temporais..... | 20 |
| 4. | SISTEMAS DE BANCO DE DADOS TEMPORAL..... | 28 |
| 4.1 | Características gerais a serem suportadas por SGBDT | 29 |
| 4.2 | Questões a serem examinadas no desenvolvimento dos Modelos de Dados Temporais | 31 |
| 4.3 | A nova geração de pesquisa em BDT..... | 33 |
| 4.4 | Consistência Temporal | 37 |

| | | |
|-----|---|----|
| 5. | IMPLEMENTAÇÃO DE BANCOS DE DADOS TEMPORAIS..... | 38 |
| 5.1 | Implementação usando o <i>Oracle 8i Time Series</i> | 38 |
| 5.2 | <i>Oracle 8i Time Series</i> e a Tecnologia Objeto-Relacional | 40 |
| 5.3 | Dados Históricos..... | 40 |
| 6. | CONSULTAS TEMPORAIS..... | 41 |
| 6.1 | Problemas no Processamento de Consultas Temporais..... | 41 |
| 7. | CONCLUSÃO..... | 43 |
| 8. | REFERÊNCIAS BIBLIOGRÁFICAS | 44 |
| | ANEXO A | 47 |
| | ANEXO B | 56 |

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1(a) – Banco de Dados contém apenas a nota atual | 11 |
| Figura 1(b) – Banco de Dados contém o histórico do aluno | 11 |
| Figura 1(c) – Banco de Dados com uso de atributos multivalorados e/ou compostos | 11 |
| Figura 2 – Classificação das entidades e relacionamentos no modelo TempER conforme a dimensão temporal..... | 14 |
| Figura 3 – Vínculos de um certo empregado emp1 com uma empresa ao longo do tempo | 15 |
| Figura 4 – Modelo no ER Convencional | 21 |
| Figura 5 – Modelo no ER Temporal – TempER | 23 |
| Figura 6 – TSESCOLA_TAB: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series..... | 57 |
| Figura 7 – TSESCOLA_MAP: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series..... | 57 |
| Figura 8 – TSESCOLA_CAL: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series..... | 58 |

LISTA DE TABELAS

| | |
|--|----|
| Tabela 1 – Dicionário de Dados para o modelo ER Convencional | 21 |
| Tabela 2 – Dicionário de Dados para o modelo TempER | 23 |
| Tabela 3 – Histórico Escolar | 39 |
| Tabela 4 – Códigos das frequências | 51 |
| Tabela 5 – Frequências e seus requerimentos | 52 |

RESUMO

Diversas aplicações de bancos de dados apresentam a necessidade de armazenar e recuperar informações temporais. Os bancos de dados comerciais não atendem tal objetivo. Um banco de dados temporal apresenta alguma forma implícita de representação de dados temporais. Tempo de transação e/ou tempo de validade fornecem mecanismos para que a evolução dos dados seja armazenada.

1. INTRODUÇÃO

O tempo está envolvido em todas as atividades do ser humano. Um banco de dados deve prover a facilidade para a modelagem de tempo em aplicações do mundo real. Sistemas de banco de dados conservam informações que representam o mundo real em um momento específico. O fato destas informações envolverem tempo e a necessidade de guardar tal evolução levaram ao desenvolvimento de Banco de Dados Temporal (BDT). Como o tempo passa, é necessário conservar todos os valores de dados definidos (passado, presente e futuro). Característica que se torna evidente em determinadas aplicações, tais como:

- registros de informações acadêmicas, nas quais devem ser armazenados todos os conceitos obtidos pelos alunos nos respectivos semestres;
- em áreas de contabilidade (datas de contas a pagar e receber, fluxo de caixa, pagamentos efetuados e recebidos), em tomadas de decisão (baseadas em informações históricas), em controle de estoques e em processos de importação/exportação;
- aplicações financeiras, como no mercado de ações, aplicações bancárias;
- companhias seguradoras, nos planos a serem oferecidos e onde os valores das apólices geralmente são baseados nas informações passadas dos clientes;
- sistemas de reservas (de companhias aéreas, de hotéis e etc);
- área médica, onde o registro das informações históricas de pacientes é fundamental.

Bancos de dados temporais permitem armazenar e recuperar todos os estados de uma aplicação (atual, passado e futuro previsto), registrando sua evolução com o passar do tempo [Edelweiss,1998].

Muitas pesquisas têm sido realizadas na área de bancos de dados temporais com o objetivo de definir conceitos e estratégias para tratar informações históricas. Os bancos de dados foram classificados em quatro categorias [Edelweiss,1998], dependendo da possibilidade de representação de informações históricas:

- *Bancos de Dados Instantâneos*: são armazenados somente os valores presentes e, ao alterar uma informação, a antiga é sobrescrita pela nova. Correspondem aos bancos de dados convencionais;
- *Bancos de Dados de Tempo de Transação*: associam aos dados apenas o tempo em que a informação foi inserida no banco de dados, recuperando somente o passado e o presente das informações;
- *Bancos de Dados de Tempo de Validade*: associam aos dados o tempo em que a informação será válida no banco de dados, possibilitando a recuperação tanto do passado, quanto do presente e do futuro das informações;
- *Bancos de Dados Bitemporais*: associam tempo de transação e tempo de validade aos dados, permitindo a recuperação de todos os estados das informações.

A noção de tempo, como datas, períodos, duração de validade de informações e intervalos temporais, surge em três diferentes níveis:

- i) Na modelagem de dados;
- i) Na linguagem de recuperação e manipulação de dados;
- ii) No nível de implementação do Sistema Gerenciador de Banco de Dados (SGBD).

O projeto objetiva discutir questões do uso banco de dados temporal em uma aplicação específica de banco de dados. Uma aplicação no meio acadêmico que justifica o uso de bancos de dados temporais é o Histórico Escolar. Visto

também que o objetivo primordial do projeto não é a criação da ferramenta e sim uma discussão embasada em torno de questões como:

- modelagem temporal dos dados;
- implementação;
- recuperação de dados temporais.

BDT tem sido uma área ativa de pesquisa nos últimos 20 anos. Durante este tempo, uma diversidade enorme de modelos de dados temporais e linguagens de consultas surgiram. A maioria das conferências de Banco de Dados inclui pelo menos um artigo em BDT. Há uma centena de pesquisadores ativos estudando BDT. A maioria das aplicações da área empresarial, engenharia, médica e científica necessitam armazenar dados históricos.

Apesar da substancial atividade de pesquisa e dos requisitos temporais das aplicações citadas, não há um SGBDT (Sistema Gerenciador de Banco de Dados Temporal) comercial. Há um embaraço na literatura de BDT: não há um consenso comum aceito em modelo de dados ou linguagens de consulta que servirão de base para projetos de BDT e otimização de consultas temporais. Em um nível mais fundamental, a terminologia de BDT é altamente não padronizada. Não existe uma infraestrutura comum para a pesquisa em BDT.

O capítulo 2 trata dos principais conceitos de representação temporal. O capítulo 3 trata da modelagem de dados em BDT. Sistemas de BDT é assunto do capítulo 4. O capítulo 5 trata da implementação de BDT. O capítulo 6 fala das consultas temporais e o capítulo 7 finaliza com as conclusões.

2. PRINCIPAIS CONCEITOS DE REPRESENTAÇÃO TEMPORAL

A forma de representação escolhida reflete-se em interpretações diferentes dos conceitos temporais. As definições completas dos conceitos apresentados pode ser encontradas em [Jensen,1994], num glossário consensual de termos relativos a BDT.

2.1 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.

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 salário de um funcionário, 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.

2.2 Tempo de Transação e Tempo de Validade

Três diferentes conceitos temporais podem ser identificados em aplicações de banco de dados [Snodgrass, 1985]:

- (i) *tempo de transação*: tempo no qual o fato é registrado no banco de dados;
- (ii) *tempo de validade*: tempo em que o valor é válido na realidade modelada;

(iii) *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 SGBD, 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.

2.3 Ordem no Tempo

A definição de uma ordem a ser seguida no tempo é fundamental quando utiliza alguma representação temporal. Há três opções de ordenação temporal:

- tempo linear: total ordenação entre quaisquer dois pontos no tempo. Este é o mais comum;
- tempo ramificado: permite a possibilidade de dois pontos diferentes serem sucessores (ramificação no futuro) ou antecessores (ramificação no passado) imediatos de um mesmo ponto. Para ambos a restrição linear é abandonada;
- tempo circular: utilizado para modelar eventos e processos recorrentes.

A maior parte dos modelos temporais se baseia no tempo linearmente ordenado. A ordenação total do tempo pode ser definida com mais precisão através da teoria dos conjuntos [Antunes, 1997].

2.4 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 morreu no dia 13/01/70.

Tempo Relativo tem sua validade relacionada à validade de outro fato, ou ao momento atual. Exemplo: a inflação não aumentou ontem.

2.5 Variação Temporal

A forma de variação temporal apresenta basicamente duas formas:

- tempo contínuo: supõe-se que o tempo é contínuo por natureza. Entretanto, sem grande perda de generalidade, o tempo pode ser considerado discreto.
- tempo discreto: esta 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.

2.6 Granularidade Temporal

A granularidade temporal de um sistema consiste na duração de um *chronon*. Entretanto, dependendo da aplicação considerada, 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 e operações disponíveis nos sistemas gerenciadores do banco de dados que implementam o modelo.

2.7 Instante no Tempo

O conceito de instante, representando um particular ponto no tempo, depende da forma de variação temporal considerada. Quando é considerado tempo contínuo, um instante é um ponto no tempo de duração infinitesimal. Neste caso os

instantes são isomórficos com os números reais. 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.

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

2.8 Intervalo Temporal

Um intervalo temporal é caracterizado pelo tempo decorrido entre dois instantes. Depende também da forma de representação temporal definida no modelo. Quando é considerado tempo contínuo, o intervalo consiste de infinitos instantes do tempo. Na variação discreta, um intervalo é representado por um conjunto finito de *chronons* consecutivos. Um intervalo é representado pelos dois instantes que o delimitam. Dependendo da pertinência ou não dos instantes limites ao intervalo este pode ser aberto, semi-aberto ou fechado. Quando um dos limites é representado pelo instante atual (*now*) temos a representação de um intervalo particular cujo tamanho varia com a passagem do tempo.

Quando considerados intervalos, a variação temporal é linear. Se um intervalo fechado for representado por $[I1, I2]$, uma das seguintes fórmulas deve ser verdadeira: $I1 < I2$ ou $I1 = I2$.

2.9 Elemento Temporal

Elemento temporal é uma união finita de intervalos de tempo. É fechado para as operações de união, interseção e complemento da teoria dos conjuntos, isto é, qualquer destas operações sobre um elemento temporal produz um novo

elemento temporal. Como estas operações encontram contrapartida nos operadores booleanos *or*, *and* e *not*, isto produz uma substancial simplificação na habilidade do usuário de expressar consultas temporais. Tendo em vista que todos os intervalos temporais são subconjuntos do eixo temporal T , um elemento temporal, também o é. Tanto um intervalo temporal como um instante temporal são elementos temporais.

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 intervalo. Outro aspecto importante é esta primitiva temporal possibilitar a representação da “reencarnação” de objetos com facilidade. Um exemplo é uma pessoa ser funcionário de uma empresa durante o intervalo [1995,1997], tendo saído da empresa em 1998 e sendo readmitida dois anos depois (2000). A validade da existência desta pessoa na empresa seria a união dos intervalos $[1995,1997] \cup [2000]$.

2.10 Duração Temporal

Durações temporais podem ser de dois tipos, dependendo do contexto em que são definidas: fixas ou variáveis. Uma **duração fixa** independe do contexto de sua definição. Exemplo: uma hora tem sempre a duração de 60 minutos, independente do contexto. Uma **duração variável** depende do contexto, um exemplo é a duração de um mês, que pode ser de 28, 29, 30 ou 31 dias.

2.11 Representação Temporal Explícita e Implícita

A definição de tempo pode ser feita de forma explícita, através da associação de um valor temporal a uma informação na forma de um rótulo temporal (*timestamping*), 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 atribuído a um atributo, o valor que corresponde à sua primitiva temporal. A representação implícita é feita através da manipulação de conhecimentos sobre a ocorrência de eventos ou do relacionamento de intervalos de tempo, como por exemplo, a aula de BDT ocorreu ontem.

3. MODELAGEM DE DADOS EM BANCO DE DADOS TEMPORAL

A representação de uma aplicação exige a sua modelagem conceitual. Um modelo de dados temporal adequado deve ser utilizado pelos seguintes motivos:

- o sistema a ser modelado 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 ser representadas;
- 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.

Diversos modelos de dados tradicionais foram estendidos para possibilitar a representação de aspectos temporais sem ter que definir novos modelos [Antunes, 1998].

Várias extensões à abordagem entidade-relacionamento (ER) original têm sido propostas com o objetivo de incorporar a possibilidade de modelar propriedades temporais. Entre as quais:

- TERM (*Temporal Entity-Relationship Model*) [Klopprogge, 1981],
- RAKE (*Relationships, Attributes, Keys, and Entities*) [Ferg, 1985],
- ERT (*Entity Relationship Time Model*) [Loucopoulos, 1991],
- TER (*Temporal Entity-Relationship Model*) [Touzovich, 1991] e
- TEER (*Temporal Enhanced Entity-Relationship Model*) [Elmasri, 1993] e a sua variante STEER [Elmasri, 1992].

Estas abordagens, entretanto, apresentam várias restrições. O TempER [Antunes, 1998] é um modelo de dados temporal que procura corrigir os problemas identificados nas demais abordagens.

Quando o modelo ER convencional é utilizado, a associação das entidades e relacionamentos com o tempo materializa-se através da inclusão de

atributos comuns que armazenam datas, horas ou qualquer outra referência temporal. Além disso, caso necessite registrar os diversos valores que um atributo pode apresentar ao longo do tempo (por exemplo, as notas de um aluno) existem duas alternativas possíveis: (i) especificar uma nova entidade composta deste atributo e mais os atributos de referência ao tempo; (ii) definir tal atributo como composto e multivalorado, como pode ser observado na Figura 1.

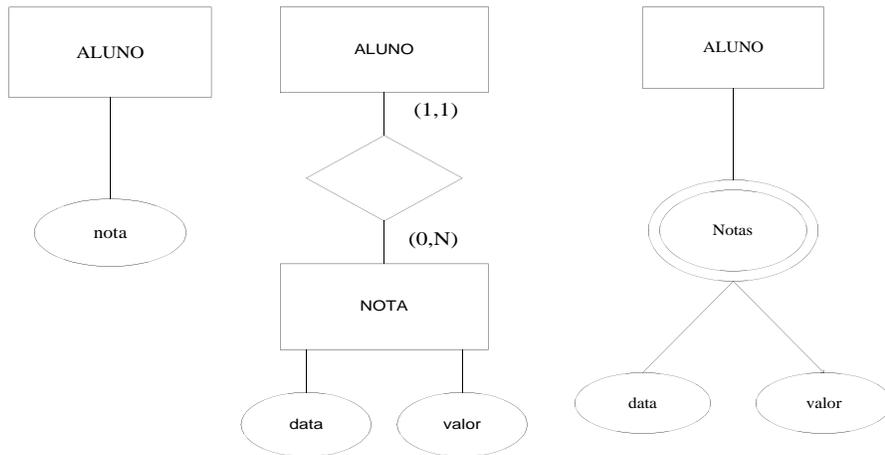


Fig.1 (a) -Banco de dados contém apenas a nota atual

Fig.1 (b) - Banco de dados contém o histórico do aluno

Fig.1 (c) – Banco de dados com uso de atributos multivalorados e/ou compostos.

Em um modelo de dados convencional, os conjuntos de entidades e relacionamentos apresentam duas dimensões: a primeira refere-se às instâncias (linhas) e a segunda aos atributos (colunas). Em um modelo ER temporal (ERT), uma nova dimensão é acrescentada: a dimensão temporal. A forma que tem se mostrado mais adequada de tratar a dimensão temporal em sistemas de informação é assumi-la como uma seqüência discreta, linear e finita de pontos consecutivos no tempo. A esta seqüência de pontos do tempo dá-se o nome de **eixo temporal**.

Um outro requisito importante a ser preenchido por um modelo de dados que incorpore a dimensão temporal é permitir que em um mesmo diagrama seja possível associar objetos (entidades, relacionamentos ou atributos) temporalizados com objetos não temporalizados. Isto se faz necessário porque em sistemas de informação, normalmente, alguns dados precisam ser explicitamente referenciados em relação ao tempo (a evolução das notas dos alunos, a alocação de alunos em projetos,...), e outros não apresentam esta necessidade, ou porque não mudam com o tempo, ou porque é irrelevante ao usuário saber quando os fatos ocorreram (o código de uma disciplina, a autoria de um artigo,...). Normalmente os objetos não temporalizados são assumidos como existindo sempre, ou seja, adquirem uma validade temporal, implícita e constante, igual a todo o conjunto de pontos do eixo temporal.

O modelo ERT permite mesclar em um mesmo diagrama objetos temporalizados e objetos não temporalizados. Contudo, apresenta alguns aspectos que dificultam o processo de modelagem:

- os atributos são explicitados graficamente, resultando em diagramas bastante carregados. Esta característica contraria um dos principais objetivos de um modelo gráfico que é apresentar facilidade de uso e visualização;
- o modelo ERT utiliza como primitiva temporal o **intervalo de tempo** que mostra-se inferior à primitiva conhecida como "elemento temporal". Quando se faz uso dos intervalos de tempos como rótulos temporais dos objetos, há a necessidade de fragmentar em diversas tuplas os objetos "reencarnados", uma tupla para cada intervalo de existência.

O modelo de dados TempER surge com a intenção de suprimir os pontos desfavoráveis identificados nas propostas de modelagem temporal pesquisadas na literatura.

O modelo TempER é um modelo de dados do tipo Entidade-Relacionamento que incorpora dispositivos que permitem referenciar os objetos

(entidades, relacionamentos ou valores de atributos) à dimensão temporal. A dimensão temporal é assumida como sendo um eixo de pontos discretos, isomórfico ao conjunto dos números inteiros.

O modelo TempER permite representar a associação entre elementos temporalizados e não temporalizados. Para tanto, adota o pressuposto que todas as entidades, sejam elas temporalizadas ou não, apresentam uma dimensão temporal, ou seja, uma "existência" ou validade temporal. No caso das entidades temporalizadas, esta existência é um subconjunto de pontos do eixo temporal. Em virtude disto são chamadas de entidades **transitórias**. Em relação às entidades não temporalizadas, é assumido que "existem" durante todo o eixo temporal, ou seja, a sua validade temporal é constante, implícita e igual a todo o eixo temporal. Por isto, são denominadas entidades **perenes**.

Qualquer que seja a classificação de uma entidade em relação ao tempo, seja transitória ou perene, ela sempre vai apresentar duas perspectivas: uma intemporal e uma temporal. No tocante aos relacionamentos, ou as entidades se associam entre si na perspectiva temporal (relacionamentos temporais) ou na perspectiva intemporal (relacionamentos intemporais).

No modelo TempER :

- a primitiva temporal utilizada é o elemento temporal e não o intervalo de tempo. O elemento temporal servirá de rótulo do tempo para as entidades, relacionamentos e atributos do modelo de dados. O elemento temporal se mostra superior ao uso da primitiva intervalo temporal;
- a “reencarnação” de objetos é implementada com facilidade. Um aluno que trancou matrícula pode ser readmitido, um aluno que tenha sido alocado a um determinado curso pode vir a ser realocado a este mesmo curso em um outro período de tempo;

- os atributos não são explicitados graficamente, mas através de um dicionário de dados associado ao diagrama ER, o que resulta em um modelo mais administrável visualmente.

No modelo TempER, existe apenas um eixo temporal e este representa o **tempo de validade**. Não é necessário referir-se ao tempo de transação porque os modelos produzidos estão no estágio conceitual, ou seja, não devem especificar aspectos inerentes à implementação física. Não se deve confundir existência com a presença física da entidade no banco de dados. **Existência** de uma entidade é o conjunto de pontos do tempo em que esta entidade é considerada como válida para efeito de inferências temporais no banco de dados [Antunes,1998].

Assim como o modelo ER convencional, o modelo TempER apresenta também os seguintes elementos básicos: entidade, relacionamento e atributos.

Aos objetos de uma aplicação que forem especificados como temporalizados, sejam eles entidades, relacionamentos ou valoração de atributos, são implicitamente anexados rótulos temporais (*timestamps*), que conterão o conjunto de pontos do tempo nos quais estes objetos são considerados como existentes no contexto da realidade modelada, como visto na Figura 2.

Notação

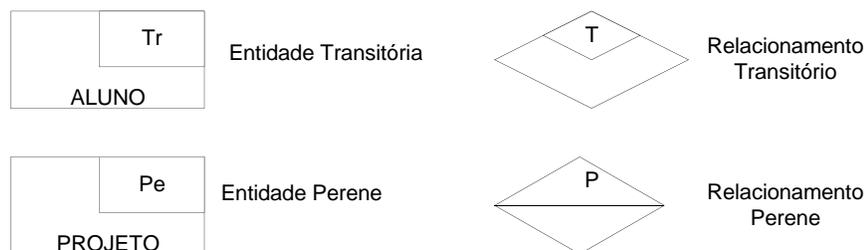


Figura 2 - As entidades e os relacionamentos no modelo TempER são classificados em transitórios e perenes, conforme a sua relação com a dimensão temporal.

3.1 Entidades transitórias

São aquelas cuja validade temporal é um subconjunto de pontos do tempo do eixo temporal. Utiliza-se este tipo de entidade quando se quer modelar entidades que valem por um certo período de tempo. Por exemplo: os empregados de uma companhia que existem como tal desde a sua admissão até a sua demissão. Ver Figura 3. Como é possível haver readmissão de empregados, a existência de uma destas entidades pode ser composta por mais de um intervalo de tempo, portanto, o subconjunto de pontos do tempo que define a validade temporal de uma entidade não necessariamente é formado por momentos consecutivos.



Figura 3 - Um certo empregado *emp1* apresenta três vínculos com a empresa ao longo do tempo.

É possível através das transações do sistema ampliar ou reduzir a validade temporal das entidades transitórias. Como resultado do funcionamento do sistema, subconjuntos de pontos do tempo podem ser adicionados ou retirados da existência deste tipo de entidade. É dessa possibilidade de modificar a validade temporal que advém o qualificativo "transitórias" aplicado a estas entidades.

3.2 Entidades Perenes

São aquelas cuja validade temporal é exatamente igual a todo o eixo temporal. Toda vez que uma entidade perene é incluída no banco de dados do sistema, assume-se que seu rótulo temporal é igual a "[« , »]", isto é, a sua validade temporal inicia no primeiro ponto do eixo temporal e se estende até o último.

Normalmente as entidades que o modelador não necessita ou não deseja associar ao tempo são consideradas como perenes.

O fato de ser perene não significa que uma entidade não possa ser eliminada do banco de dados. Entretanto, enquanto uma entidade perene estiver presente no banco de dados, a sua validade temporal será constante, igual ao conjunto de todos os pontos do eixo temporal. Portanto, diferentemente das entidades transitórias, a validade temporal das entidades perenes de forma alguma pode sofrer acréscimo ou redução.

Por ser constante, a validade temporal das entidades perenes não precisa ser registrada no banco de dados. O modelador deve visualizar o conjunto de pontos de tempo que define a existência das entidades perenes como sendo implicitamente especificado, ou seja sempre igual ao conjunto total de pontos do eixo temporal.

3.3 O identificador interno de entidades - OID

É assumido que todas as instâncias das entidades possuem um identificador interno, gerado pelo sistema, que é denominado de OID (*object identifier*). Cada OID, por princípio, é único no âmbito do universo do discurso da aplicação, é invisível ao usuário e define a identidade de uma instância de uma entidade. Quando duas instâncias de entidades se associam, este fato pode ser representado pelo relacionamento dos OID's respectivos. Isto faz com que um relacionamento seja identificável pela composição dos OID's das instâncias das entidades associadas.

A presença do OID não descarta a necessidade de que haja um atributo (ou composição de atributos) que desempenhe o papel de chave primária de uma entidade, de forma que um usuário do sistema possa identificar e acessar esta entidade, já que o OID é invisível externamente. Embora possa parecer redundante coexistirem OID's e chaves primárias, o que se busca é aproximar o

modelo TempER de um dos mais importantes princípios da orientação a objetos, que é a identidade dos objetos [Khoshafian, 1986]. Além disso, existe a vantagem de ser possível alterar a chave primária de uma entidade sem que isto afete os relacionamentos em que ela participe.

3.4 Relacionamentos Temporais

São os relacionamentos que associam duas entidades no âmbito da dimensão temporal, isto é, além dos OID's das instâncias das entidades, estes relacionamentos também se associam a pontos do tempo do eixo temporal. Este tipo de relacionamento serve para modelar as associações das quais se necessita conhecer a validade temporal. Os relacionamentos temporais só são válidos nos momentos especificados pelos seus rótulos temporais. A validade no tempo de um relacionamento temporal sempre está contida dentro da interseção das existências das entidades associadas. Ou seja, o conjunto de pontos do tempo que define a validade de um relacionamento do tipo temporal é um subconjunto da interseção dos conjuntos de pontos que definem as existências das entidades associadas.

3.5 Relacionamentos Intemporais

São relacionamentos que não levam em consideração a dimensão temporal, ou seja, neste tipo de relacionamento apenas participam os OID's das entidades, sem qualquer menção a pontos do tempo. Os relacionamentos intemporais se materializam no nível das perspectivas intemporais das entidades. Por exemplo, uma relação que não tem momento inicial ou final de validade é intemporal.

3.6 As perspectivas temporal e intemporal das entidades

As entidades são os elementos básicos de um modelo de dados que emprega a abordagem Entidade-Relacionamento. No modelo TempER, em virtude da

dimensão temporal, as entidades apresentam sempre duas perspectivas: uma perspectiva temporal e uma perspectiva intemporal.

- **Perspectiva intemporal:** não leva em consideração a dimensão temporal, isto é, o OID da entidade não é visualizado como associado a um conjunto de pontos do tempo. Nesta perspectiva as entidades apresentam apenas duas dimensões: atributos e valores;
- **Perspectiva temporal:** leva em conta a dimensão temporal ao referenciar um dado. Nesta perspectiva, o OID de uma entidade é visualizado como estando associado a um conjunto de pontos do tempo, conjunto este que define a validade temporal (ou existência) da entidade no contexto do banco de dados do sistema. As entidades, vistas pela perspectiva temporal, apresentam três dimensões: atributos, valores e tempo.

3.7 Atributos

Tanto as entidades como os relacionamentos apresentam propriedades que os caracterizam, as quais são denominadas de atributos. Eles são especificados através do dicionário de dados, que complementa um diagrama TempER. A descrição de um atributo apresenta os seguintes elementos: o nome do atributo, o domínio dos valores primitivos que podem ser associados ao atributo e a classificação do atributo em relação ao tempo (indicando se o atributo é temporal ou intemporal).

Uma entidade normalmente possui um identificador de uso externo, uma chave primária. Em geral trata-se de um único atributo, no entanto pode resultar da composição de dois ou mais atributos. O valor de um identificador é único dentro do contexto de uma entidade, isto é, não existem duas instâncias com o mesmo valor para o identificador. O identificador deve sempre ser formado por atributo(s) intemporal(is).

As entidades, sejam elas transitórias ou perenes, por apresentarem duas perspectivas em relação ao tempo, podem combinar atributos temporais com atributos intemporais. Isto não acontece com os relacionamentos. Se forem temporais possuem apenas atributos temporais, se forem intemporais apresentam apenas atributos intemporais.

3.7.1 Atributos temporais

São aqueles cuja valoração (conteúdo) deve ser referenciada a pontos do tempo. O atributo “nota” do relacionamento CURSA (Tabela 2) é um exemplo de atributo temporal. É importante salientar que, primeiramente não há nada impedindo que um mesmo valor apareça em pontos diferentes do tempo, em segundo lugar a série de valores que um atributo temporal apresenta ao longo do tempo não deve ser confundida com multivaloração de atributo, pois a cada momento do tempo apenas um, e somente um, valor é permitido.

O objetivo dos atributos temporais é permitir que registre a evolução dos valores de uma propriedade ao longo do tempo.

3.7.2 Atributos intemporais

São aqueles cujos valores não apresentam qualquer associação com a dimensão temporal. Através dos atributos intemporais é possível representar os atributos convencionais do modelo Entidade-Relacionamento. Exemplos deste tipo de atributo são: “mat” e “sexo” da entidade ALUNO (Tabela 1).

Os atributos intemporais existem apenas na perspectiva intemporal das entidades ou associados a relacionamentos intemporais.

3.8 Restrições de Cardinalidade

A restrição de cardinalidade especifica o número de instâncias de um tipo de relacionamento que uma entidade pode participar. Refere-se à participação das entidades nos conjuntos-relacionamento [Elmasri, 1994].

3.8.1 Relacionamentos intemporais

Os relacionamentos intemporais apresentam as seguintes restrições de cardinalidade:

- (0,1) a entidade pode não participar em relacionamentos; se participa, limita-se a no máximo uma vez;
- (0,N) a entidade pode não participar em relacionamento, ou pode participar mais de uma vez;
- (1,1) a entidade participa uma única vez do relacionamento;
- (1,N) a entidade participa no mínimo uma vez em relacionamento, podendo participar mais vezes.

3.8.2 Relacionamentos temporais

Os relacionamentos temporais apresentam as seguintes restrições de cardinalidade:

- (0,1) a entidade pode não participar em relacionamentos; se participa, limita-se a no máximo uma vez a cada momento do tempo;
- (0,N) a entidade pode não participar em relacionamento, ou pode participar mais de uma vez a cada momento do tempo;
- (1,1) a entidade participa uma única vez do relacionamento, a cada momento do tempo;
- (1,N) a entidade participa no mínimo uma vez em relacionamento, podendo participar mais vezes a cada momento do tempo.

Na proposta de notação dos elementos gráficos do modelo TempER buscou-se manter os símbolos originais propostos no modelo ER convencional, adicionando-lhes apenas os sinais necessários para determinar qual é o tipo de relação destes elementos com a dimensão temporal, como observado na Figura5.

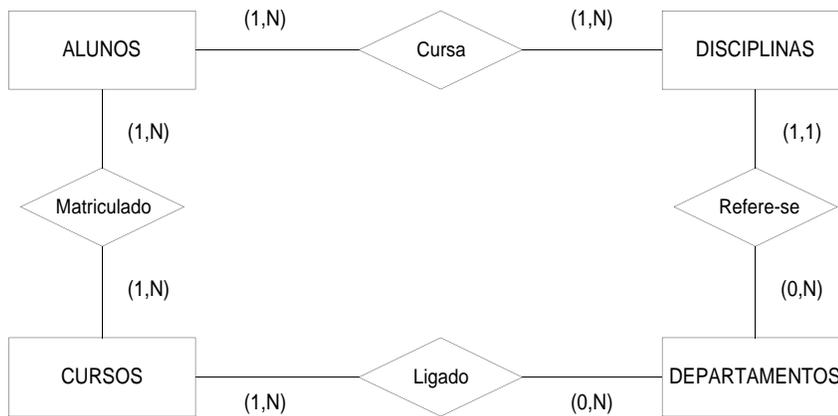


Figura 4 - Modelo no ER Convencional

Contrariando a idéia do modelo ER Convencional, onde diz que os atributos devem ser explicitados graficamente, mostra-se abaixo um dicionário de dados para o respectivo modelo, afim de facilitar a sua visualização.

Tabela 1 - Dicionário de Dados para o modelo ER Convencional

| entidade ALUNOS | entidade CURSOS | entidade DISCIPLINAS |
|------------------------|---------------------------|-----------------------------|
| atributos: | atributos: | atributos: |
| <u>mat</u> : STRING; | codCurso: STRING; | codDisc: STRING; |
| nome: STRING; | nome: STRING; | nome: STRING; |
| endereço: STRING; | sigla: STRING; | sigla: STRING; |
| telefone: STRING; | situação: STRING; | situação: STRING; |
| sexo: CHAR; | duração: STRING; | programa: MEMO; |
| dataNasc: DATE; | condiçãoLegal: STRING; | cargaHora: STRING; |
| filiação: STRING; | coordenador: STRING; | numAlunos: NUMERAL; |
| estadoCivil: CHAR; | dataCriação: DATE | numAlunosideal: NUMERAL; |
| cpf: STRING; | | profResponsável: STRING; |
| rg: STRING; | identificador: (codCurso) | semestre: STRING |
| foto: GRAFIC; | | |
| situação: STRING; | | identificador: (codDisc) |

| | | |
|--|--|--|
| modulo: STRING turma: STRING; numChamada: NATURAL; localSegGgrau: STRING; anoSegGrau: STRING; endSegGrau: STRING; ufSegGrau: STRING; formAdmissao: STRING; sitMat: STRING; expediçãoDiploma: DATE; colação: DATE; dataMatricula: DATE; semestre: STRING; dataDesligamento: DATE identificador: (mat) | | |
| entidade DEPARTAMENTOS atributos: codDep: STRING; sigla: STRING; ramal: STRING; localidade: STRING; chefe: STRING; dataCriação: DATE identificador: (CodDep) | relacionamento CURSA atributos: semestre: STRING; nota: REAL; situação: STRING; | Os demais relacionamentos: Matriculado; Ligado e Refere- se não possuem atributos. |

Tabela 1 (continuação) - Dicionário de Dados para o modelo ER Convencional

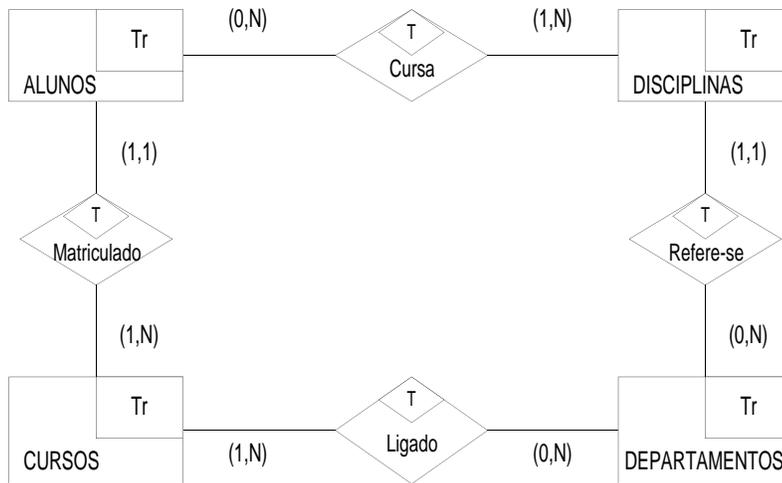


Figura 5 – Modelo no ER temporal (TempER)

Tabela 2 - Dicionário de Dados para o modelo TempER

| entidade ALUNOS | entidade CURSOS | entidade DISCIPLINAS |
|----------------------------------|------------------------------------|------------------------------------|
| atributos: | atributos: | atributos: |
| <u>mat</u> : STRING, Intemporal; | codCurso: STRING, Intemporal; | codDisc: STRING, Intemporal; |
| nome: STRING, Intemporal; | nome: STRING, Temporal; | nome: STRING, Temporal; |
| endereço: STRING, Intemporal; | sigla: STRING, Intemporal; | sigla: STRING, Temporal; |
| telefone: STRING, Intemporal; | situação: STRING, Intemporal; | situação: STRING, Temporal; |
| sexo: CHAR, Intemporal; | duração: STRING, Intemporal; | programa: MEMO, Temporal; |
| dataNasc: DATE, Intemporal; | condiçãoLegal: STRING, Intemporal; | cargaHora: NATURAL, Temporal; |
| filiação: STRING, Intemporal; | coordenador: STRING, Intemporal; | numAlunos: NUMERAL, Temporal; |
| estadoCivil: CHAR, Intemporal; | identificador: (codCurso) | numAlunosideal: NUMERAL, Temporal; |
| cpf: STRING, Intemporal; | | profResponsável: STRING, Temporal; |
| rg: STRING, Intemporal; | | identificador: (codDisc) |
| foto: GRAFIC, Intemporal; | | |
| situação: STRING, Temporal; | | |
| modulo: STRING, | | |

| | | |
|---|---|--|
| <p>turma: Intemporal; STRING, Intemporal; numChamada: NATURAL, Intemporal, localSegGgrau: STRING, Intemporal; anoSegGrau: STRING, Intemporal; endSegGrau: STRING, Intemporal; ufSegGrau: STRING, Intemporal; formAdmissao: STRING, Intemporal; sitMat: STRING, Intemporal; expediçãoDiploma: DATE, Intemporal; colação: DATE, Intemporal; dataMatricula: DATE, Intemporal; semestre: STRING, Intemporal; dataDesligamento: DATE, Intemporal;</p> <p>identificador:(mat)</p> | | |
| <p>entidade DEPARTAMENTOS atributos: codDep: STRING, Intemporal; sigla; STRING, Temporal; ramal: STRING, Intemporal; localidade: STRING, Intemporal; chefe: STRING, Temporal; identificador: (CodDep)</p> | <p>relacionamento CURSA atributos: nota: REAL, Intemporal; situação: STRING, Intemporal</p> | <p>Os demais relacionamentos: Matriculado, Ligado e Refere-se não possuem atributos.</p> |

Tabela 2 (continuação) - Dicionário de Dados para o modelo TempER

A definição de um atributo como ‘temporal’ ou ‘intemporal’ leva em consideração o escopo da aplicação em questão. Por exemplo, o atributo “Nome” do aluno pode ser definido como temporal, pois seu nome pode mudar quando se casa, ou se divorcia ou por outros motivos. Mas, o armazenamento de todos os nomes e sua variação temporal não interessa à aplicação de Histórico Escolar. Mas, já em uma aplicação de Cartório, esta história do “nome” de uma pessoa deveria ser armazenada. Na aplicação de Histórico Escolar somente interessa para a instituição, o nome no instante de quando foi efetuada a matrícula, definindo-o, como intemporal.

Suponha que o modelo apresente a seguinte restrição: um aluno obrigatoriamente deve estar matriculado em um curso em cada momento da sua existência como aluno, não podendo estar matriculado em mais de um curso dentro da própria instituição ao mesmo tempo. Um outro requisito é a necessidade de representar as possíveis matrículas que um aluno pode apresentar ao longo do tempo, em função de sua transferência de um curso para outro.

No ER convencional não é possível especificar a restrição que determina que um aluno não pode estar matriculado em mais de um curso em cada momento do tempo. A cardinalidade "(1, N)", que aparece na ligação entre ALUNO e CURSO, especifica que um aluno deve estar associado a no mínimo um curso, podendo estar associado a mais de um (considerando toda carreira acadêmica na qual este aluno pode ser transferido de curso) .

A responsabilidade de tratar as restrições temporais no ER é transferida para a modelagem dinâmica, ou seja, no caso do exemplo em questão, são as transações do sistema que devem se preocupar em impedir que um aluno possa estar matriculado em dois ou mais cursos ao mesmo tempo, no caso da mesma faculdade.

Este problema deixa de existir quando se utiliza o modelo TempER. A cardinalidade que aparece na ligação entre ALUNO e CURSO agora é "(1, 1)", a

qual tem o seguinte significado: um aluno participa do relacionamento “está matriculado” no mínimo uma vez e no máximo uma vez a cada momento do tempo. Além disto, os atributos referindo-se a pontos do tempo que estavam no diagrama ER convencional deixam de ser necessários, pois são substituídos por rótulos temporais implícitos.

A vantagem de utilizar um modelo de dados temporal, no lugar de um convencional, está na sua capacidade de expressar a associação dos elementos com o tempo e de especificar as restrições decorrentes disto. Nos modelos de dados convencionais o tempo é representado por atributos comuns (datas, horas, etc.) e não estão disponíveis mecanismos para representar restrições temporais, exigindo, por consequência, que o modelador as especifique ao nível do modelo dinâmico.

Pesquisadores não poupam esforços no desenvolvimento de modelos de dados temporais e linguagens de consultas. Nos últimos 20 anos de trabalho, cerca de 24 extensões temporais do modelo relacional foram propostas. Aproximadamente metade destes modelos suportam unicamente tempo de validade, 10% suportam unicamente tempo de transação e o restante relações bitemporais.

Os modelos de dados temporais podem ser comparados [Snodgrass,1994] pelas quatro questões:

- como o tempo de validade é representado (alternativas incluem, intervalo ou elemento temporal de atributos individuais ou tuplas);
- como o tempo de transação é representado (alternativas incluem evento, elemento temporal de atributos individuais ou tuplas, ou conjunto de tuplas);
- como os valores dos atributos são representados (valores atômicos, pares ordenados, valores triplos ou conjunto triplo de valores);
- o modelo é homogêneo (permite agrupamento).

A maioria dos modelos de dados temporais tem associado uma ou mais linguagens de consulta temporal. Cerca de 24 linguagens de consulta relacional temporal foram propostas, baseadas na álgebra relacional, extensões Quel, extensões do SQL e baseadas em outros formalismos.

Suporte a tempo em sistemas de BD convencionais é feito totalmente no nível de tempo definido pelo usuário. Estas implementações são limitadas no escopo e, em geral, não metódicas em seus projetos.

Muitos da comunidade de pesquisa de BDT percebem que o tempo veio a consolidar-se próximo ao modelo de dados temporal e linguagem de consulta que são baseados em álgebra e cálculo. Com uma ampla diversidade de linguagens e modelos de dados, temas comuns continuam emergindo. Entretanto, a comunidade está muito fragmentada, com cada projeto de pesquisa estando baseado em um particular e diferente conjunto de suposições e abordagens. Frequentemente estas suposições não são originárias da pesquisa em si, mas são colocadas simplesmente porque a pesquisa requer um modelo de dados ou linguagem de consulta com certas características que aparecem em uma particular, escolhida arbitrariamente. Por exemplo, pesquisa em otimização de consulta deve assumir alguns modelos de dados e algumas linguagens de consulta, mas os detalhes são frequentemente não críticos. Seria melhor em tais circunstâncias para projetos de pesquisa escolherem a mesma linguagem. Infelizmente, nenhuma linguagem existente tem atraído seguidores suficientes para tornar-se uma escolha única.

É difícil chegar a um único modelo de dados quando há desacordos fundamentais referindo-se em que extensão o tempo deveria ser incorporado ao modelo. Determinar a funcionalidade que é desejada em uma linguagem de consulta temporal é uma importante questão. A pesquisa em BDT está ativa, com novas idéias aparecendo tempo todo. É importante fazer o projeto de modo correto.

4. SISTEMAS DE BANCO DE DADOS TEMPORAL

Há três tipos de tempos que podem ser usados em um BDT [Snodgrass, 1994]:

- 1) Tempo definido pelo usuário: tem suporte garantido na maioria dos SGBDs comerciais (e está presente no SQL padrão);
- 2) Tempo de Validade;
- 3) Tempo de Transação: é suportado em alguns SGBDs Objeto-Relacional (Oracle 8) e em um relacional (Montage).

Entretanto, as aplicações que poderiam suportar tempo de validade (maioria das aplicações), bem como as que poderiam suportar todos três tipos de tempo (não é a maioria, mas certamente uma porção considerável) impõe que uma extensão bitemporal de SQL seja garantida.

Suporte temporal deveria ser opcional em ambos os esquemas e na linguagem de consulta. Esta necessidade trata da habilidade para especificar relações instantâneas (nenhum suporte temporal é requerido), bem como relações de tempo de validade, tempo de transação na expressão CREATE TABLE. Consultas deveriam incluir múltiplos tipos de relações na cláusula FROM e evoluir para múltiplos tipos de relações. Por exemplo, deveria ser possível computar via SELECT uma relação instantânea de uma relação bitemporal. A extensão do SQL deveria ser compatível com a TSQL (*Temporal Structured Query Language*). Consultas existentes em SQL deveriam permanecer válidas em TSQL.

A representação do tempo (como oposto ao modelo conceitual) deveria ser discreta. É fácil perceber fazendo uma analogia aos números reais e suas representações. Enquanto números de ponto flutuante em programas computacionais podem ser conceitualizados pelos programadores como real (contínuo), sua representação deve ser necessariamente discreta. O mesmo deveria acontecer nos *timestamps* do modelo de dados da TSQL.

Há uma discussão a respeito da distinção entre intervalos abertos e fechados. Um intervalo semi-aberto, denotado por $[a,b)$, onde a e b são os *timestamps*, contém o tempo entre a e b , bem como o instante a , mas não o instante b . O intervalo fechado $[a,b]$ contém o instante b . Em uma representação discreta $[a, b+1] \equiv [a,b]$, em modelo de tempo contínuo, não há sucessor para b , portanto, os dois não são comparáveis. No nível de representação do TSQL, em que usa tempo discreto, a distinção não é importante. No nível de linguagem, em que o usuário visualiza é baseado em tempos contínuos, a linguagem deveria suportar ambos intervalos aberto e fechado na apresentação (entrada e saída) dos valores temporais.

Enquanto a maioria das aplicações necessita armazenar a variação temporal dos dados, não há BDT comerciais usados amplamente. Uma razão para a falta de transferência tecnológica da pesquisa para a prática é a necessidade de consenso comum aceito de modelo de dados ou linguagens de consultas que são a base da pesquisa e desenvolvimento. A terminologia é inconsistente.

4.1 Características gerais a serem suportadas por SGBDT

Uma importante consideração trata da questão de dados temporais em BDT. O suporte a dados definidos como temporal em SGBDT. Este suporte deveria ser análogo ao que ocorre nos BD convencionais com os atributos já predefinidos. Se um atributo é definido como inteiro, não é possível inserir caracteres. Analogamente, há a necessidade de testes de consistência para justificar a definição de um atributo como temporal. Por exemplo, suponha-se que o nome de uma função determina o salário. O SGBDT deve garantir que o mesmo nome não esteja associado com dois salários diferentes ao mesmo tempo. Se o atributo temporal é tratado simplesmente como outro atributo qualquer, parece não ser necessário defini-lo especialmente como temporal.

Outra questão é o suporte a valores de tempo de muitas granularidades diferentes. Operações apropriadas devem ser suportadas para realizar conversões de valores tempo entre as diferentes granularidades e para formular consultas e apresentar os resultados em uma forma apropriada. Há a necessidade de uma operação de *merge* para tornar o trabalho com dados vindos de diferentes BD ou relações definidas em diferentes níveis de granularidade. Em BD científicos, BD de planejamento e *scheduling* é essencial o suporte não apenas a tempos baseados na linha do tempo (tempo absoluto), mas também a tempos relativos.

Várias características relacionadas a consultas são também identificadas:

- suporte a tempos relativos (atributos ordenados temporalmente);
- suporte aos operadores de agregação sobre tempo;
- suporte a junção temporal.

Devido a muitos usuários em potencial de BDT e às necessidades diversas, a funcionalidade central deve ser minimizada e a facilidade de extensibilidade deve ser maximizada. Relações temporais apropriadas (possivelmente imprecisas) tem sido definidas e suportadas em SGBDT (relações ponto-a-ponto, intervalo-ponto, intervalo-intervalo). Entretanto, suporte a um armazenamento único deve ser fornecido: razão que as vezes é considerada fora do escopo de BDT.

Focalizando em **dados semânticos** (significado dos dados armazenados no modelo de dados), ou em **dados de apresentação** (como dados temporais são mostrados ao usuário), ou em **dados armazenados** (como a estrutura de armazenamento convencional pode ser empregada com dados temporais), ou em **uma eficiente linguagem de consulta**, complicou a tarefa primordial de capturar a semântica de variação do tempo. O resultado é uma diversidade de modelos de dados e linguagens de consultas incompatíveis e uma correspondente descrença no projeto de BDT e implementação de estratégias que poderiam ser empregadas fazendo-se uso destes modelos temporais. A

maioria dos modelos propostos apontam como sendo adequados para dados de apresentação, dados armazenados e para capturar a semântica temporal dos dados. Ao vê-los unicamente com o objetivo de capturar a semântica temporal, tais modelos exibem anomalias de apresentação e representação porque colocam a semântica temporal de uma maneira mais complicada que a necessária. A semântica de variação temporal é obscurecida no esquema de representação por outras considerações de apresentação e implementação.

Adotar um simples modelo conceitual de dados que capture a essência da semântica de variação temporal das relações, mas que não tenha ilusões quanto uma existência adequada para apresentação, armazenamento, ou evolução da linguagem de consulta, é a tarefa mais importante. Um modelo conceitual adequado deve permitir equivalências a serem demonstradas com muitos dos modelos de representação. Esta equivalência deve estar baseada no que acontece com os BD convencionais, onde se diz que duas relações são equivalentes se todos seus instantes que pegam todos os tempos (validade e transação) são idênticos.

4.2 Questões a serem examinadas no desenvolvimento dos Modelos de Dados Temporais

- Homogeneidade: em alguns modelos de dados, todos os atributos de uma tupla são definidos sobre o mesmo intervalo de tempo. Outros permitem atributos a serem definidos sobre diferentes intervalos de tempo, em parte para permitir produto cartesiano;
- Agrupamento: alguns modelos requerem que tuplas de valor equivalente (aquelas com idênticos valores de atributos explícitos), sejam agrupadas se elas estão sobrepostas no tempo, que é, combinadas em uma tupla. Outros modelos não permitem o agrupamento de tuplas de valor equivalente;

- Atributo-valor x rótulo-tupla: a comunidade de pesquisa está separada neste aspecto, com metade dos modelos de dados adotando a primeira abordagem que é vista por alguns como sendo próxima de orientação-objeto e a outra metade adotando a segunda abordagem que vista por alguns como sendo mais eficiente;
- Níveis de alinhamento: alguns modelos de atributo-valor podem ser vistos como uma extensão dos modelos relacionais alinhados. Alguém pode perguntar se o alinhamento deve ser restrito a um nível ou se múltiplos níveis de alinhamento devem ser permitidos;
- *Vacuuming*: quando tempo de transação é suportado, todas as alterações, incluindo remoção lógica, são implementadas como inserções físicas, levando a um crescimento do BD. No modelo relacional padrão, uma notação adicional de remoção é necessária em ordem para controlar o tamanho e o conteúdo do BD. *Vacuuming* (gerar vácuo) é esta notação e denota a flexibilidade, a remoção física, consistente com a semântica do tempo de transação dos dados em uma BDT que suporta tempo de transação. A eliminação física de dados temporais não relevantes é permitida, para os quais o custo de armazenamento é significativo;
- Tempo periódico: há modelos que suportam eventos periódicos (exemplo: um avião que voa todo dia um certo tempo);
- Tempo “manufaturado”: alguns defendem a semântica de interseção, onde o período de validade de uma tupla resultante do produto cartesiano é a interseção do período de validade de duas tuplas básicas e caracteriza qualquer tempo de saída como manufaturado e, em alguns casos, artificial.

Outros consideram que há aplicações para outras semânticas tal como união, sendo totalmente natural.

O modelo de dados temporal básico e a linguagem devem ser projetados unicamente em termos de suas propriedades semânticas, com possivelmente múltiplos modelos de dados distintos sendo empregados para representação e apresentação.

4.3 A nova geração de pesquisa em BDT

BDT podem ser usados em grande variedade de aplicações. Em adição às aplicações convencionais, outras como logística, aplicações científicas, sistemas de informações geográficas (GIS) e de IA (Sistemas Baseados em Conhecimento), apresentam novos requerimentos da pesquisa de BDT.

As duas características não encontradas nos SGBDs atuais propostos para BDT, características adicionais que um SGBDT de suportar [Snodgrass, 1994] são:

1ª) Múltiplas linhas de tempo

Tradicionalmente uma única linha de tempo é associada a uma dada entidade. Entretanto, há casos em que várias linhas de tempo parecem ser necessárias. Por exemplo: um cheque com uma data de pagamento associada. Há pelo menos três datas associadas com este cheque: a data em que este foi emitido, a data em que este poderá ser descontado e a data em que este foi realmente descontado. Neste exemplo, três tempos de validade são necessários.

2ª) Uma operação de DESFAZER (*undo operation*)

Em várias aplicações há a necessidade de desfazer uma operação. Uma operação de *rollback* neste caso não é apropriada, uma vez que desfazer uma operação deve ser limitada a uma única instância, não extendendo a todo BD.

Casos típicos em que esta operação é necessária está no domínio de BD CAD (Projeto Auxiliado por Computador).

Uma característica comum destas aplicações é que os dados temporais são altamente imprecisos e referem-se a tempos relativos. Relacionamentos entre dois eventos são mais frequentes que uma localização precisa no eixo do tempo dos eventos. Havendo a necessidade de junção de diferentes BD, possivelmente definidos em diferentes granularidades de tempo.

A maior parte do trabalho em BDT tem sido no contexto do modelo de dados relacional. Um número considerável de abordagens baseadas nos modelos ER e modelos baseados em objetos têm aparecido na literatura. O modelo de dados relacional é considerado insuficientemente expressivo para complexas aplicações de BD, tais como multimídia, sistemas de informação executiva, CAD e GIS. Estas aplicações apresentam requisitos fortes para modelar as relações temporais entre objetos. Temporalidade é uma parte importante (se não integral) desta geração de BD. Outra abordagem que começou nos anos oitenta é a incorporação de restrições, *triggers*, e regras em BD relacionais e BD Orientado-Objetos.

Identificar uma infraestrutura comum para esta geração de BDT, incluindo extensões para o modelo relacional e modelos baseados em objetos, é uma importante questão.

Um modelo temporal pode ser discutido em termos de três partes distintas: estruturas, operações e *constraints*. O componente estrutural de um modelo de dados preocupa-se com os objetos e suas relações. O componente operacional preocupa-se com a manipulação destes objetos. O componente *constraint* preocupa-se com as regras para a integridade estrutural e de manipulação sobre tempo.

Os conceitos básicos e componentes identificados para esta geração de modelos de dados temporais são classificados em três categorias:

- Propriedade Estrutural Temporal: descreve os objetos da aplicação em termos de suas propriedades e suas relações com outros objetos com o respectivo tempo;
- Propriedade Operacional Temporal: descreve a ação dos objetos quanto ao tempo, como reflexo das mudanças em suas propriedades;
- Propriedade *Constraint* Temporal: descreve as condições que devem ser satisfeitas durante a vida do objeto.

Um **objeto temporal** é definido como conjunto de uma ou mais propriedades temporais. Estas propriedades descrevem características estruturais, operacionais e de restrição dos objetos sobre o tempo. Uma **restrição temporal** é uma regra de BD que inclui também seu período de validade e é dividida em três partes: evento, condição e ação. Todas estas partes devem referir-se a pontos no tempo, mas pelo menos o evento ou a condição constituem a parte para a regra ser caracterizada como temporal.

O projeto da linguagem de consulta associada com os conceitos acima deve levar em consideração também as questões da linguagem tradicional tais como facilidade de uso, otimização, expressividade e implementabilidade.

As regras temporais e não-temporais podem ser vistas como a primeira classe de BD baseados em objetos. No caso das regras temporais, a história das regras deve ser guardada. Cada regra é associada com o tempo de transação e tempo de validade. O tempo de transação refere-se ao tempo de quando a regra foi gravada no BD. O tempo de validade refere-se aos pontos no tempo de quando esta regra foi aplicada (ativada). O tempo de validade de uma regra pode ser especificado explicitamente em termos de condições ou ocorrência de algum(s) evento(s). A ativação e desativação de regras é feita pela mudança de

seus tempos de validade. Há duas alternativas básicas para modelagem da histórias das regras:

1ª) A regra é considerada como uma unidade simples, então a mudança em um dos componentes é considerada como remoção (no caso de BDT, remoção da regra é a adição de uma nova regra);

2ª) A regra é considerada como sendo um objeto complexo e a história dos componentes individuais é mantida, pode-se representar diferentes versões da mesma regra.

Quando uma regra é ativada, a condição passa a ser verdadeira, a parte de ação da regra é executada. Em BDT, a ação pode incluir qualquer alteração no BD, alterações no tempo passado ou tempo futuro dos dados. Tais alterações são chamadas retro-ativas e pós-ativas, respectivamente.

Se alterações pós-ativas ou retro-ativas são permitidas em BDT, regras podem causar efeito nos dados de várias maneiras. A caracterização dos efeitos retro-ativos é mencionada a seguir. Nas seguintes definições, o “passado” é relativo ao tempo da operação ou *trigger* de uma regra:

Definição I - *Update* Retro-ativa: uma operação de alteração que modifica valores passados.

Definição II - Regra Retro-ativa: uma regra que inclui ação de uma *update* retroativa.

Definição III - Ativação de uma Regra retro-ativa: a aplicação de uma regra ao passado instantâneo.

Estas definições indicam que as regras podem causar efeitos nos dados retroativamente de duas maneiras: devido as regras retro-ativas ou devido a ativação das regras retro-ativas. O último caso pode ocorrer por duas razões: i) a regra foi introduzida no sistema com um tempo de validade que inclui intervalos

do tempo passado, ou; ii) uma alteração retro-ativa ocorreu, e a alteração nos elementos dispara uma regra em que era válida no tempo passado.

Foi apresentado apenas o caso retro-ativo, alterações pós-ativas são similares.

4.4 Consistência Temporal

A consistência de um BD é relativa ao efeito da execução em série de um conjunto de transação em um estado que é assumido ser consistente (condição de serializabilidade) e também é relativa a um conjunto de restrições que limitam o espaço do estado do BD. Assume-se que a condição de serializabilidade é satisfeita e a consistência está apenas no contexto das condições. Condições podem ser não-temporais, referindo-se a qualquer tempo de validade instantâneo, ou temporais, referindo-se a instantes particulares. É assumido que as condições podem ser compiladas em regras que reforçam-nas.

Para caracterizar as ações das regras em um BDT, há uma necessidade de distinguir entre um estado e um instante do BD. As seguintes definições referem-se a BD bitemporais. O termo **tempo do sistema** refere-se aos valores de tempo gerados pelo *clock* do sistema. É assumido que este valor de tempo é usado no domínio do tempo de transação. **Tempo de observação** diz respeito ao ponto de referência na linha de tempo da qual o estado do BD é observado. Em BD convencionais, o ponto de observação é sempre *now*. Em BDT, o ponto de observação pode ser menor ou igual ao *now*. Apenas os dados com tempo de transação menor que o tempo de observação podem ser vistos por uma consulta (ou uma transação).

5. IMPLEMENTAÇÃO DE BANCOS DE DADOS TEMPORAIS

Apesar das pesquisas de BDT já se estenderem por mais de 20 anos, existem poucos sistemas realmente utilizáveis. Experiências nas quais estudos de problemas encontrados (armazenamento e recuperação de informações) e mapeamento de modelos temporais para BDs tradicionais tendem a facilitar a criação de tais sistemas.

A implementação de bancos de dados temporais em SGBDs convencionais deve-se à inexistência de um SGBD totalmente temporal. Bancos de dados tradicionais podem ser utilizados se existir um mapeamento adequado entre o modelo temporal e o modelo do banco de dados utilizado [Hubler, 1999].

5.1 Implementação usando o *Oracle 8i Time Series*

Oracle Time Series [Oracle Corp., 2001] é uma extensão do *Oracle8i* que armazena e recupera dados baseados em tempo através de *object data types* (ODTs). Este pacote oferece um conjunto básico de funções e tipos de dados que acompanham a ferramenta *Oracle8i*. Estes tipos de dados, juntamente com as funções, permitem administrar e processar dados temporais de maneira mais conveniente do que seria possível usando unicamente tipos de dados e funções definidas pelo usuário. Pode-se usar ou adaptar tabelas para aplicações de *time series*, ou criar novas tabelas, adicionar ou modificar funções a fim de estender as capacidades do *Oracle Time Series*. As funções incluídas no pacote permitem realizar operações que vão de uma escala de simples a complexas, tais como:

- Encontrar a nota em uma disciplina em uma data específica;
- Calcular a nota média para uma disciplina em um ano em específico;

Tabela 3 – Histórico Escolar

| CodDisc | NomeDisc | Semestre | Nota | Situação |
|----------------|--------------------------------|-----------------|-------------|-----------------|
| CEX 103 | Cálculo I | 11/1997 | 72 | A |
| CEX 125 | Álgebra | 11/1997 | 54 | A |
| COM 101 | Fundamentos de Informática | 11/1997 | 84 | A |
| COM 102 | Algoritmos I | 11/1997 | 49 | R |
| EAS 110 | Teoria Geral Administração | 11/1997 | 78 | A |
| EAS 165 | Int aos Estudos Universitários | 11/1997 | 96 | A |
| COM 102 | Algoritmos I | 05/1998 | 49 | R |
| CEX 104 | Cálculo II | 05/1998 | 89 | A |
| CEX 109 | Física I | 05/1998 | 66 | A |
| CEX 110 | Física II | 05/1998 | 48 | R |
| COM 102 | Algoritmos I | 11/1998 | 50 | A |
| CEX 110 | Física II | 11/1998 | 80 | A |
| CEX 105 | Cálculo III | 11/1998 | 97 | A |

Esta parte de um Banco de Dados Escolar (Tabela 3) apresenta as seguintes características gerais:

- Múltiplas disciplinas, cada uma identificada por “CodDisc” que serão armazenadas no BD.
- Cada disciplina tem um ou mais rótulos temporais associados, a mesma disciplina pode ser cursada em semestres diferentes, se o aluno obter, por exemplo, uma ou mais reprovações.
- Cada rótulo temporal tem um calendário associado.

Oracle Time Series é parte de uma aplicação, ao invés de ser uma aplicação completa de usuário final. As aplicações que acessam e manipulam *time series* usam SQL ou PL/SQL.

No Anexo A, encontra-se um glossário de termos do pacote *Oracle Time Series*.

No Anexo B, encontra-se o script de implementação.

5.2 Oracle 8i Time Series e a Tecnologia Objeto-Relacional

A opção de manipulação de objetos no *Oracle8i* faz deste um SGBD-Objeto Relacional. Isto significa que usuários podem definir tipos de dados adicionais, especificando a estrutura dos dados e a maneira de operá-los. Trabalha de maneira eficiente com aplicações desenvolvidas usando as técnicas de programação orientada a objeto.

Oracle 8i Time Series focaliza-se em mecanismos de representação e acesso a dados baseados em tempos, suficientes para suportar muitas aplicações e o desenvolvimento de funções especializadas de *time series*.

5.3 Dados Históricos

Oracle 8i Time Series é especialmente usado na manipulação de dados históricos (Tabela 3). Este tipo de dado geralmente tem relativa simplicidade na estrutura, mas requisitos massivos de armazenamento. Os atributos (colunas) são relativamente poucos e fáceis de estruturar, entretanto o número de linhas é enorme. O número de funções que usuários podem querer realizar nos dados é significativa: encontrar várias somas, valores máximos e mínimos, médias, número de dias entre duas datas, média de progresso, e assim por diante.

6. 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 máximo 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).

6.1 Problemas no Processamento de Consultas Temporais

Além dos diversos problemas encontrados no processamento de consultas, o processamento de consultas temporais apresentam os seguintes:

- 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;
- 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.

7. CONCLUSÃO

As aplicações com dados temporais, geralmente, devem suportar um grande volume de dados. Por exemplo, o armazenamento do histórico de 2000 alunos de uma universidade, com cerca de 4 ou 5 anos e várias disciplinas cursadas semestralmente.

O tamanho total de um banco de dados pode variar, podendo chegar até centenas de gigabytes. Um SGBD que apresente uma maneira eficiente de armazenamento e acesso aos dados deve ser usado.

O SGBD deve permitir armazenar e manipular dados temporais com outros dados, considerando o mesmo BD. O BD pode conter, tanto tabelas temporais, quanto tabelas tradicionais.

Para acessar e manipular dados temporais, não é necessário empregar métodos ou linguagens diferentes, um uso adequado de SQL garante a condição temporal dos dados.

8. REFERÊNCIAS BIBLIOGRÁFICAS

ANTUNES, Dante C; HEUSER, Carlos A; EDELWEISS, N. **TempER: Uma Proposta de Modelagem de Dados Temporal**. 1998
www.pr.gov/celepar/celepar/ba/1998/bb75/dados.htm

ANTUNES, Dante C. **Modelagem Temporal de Sistemas: Uma Abordagem Fundamentada em Redes de Petri**. Porto Alegre: CPG da UFRGS, 1997. Dissertação de Mestrado.

EDELWEISS, N. **Bancos de Dados Temporais: Teoria e Prática**. XVII Jornada de Atualização em Informática – JAI. XVIII SBC. Belo Horizonte, 1998.

ELMASRI, R; KOURAMA-JIAN, Vram. **A Temporal Query Language Based on Conceptual Entities and Roles**. In *International Conference on the Entity-Relationship Approach*, 11, 1992, Karlsruhe, Germany. Proceedings Berlin: Springer Verlag, 1992. p.375-388. (Lecture Notes in Computer Science, v.645)

ELMASRI, R.; WUU, Gene T. J.; KORAMAJIAN, Vram. **A Temporal Model and Query Language for EER Databases**. In: Tansel, A. et al. *Temporal Databases: theory, designs and implementation*. Redwood City: The Benjamin/Cummings Publishing, 1993. p.212-229.

ELMASRI, R.; NAVATHE, S. **Fundamentals of Database Systems**. Benjamin/Cummings Publishing, Second Edition 1994.

FERG, S. **Modeling the Time Dimension in an Entity-Relationship Diagram**. In *4th International Conference on the Entity-Relationship Approach*, p. 200-286, Silver Spring, MD, 1985. Computer Society Press.

HUBLER, Patrícia Nogueira; EDELWEISS, N. **Implementação de um Sistema de Banco de Dados Temporal para o Modelo TF-ORM**. 1999
www.inf.ufrgs.br/pos/SemanaAcademica/Semana99/hubler/hubler.html

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

- KLOPPROGGE, M. R. **TERM: An Approach to Include the Time Dimension in the Entity-Relationship Model.** In *Proceedings of the Second International Conference on the Entity Relationship Approach*, p. 477-512, Washington, DC, October 1981.
- KHOSHAFIAN, S. N.; COPELAND, G.P. Object Identify. ACM SIGPLAN Notices, New York, v.21, n.11, p. 214-223, Sep. 1986. Trabalho apresentado na *Conference on Object-Oriented Programming, Languages and Applications*, OOPSLA, 1986, Portland.
- LOUCOPOULOS, P.; THEODOULIDIS, C.; WANGLER, B. **The Entity Relationship Time Model and Conceptual Rule Language.** In *International Conference on the Entity Relationship Approach*, 10., 1991, San Mateo, California.
- MOREIRA, Viviane Pereira; EDELWEISS, N. **Queries to Temporal Databases Supporting Schema Versioning.** XIV SBBD. Florianópolis, 1999.
- OLIVEIRA, José Palazzo M.; EDELWEISS, Nina. **Modelagem de Aspectos Temporais de Sistemas de Informação.** IX Escola de Computação, Recife, 1994.
- ORACLE Corporation; Oracle8i Time Series User's Guide.
<http://technet.oracle.com/doc/inter.815/a67294/toc.htm>
- SNODGRASS, R.; AHN, I. **A Taxonomy of time in databases.** Proceedings of the ACM SIGMOD International Conference on Management of Data, Texas, May 28-31, 1985. p. 236-246.
- SNODGRASS, R.; et al. **Towards an Infrastructure for Temporal Databases.** Report of an Invitational ARPA/NSF Workshop. March, 1994.
- TANSEL, A. et al. **Temporal Databases: Theory, Design and Temporal Model and Query Language for EER Databases.** In: Implementation. Redwood City: The Benjamin/Cummings Publishing, 1993. p.212-229.

TAUZOVICH, Branka. **Towards Temporal Extensions to the Entity Relationship Approach**, 10., 1991, San Mateo, California.

ANEXO A

I - Glossário de termos para o *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-2001 é especificada, então o 7º dia começa em cada Domingo; e Domingos e Sábados (0) são excluídos do calendário, enquanto Segunda a Sextas-Feiras (1), são incluídas no calendário.

Calendário: uma estrutura de dado 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).

Exceções: um rótulo temporal que não está em conformidade com um calendário padrão, mas que é significante 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-2001, 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-2001, será incluído no calendário quando normalmente Sábados são excluídos, define a data com uma on-exceção.

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 calendário.

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 CodDisc) e múltiplos valores temporais (datas em que determinada Disciplina foi cursada por um aluno) .

Time series irregular: um *time series* que não tem um calendário associado. Frequentemente, *time series* irregulares são dados temporais não previsíveis referindo-se a pontos não especificados no tempo ou a rótulos temporais que não podem ser caracterizados por um padrão de repetição. Entretanto, um *time series* irregular pode ser usado com dados previsíveis onde simplesmente não é necessário associar com um calendário. Um *time series* irregular pode ter longos períodos sem nenhum dados e curtos períodos com grande quantidade de dados. Exemplo: transações em um caixa eletrônico. Entretanto, um *time series*

irregular não precisa unicamente ser usado com grande volume de dados imprevisíveis.

Time series Regular: um *time series* que tem um calendário associado. Dados chegam previsivelmente em intervalos pré-definidos. Por exemplo, semestralmente, cadastram-se as notas dos alunos.

Time series grupo: o esquema de objetos para um *time series*. Este grupo é criado por procedimentos que acompanham o pacote *time series (TS_Tools)*, começando com a chamada ao procedimento *Begin_Create_TS_Group* e finalizando com a chamada *End_Create_TS_Group*.

II - Modelos no Oracle 8i Time Series

A maioria dos usuários enquadram-se em um dos poucos modelos, dependendo de suas necessidades. Há dois modelos básicos:

1) **Nenhuma necessidade de calendários:** 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*.

2) Necessidade de calendários: usuários necessitam usar calendários com seus rótulos temporais quando pretendem usar funções específicas do pacote, garantir a validade das operações de manipulação de dados (insert, update e delete) nos rótulos temporais. Entretanto, dependendo das necessidades, 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).

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 nos dados de entrada estão válidos. Por exemplo, os seguintes erros podem ser detectados:

- Há um rótulo temporal de 01-Jun-2000 quando na verdade deveria ser 01-Jun-2001, o ano foi incorretamente digitado e os *times series* estão restritos a dados de 2001;

- 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.

III - Componentes de um calendário

Os componentes de um calendário são: frequência, padrão e exceções.

- **Frequência:** uma frequência específica 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 4) que é usado nas chamadas de funções.

Tabela 4 – Códigos das frequências

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

Algumas frequências são flexíveis quanto à definição de **data âncora padrão**, outras frequências são mais restritivas (Tabela 5).

Tabela 5 – Freqüências e seus requerimentos

| Freqüência | Requerimentos |
|-------------------|--|
| segundo | A data âncora pode ser qualquer rótulo temporal válido para segundos. |
| minuto | A data âncora pode ser qualquer rótulo temporal válido para minutos. Os valores para segundos devem ser zero. |
| hora | Os valores para minutos e segundos devem ser zero. |
| dia | Os valores para horas, minutos e segundos devem ser zero. |
| semana | A data âncora pode iniciar-se em qualquer dia da semana. Por exemplo, definir um calendário semanal com uma data âncora de 23-Jun-2001 significa que cada rótulo temporal segue a partir daquele Sábado. |
| mês | Definir um calendário mensal com uma data âncora de 01-Jul-2001 significa que cada rótulo temporal é o primeiro dia de cada mês. |
| trimestral | Quatro vezes por ano. Definir um calendário com uma data âncora de 01-Jan-2001 significa que cada rótulo temporal é 1º de Janeiro, 1º Abril, 1º Julho e 1º Outubro. |
| anual | Uma vez por ano. Um calendário anual com uma data âncora de 15-Fev-2001, significa que cada rótulo temporal é o 15º dia de Fevereiro de cada ano que se segue. |
| 10-dias | Os 1º, 11º e 21º dias de cada mês. Nenhuma outra data âncora é permitida, qualquer outra data âncora é ignorada. |
| quinzenal | O 1º e o 16º dia de cada mês. Nenhuma data âncora é permitida, qualquer outra data âncora é ignorada. |
| semestral | Duas vezes por ano. Definir um calendário semestral com uma data âncora de 01-Jan-2001, significa que cada rótulo temporal é o 1º dia de Janeiro e o 1º dia de Julho. Definir um calendário com uma data âncora de 15-Fev-2001, significa que cada rótulo temporal deva ser 15º dia de Fevereiro e Agosto. |

Cada freqüência tem uma precisão associada. As funções do *Oracle8i Time Series* requerem que os rótulos temporais de entrada estejam precisos quanto à freqüência 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 freqüência do calendário associado. Um rótulo temporal de 09-Set-1997 está impreciso se a freqüência é mês(6) ou ano(8) 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-200 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)) 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:** rótulos temporais que não estão em conformidade com o padrão do calendário, mas que são importantes na sua definição.

Oracle Time Series providencia tipos de dados para se trabalhar com calendários e *time-series*. Todos estes tipos de dados são instalados sob o esquema **ORDSYS**.

IV - Descrição do Calendário

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 cria um tabela chamada **tab_calendario_comercial** e define um calendário chamado **DiasComercio**. O calendário **DiasComercio** inclui os dias da semana - segundas a sextas-feiras, mas exclui os feriados 01-Maio-2001 e 25-Dez-2001 e sábados e domingos.

```
CREATE TABLE tab_calendario_comercial OF ORDSYS.ORDTCalendar;  
  
INSERT INTO tab_calendario_comercial VALUES  
  ( ORDSYS.ORDTCalendar(  
    0,  
    'DiasComercio',  
    4,  
    ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(0,1,1,1,1,1,0),  
      to_date('01/01/1995', 'DD-MM-YYYY')),  
    to_date('01/01/1990', 'DD-MM-YYYY'),  
    to_date('01/01/2001', 'DD-MM-YYYY'),  
    ORDSYS.ORDTExceptions(to_date('01/05/1995', 'DD-MM-YYYY'),  
      to_date('25/12/1995', 'DD-MM-YYYY')),  
    ORDSYS.ORDTExceptions()  
  ));
```

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

```
CREATE TYPE ORDSYS.ORDTCalendar AS OBJECT (  
  caltype INTEGER, --indica o tipo do calendário(0 = padrão)  
  name VARCHAR2(256), --nome para o calendário  
  frequency INTEGER, -- dia, mês, semestre, ano, etc.
```

```
pattern ORDSYS.ORDTPattern, --modelo de repetição dos time
    --series
minDate DATE, -- o calendário começa com esta data. Ex:
    -- data_matrícula do aluno
maxDate DATE -- o calendário finaliza com esta data. Ex:
    -- data_colação_grau
offExceptions ORDSYS.ORDTExceptions,
onException ORDSYS.ORDTExceptions);
```

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.

ANEXO B

Script de Implementação usando o SQL-PLUS

```
SQL> CONNECT;  
Enter user-name: rose  
Enter password: *****  
Connected.  
  
SQL> exec ORDSYS.TSTools.Begin_Create_TS_Group('tsecola','flat');  
-- A função Begin_Create_TS_Group inicializa o contexto para criação de um  
sistema time series,  
-- 'tsecola' é o nome usado nos procedimentos da ferramenta administrativa do  
pacote time series.  
-- Parâmetros para a função:  
-- name - nome do grupo time series a ser criado;  
-- storage_model - armazenamento do time series (modelo de armazenamento).  
  
SQL> exec ORDSYS.TSTools.Set_Flat_Attributes(tsname_colname =>  
'CodDisc',tsname_length =>10);  
-- 'CodDisc' é o nome do time series para funções.  
-- Exemplos de valores para CodDisc incluem CEX101, ...  
  
SQL> exec ORDSYS.TSTools.ADD_VARCHAR2_COLUMN('NomeDisc',40);  
SQL> exec ORDSYS.TSTools.ADD_NUMBER_COLUMN('Nota');  
SQL> exec ORDSYS.TSTools.ADD_VARCHAR2_COLUMN('Situação',10);  
-- definição explícita dos atributos (Tabela 3)
```

```
SQL> exec ORDSYS.TSTools.End_Create_Ts_Group;
-- ao fim da criação e especificação do grupo Time Series, os esquemas dos
seguintes objetos são criados automaticamente com a chamada de
End_Create_Ts_Group;
-- TSESCOLA_TAB: tabela de armazenamento para os dados (Figura 6);
```

| Nome | Esquema | Tipo de Dados | Tamc... | Escala | Ref. | Nulo... | Valor Default |
|----------|----------|---------------|---------|--------|------|---------|---------------|
| CODDISC | <Nenhum> | VARCHAR2 | 10 | | ✗ | ✗ | |
| TSTAMP | <Nenhum> | DATE | | | ✗ | ✗ | |
| NOMEDISC | <Nenhum> | VARCHAR2 | 20 | | ✗ | ✓ | |
| NOTA | <Nenhum> | NUMBER | 0 | 0 | ✗ | ✓ | |
| SITUAÇÃO | <Nenhum> | VARCHAR2 | 10 | | ✗ | ✓ | |

Figura 6 – TSESCOLA_TAB: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series

```
-- TSESCOLA_MAP: tabela de mapeamento que associa um calendário com
cada time series (Figura 7);
```

| Nome | Esquema | Tipo de Dados | Tamc... | Escala | Ref. | Nulo... | Valor Default |
|---------|----------|---------------|---------|--------|------|---------|---------------|
| CODDISC | <Nenhum> | VARCHAR2 | 10 | | ✗ | ✗ | |
| CALNAME | <Nenhum> | VARCHAR2 | 256 | | ✗ | ✓ | |

Figura 7 – TSESCOLA_MAP: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series

-- TSESCOLA_CAL: tabela de definição de um calendário (Figura 8);

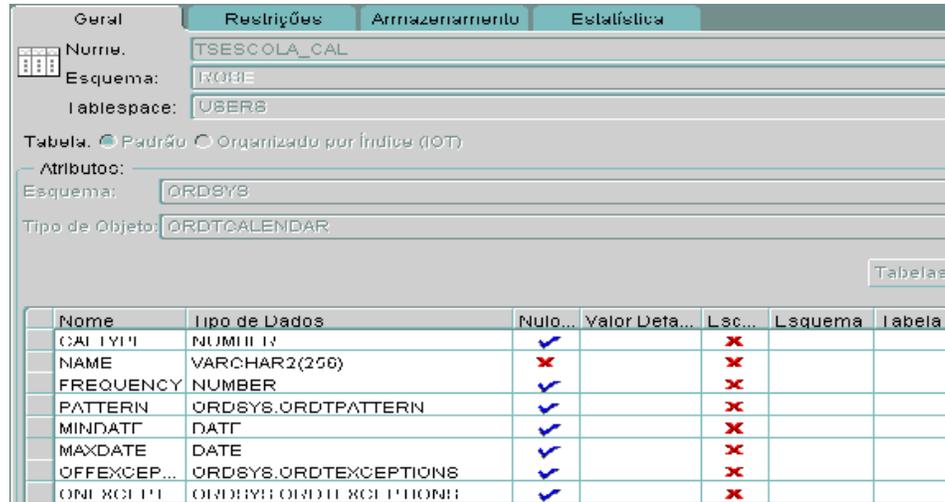


Figura 8 – TSESCOLA_CAL: Tabela automaticamente criada pelo SGBD ao término da definição de um grupo time series

-- TSESCOLA_RVW: visão relacional para operações de insert, update e delete protegidas.

-- TSESCOLA: visão de objeto.

```
SQL> CREATE TABLE meu_calendario of ORDSYS.ORDTCalendar;
```

-- comando apenas ilustrativo, está tabela já foi criada (TSESCOLA_MAP) ao término da criação do grupo time series

```
SQL> INSERT INTO tsescola_cal VALUES
( ORDSYS.ORDTCalendar(
    0,
    'Semestral',
    18,
    ORDSYS.ORDTPattern(ORDSYS.ORDTPatternBits(1,1),
    to_date('01/08/1997', 'DD-MM-YYYY')), -- data âncora (as
    --notas são cadastradas no final de cada semestre!!!
    to_date('08/08/1997', 'DD-MM-YYYY')),
```

```

        to_date('07/07/2001', 'DD-MM-YYYY'),
        null,
        null));

SQL>
DECLARE
    tstCal ordsys.ordtcalendar;
    temp integer;
    validFlag integer;
BEGIN
    -- Selecione um calendário into tstCal from tsescola.
    select value(cal) into tstCal
    from tsescola_cal cal
    where cal.name = 'Semestral';

    -- Mostra o calendário
    select ordsys.timeseries.display(tstCal) into temp from dual;
    dbms_output.new_line;

    validFlag := ORDSYS.CALENDAR.IsValidCal(tstCal);

    if (validFlag = 1) then
        dbms_output.put_line('O calendário Semestral está valido.');
```

-- procedimento para validar o calendário para certeza de que não há nenhum erro na definição do calendário. Resultado do procedimento:

```

Calendar Name = Semestral
Frequency = 18 (semi-annual)
MinDate = 08-AUG-97
MaxDate = 07-JUL-01
patBits: 1,1
patAnchor = 08-AUG-97
onExceptions : Atomic NULL
offExceptions : Atomic NULL
O calendário Semestral está valido.
```

-- Quando se cria um grupo time_series, uma tabela de mapeamento (map table) é automaticamente criada.

-- Esta tabela tem por default o nomegrupo_MAP. Neste exemplo: TSESCOLA_MAP.

-- Um map table mantém um junção entre um time series (tal como especificado pelo CodDisc) e um calendário.

-- O seguinte script mostra a tabela de mapeamento:

```
CREATE TABLE tsescola_map (  
    CodDisc VARCHAR2(10),  
    calname VARCHAR2(256),  
    CONSTRAINT pk_tsescola_map PRIMARY KEY (CodDisc));
```

-- Para cada linha em um map table, o nome do calendário (calname) pode ser nulo ou conter o nome de um calendário.

-- Se na coluna nome do calendário for nula, nenhum calendário será usado para as funções Time Series.

-- O seguinte exemplo cria um linha na tabela TSESCOLA_MAP para cada Cod_Disc na tabela TSESCOLA_TAB

```
-- INSERT INTO tsescola_map(CodDisc)  
    SELECT DISTINCT CodDisc FROM tsescola.tsescola_tab;
```

-- Se na coluna nome do calendário contém o nome de um calendário, tal calendário é usado para funções Time Series especificando qual o rótulo temporal.

-- O exemplo cria duas linhas na tabela tsescola_map, associado dois Cod_Disc com o calendário Semestral:

```
-- INSERT INTO tsescola_map VALUES ('CEX103', 'Semestral');  
-- INSERT INTO tsescola_map VALUES ('EAS165', 'Semestral');
```

-- Para linhas onde o nome de um calendário é especificado, pode-se adotar qualquer uma das seguintes estratégias, dependendo de qual calendário se aplica a qual time series:

-- 1) Usar o mesmo calendário para todos time series. Por exemplo, mapear todos CodDisc para um calendário único.

-- 2) Usar um calendário separado para cada time series. Por exemplo, criar um calendário COM128 para esta disciplina, um outro calendário para outra disciplina, com outra granularidade diferente.

-- 3) Usar uma combinação das abordagens: usar o mesmo calendário para múltiplos time series, e talvez usar um calendário para um único time series.

-- Popular a tabela TSESCOLA_TAB com os seguintes dados:

CEX103, CálculoI, 11/1997, 72, A
CEX125, Álgebra, 11/1997, 54, A
COM101, Fundamentos, 11/1997, 84, A
COM102, AlgoritmosI, 11/1997, 49, R
COM102, AlgoritmosI, 05/1998, 49, R
COM102, AlgoritmosI, 11/1998, 50, A
...

```
SQL> INSERT INTO tsescola_tab values ('CEX103','01-NOV-97',  
'CálculoI', 72, 'A');
```

-- Se há uma grande quantidade de dados, pode-se usar o utilitário SQL*Loader.

-- Formular as consultas

```
SQL> SELECT CodDisc, to_char(tstamp) Semestre, Nota, Situação  
2 FROM tsescola_tab  
3 WHERE CodDisc = 'COM102';
```

| CODDISC | SEMESTRE | NOTA | SITUAÇÃO |
|---------|-----------|------|----------|
| COM102 | 01-JUN-98 | 49 | R |
| COM102 | 01-NOV-98 | 50 | A |

-- 1ª Maneira para exibir timeseries 'COM102'

```
SQL> SELECT * FROM the
      2      (SELECT CAST(ORDSYS.TIMESERIES.ExtractTable(ts.nota) AS
                    ordsys.ORDTNumTab)
      3      FROM tsescola ts
      4      WHERE ts.CodDisc = 'COM102');
```

| TSTAMP | VALUE |
|-----------|-------|
| 01-JUN-98 | 49 |
| 01-NOV-98 | 50 |

-- A função ExtractTable retorna uma tabela timeseries (ORDTNumTab ou ORDTVvarchar2Tab0 associaco com um time series

-- CREATE TYPE ORDSYS.ORDTNumCell AS OBJECT

```
( tstamp DATE
  value NUMBER);
```

-- CREATE TYPE ORDSYS.ORDTNumTab AS TABLE OF ORDSYS.ORDTNumCell;

-- 2ª Maneira para exibir o timeseries 'COM102'

```
SQL> DECLARE
      2  varTemp INTEGER;
      3  BEGIN
      4      SELECT ORDSYS.TimeSeries.Display(ts.nota) into varTemp
      5      FROM tsescola ts
      6      WHERE ts.CodDisc = 'COM102';
      7  END;
      8  /
```

Name = NOTA COM102

Calendar Data: Atomic NULL

Series Data:

```
-----  
Date      Value  
01-JUN-98  49  
01-NOV-98  50
```

```
SQL> SELECT ORDSYS.TimeSeries.TSAvg(nota),  
2          ORDSYS.TimeSeries.TSVariance(nota),  
3          ORDSYS.TimeSeries.TSStdDev(nota)  
4 FROM tsescola  
5 WHERE CodDisc= 'COM102';
```

```
ORDSYS.TIMESERIES.TSAVG(NOTA):      49.5  
ORDSYS.TIMESERIES.TSVARIANCE(NOTA): .5  
ORDSYS.TIMESERIES.TSSTDDEV(NOTA):  .707106781
```

-- Calculando várias funções estatísticas