



INSTITUTO TECNOLÓGICO DE AERONÁUTICA
CE-240 PROJETO DE SISTEMAS DE BANCO DE DADOS

Prof. Dr. Adilson Marques da Cunha

Lista de Exercícios 4

**Implementação de um Banco de Dados Relacional e sua conversão para os
modelos Hierárquico, Rede e Orientado a Objetos**

Antônio Magno Lima Espescht
antonio.espescht@gmail.com

São José dos Campos - SP

18 de maio de 2009

Sumário

1	Introdução	1
1.1	Título	1
1.2	Motivação	1
1.3	Objetivo	1
2	Conteúdo.....	1
2.1	Instalação do Oracle 11g	1
2.2	Implementação do Módulo AFE.....	2
2.3	Inserção de Dados no Módulo AFE.....	2
2.4	Realização de Queries no Banco de Dados	2
2.4.1	Query com Uma Tabela.....	2
2.4.2	Query com Duas Tabelas	3
2.4.3	Query com Três Tabelas	3
2.4.4	Query Georeferenciada	3
2.5	Dicionário de Dados.....	4
2.5.1	Dicionário de Dados Propriamente Dito.....	4
2.5.1.1	Tabela ESTATISTICA.....	4
2.5.1.2	Tabela FILTRO.....	6
2.5.1.3	Tabela MEDIDOR	6
2.5.1.4	Tabela OPERADOR	7
2.5.1.5	Tabela PCD.....	7
2.5.1.6	Tabela PERIODO	8
2.5.1.7	Tabela SERIE_HISTORICA	8
2.5.2	Diretório de Dados	9
2.5.2.1	Peopleware:	9
2.5.2.2	Software:.....	9
2.5.3	Dicionário de Recursos de Dados	10
2.5.4	Dicionário de Metadados.....	10
2.6	Conversão para o Modelo Hierárquico	11
2.7	Conversão para o Modelo de Rede	12
2.8	Conversão para o Modelo Orientado a Objetos.....	12
3	Conclusão e Comentários.....	13
4	Referências	14
5	Anexos.....	15
5.1	Instalação do Oracle 11g	15
5.2	Modelo Físico na 3NF.....	16
5.3	Comandos para Criação de Tabelas no Oracle 11g	16
5.4	Comandos SQL para Alterar a Tabela PCD.....	31
5.5	Exemplos de Dados.....	32

5.6 Comandos SQL para Inserir Exemplos de Dados 34

1 Introdução

Em atendimento à Lista de Exercícios (ListEx) 4, foram realizadas as atividades:

- Implementar, na Terceira Forma Normal (3NF) e no modelo relacional, a versão 1.0 do módulo **AFE** (Aplicação de Filtros e Cálculo de Estatísticas) do *Sistema de Aquisição, Tratamento, Monitoramento e Difusão de Dados Hidrológicos (Sis ATMH)* no sistema Oracle 11g.
- Testar as funcionalidades do Módulo AFE, realizando diversas queries.
- Apresentar a versão 1.0 do Dicionário de Dados do Módulo AFE.
- Pesquisar os modelos de dados Hierárquico, Rede e Orientado a Objeto, convertendo a implementação relacional para estes modelos.

Este texto sumaria os resultados obtidos.

1.1 Título

Implementação, no modelo relacional e em sua Terceira Forma Normal (3NF), da versão 1.0 do módulo **AFE** (Aplicação de Filtros e Cálculo de Estatísticas) do *Sistema de Aquisição, Tratamento, Monitoramento e Difusão de Dados Hidrológicos (Sis ATMH)*.

1.2 Motivação

Resolver esta lista de exercícios é uma oportunidade para ver, na prática, como são implementadas as tabelas definidas de modo teórico na ListEx 3. Será também a primeira oportunidade, neste curso, para inserirmos dados e efetuarmos consultas em nossas tabelas usando o Oracle 11g com recursos georeferenciados.

Outra motivação é a oportunidade para discutir vantagens e desvantagens de outros modelos de dados (rede, hierárquico e orientado a objeto) em comparação com o modelo relacional.

1.3 Objetivo

Desenvolver a versão 1.0 do modelo de dados do *Módulo de Aplicação de Filtros e Cálculo de Estatísticas (AFE)*, na *Terceira Forma Normal (3NF)*, visando melhorar os tempos de acesso e reduzir anomalias de inclusão, atualização e exclusão de dados.

2 Conteúdo

2.1 Instalação do Oracle 11g

Após várias tentativas infrutíferas de acesso ao Oracle 11g com o SQL Developer no endereço host 161.24.9.5:1521, decidimos instalar este banco de dados localmente.

A partir da URL <http://www.oracle.com/technology/products/database/oracle11g> foi baixado o Oracle 11.1.0.6.0 com bug fix 7389110.

Este banco de dados foi instalado e configurado em uma máquina PC com processador Intel Core 2 Quad CPU Q6600 @2.4GHz 2.75GB RAM HD 500GB 7500RPM cujo nome na rede interna é TONI10.

Logo, a URL para administração do Oracle 11g ficou sendo <https://toni10:1158/em> (Cf. Anexo I, Figura 5, Pág. 15).

2.2 Implementação do Módulo AFE

Na ListEx 3 foi utilizado o ERWin 4.0 Build 1338 para obter o Modelo Físico do *Módulo de Aplicação de Filtros e Cálculo de Estatísticas (AFE)*, na *Terceira Forma Normal (3NF)*, (Cf. Anexo I, Figura 6, Pág. 16).

A partir deste modelo, utilizou-se o ERWin para gerar comandos em SQL para criar tabelas no Oracle 11g (Cf. Anexo I, Seção 5.3. Pág. 16).

Para criar as tabelas foi utilizado o aplicativo SQL*Plus do Oracle 11g. Bastou copiar os comandos em SQL para o arquivo `d:\listex4.SQL` e, em seguida, executar o comando `@d:\listex4.SQL` no SQL*Plus. Para mais informações sobre o SQL*Plus, basta consultar: http://www.oracle.com/technology/tech/sql_plus.

2.3 Inserção de Dados no Módulo AFE

Durante a inserção de dados, observou-se que a tabela PCD não estava corretamente georeferenciada de acordo com a sintaxe do Oracle 11g.

Assim, foi criado, e executado via SQL*Plus, o script `listex4_pcd.SQL` (Cf. Anexo I, Seção 5.4, Pág. 31) que remove os campos `pcd_altitude`, `pcd_longitude` e `pcd_altitude` e adiciona o campo `pcd_localizacao` que é georeferenciado.

Como “massa de dados para teste” foram usados os dados de exemplo da ListEx 3 (Cf. Anexo I, Seção 5.5. Pág. 32).

Os comandos SQL para inserir dados foram digitados no script `listex4_dados.SQL` e estão reproduzidos no Anexo I, Seção 5.6. Pág. 34).

2.4 Realização de Queries no Banco de Dados

Foram criadas queries de exemplo envolvendo de uma a três tabelas de nosso modelo. Em seguida, foi criada uma quarta query envolvendo dados georeferenciados.

Cada query está descrita em linguagem natural e em SQL, mostrando-se em seguida uma tabela com o resultado obtido.

2.4.1 Query com Uma Tabela

Listar nome e telefone dos operadores.

```
SELECT ope_nome as Nome,  
       ope_telefone as Telefone  
FROM operador  
ORDER BY 1,2;
```

Nome	Telefone
Cacique Juruna	
José Silva	55 92 1234 5678
Maria Silva	55 68 4321 8765

2.4.2 Query com Duas Tabelas

Listar nome e telefone dos operadores cujos dados apresentaram pelo menos um outlier.

```
SELECT DISTINCT ope_nome as Nome,
               ope_telefone as Telefone
FROM operador o
JOIN serie_historica sh on sh.ope_chapa = o.ope_chapa
WHERE shi_outlier = 'T'
ORDER BY 1,2;
```

Nome	Telefone
José Silva	55 92 1234 5678

2.4.3 Query com Três Tabelas

Listar nome e telefone dos operadores cujos dados apresentaram pelo menos um outlier, exibindo também o nome do medidor e a data em que os outliers foram medidos.

```
SELECT ope_nome as Nome,
       ope_telefone as Telefone,
       med_descricao as Medidor,
       shi_data as Data,
       shi_valor as Valor
FROM operador o
JOIN serie_historica sh on sh.ope_chapa = o.ope_chapa
JOIN medidor m on m.med_id = sh.med_id
WHERE shi_outlier = 'T'
ORDER BY 1,2,3,4;
```

Nome	Telefone	Medidor	Data	Valor
José Silva	55 92 1234 5678	XPT01	01-JAN-00	42

2.4.4 Query Georeferenciada

Listar identificador e coordenadas dos PCDs que não têm outliers.

```
select a.pcd_id,a.pcd_localizacao
from pcd a
where a.pcd_id not in (select b.pcd_id
                      from pcd b
                      join serie_historica c on c.pcd_id = b.pcd_id
                      where shi_outlier = 'T');
```

```

id   pcd_localizacao
2    sdo_geometry(3001,8292,mdsys.sdo_point_type(-43.7775, -22.5389,
389),null,null)
3    sdo_geometry(3001,8292,mdsys.sdo_point_type(-44.1247, -22.7489,
440),null,null)

```

2.5 Dicionário de Dados

De acordo com a 9ª Técnica de Bancos de Dados[1], um Sistema de Dicionário de Dados é composto de:

1. Dicionário de Dados propriamente dito: descreve atributos, entidades, relacionamentos e objetos associados.
2. Diretório de Dados: descreve processos associados às entidades.
3. Dicionário de Recursos de Dados: descreve fisicamente entidades e ambientes associados.
4. Dicionário de Metadados: descreve as entidades conceitualmente, no nível máximo de abstração.

2.5.1 Dicionário de Dados Propriamente Dito

No dicionário abaixo considerou-se:

1. A **SEGURANÇA** de todas as tabelas e todos os atributos é garantida por procedimentos de backup/restore bem como acesso restrito a técnicos devidamente autorizados e capacitados para realizar inserções, remoções e alterações dos dados.
2. A **PRIVACIDADE** das tabelas e atributos é garantida por procedimentos de acesso autorizado por senha a qual é gerada, distribuída, armazenada e verificada por procedimentos específicos que propiciam o nível de privacidade necessário.
3. A segurança dos atributos é garantida por relacionamentos de chave estrangeira, pela escolha do formato e do tamanho de cada atributo, pelo uso das diretivas NULL e NOT NULL bem como procedimentos de depuração realizados via stored procedures e/ou pela interface de entrada de dados.
4. Além das chaves estrangeiras, utilizou-se triggers criadas pelo ERWin 4.0 para garantir a integridade dos dados.
5. PK é sigla para Primary Key (Chave Primária).
6. FK é sigla para Foreign Key (Chave Estrangeira).

O relacionamento entre as entidade pode ser visto no Anexo I, Figura 6, Pág. 16.

2.5.1.1 Tabela ESTATISTICA

A tabela ESTATISTICA armazena estatísticas dos MEDIDORES dos Pontos de Coleta de Dados (PCDs) em determinados PERIODOS de tempo.

2.5.1.1.1 Atributos da Tabela ESTATISTICA

Atributo	Formato	Tamanho	NULL	Descrição
pcd_id	INTEGER	38	NOT NULL	Identificador do PCD
med_id	INTEGER	38	NOT NULL	Identificador do MEDIDOR
per_id	CHAR	1	NOT NULL	Identificador do PERIODO
est_dia	CHAR	2	NULL	<p>Dia ao qual se refere o PERIODO em que a estatística foi efetuada. É preenchido com NULL se a estatística se refere a um determinado mês e ano.</p> <p>Valores válidos 01 a 31 (sempre com dois dígitos).</p> <p>Valor máximo de acordo com o valor de est_mes.</p>
est_mes	CHAR	2	NULL	<p>Mês ao qual se refere o PERIODO em que a estatística foi efetuada. É preenchido com NULL se a estatística se refere a um determinado ano.</p> <p>Valores válidos 01 a 12 (sempre com dois dígitos).</p>
est_ano	CHAR	4	NOT NULL	<p>Ano ao qual se refere o PERIODO em que a estatística foi efetuada.</p> <p>Valores válidos 1900 a 2050 (sempre com quatro dígitos).</p>
est_minimo	FLOAT	126	NULL	<p>Valor mínimo da medida med_id do PCD pcd_id no período per_id e na data formada por est_dia, est_mes e est_ano.</p> <p>Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).</p>
est_media	FLOAT	126	NULL	<p>Valor médio da medida med_id do PCD pcd_id no período per_id e na data formada por est_dia, est_mes e est_ano.</p> <p>Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).</p>
est_maximo	FLOAT	126	NULL	<p>Valor máximo da medida med_id do PCD pcd_id no período per_id e na data formada por est_dia, est_mes e est_ano.</p> <p>Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).</p>
est_desvio_padrao	FLOAT	126	NULL	<p>Desvio-padrão da medida med_id do PCD pcd_id no período per_id e na data</p>

Atributo	Formato	Tamanho	NULL	Descrição
				formada por est_dia, est_mes e est_ano. Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).

2.5.1.1.2 Restrições da Tabela ESTATISTICA

Nome	Tipo	Campos
PK_ESTATISTICA	PK	pcd_id, med_id, per_id
FK_ESTATISTICA_PER_ID	FK PERIODO (per_id)	per_id
FK_ESTATISTICA_MED_ID	FK MEDIDA (med_id)	med_id
FK_ESTATISTICA_PCD_ID	FK PCD (pcd_id)	pcd_id

2.5.1.2 Tabela FILTRO

A tabela FILTRO armazena dados sobre os filtros utilizados no Módulo AFE.

2.5.1.2.1 Atributos da Tabela FILTRO

Atributo	Formato	Tamanho	NULL	Descrição
fil_id	INTEGER	38	NOT NULL	Identificador do Filtro
fil_nome	VARCHAR	200	NOT NULL	Nome do Filtro
fil_desc	VARCHAR	200	NOT NULL	Descrição do Filtro.

2.5.1.2.2 Restrições da Tabela FILTRO

Nome	Tipo	Campos
PK_FILTRO	PK	fil_id

2.5.1.3 Tabela MEDIDOR

A tabela MEDIDOR armazena dados sobre os tipos de medidores instaláveis nos PCDs.

A leitura destes medidores é armazenada na tabela SERIE_HISTORICA.

Estatísticas dos dados destes medidores são armazenadas na tabela ESTATISTICA.

2.5.1.3.1 Atributos da Tabela MEDIDOR

Atributo	Formato	Tamanho	NULL	Descrição
med_id	INTEGER	38	NOT NULL	Identificador do Medidor
med_parametro	VARCHAR	200	NOT NULL	Nome do parâmetro medido.
med_descricao	VARCHAR	200	NOT NULL	Descrição do Medidor.

2.5.1.3.2 Restrições da Tabela MEDIDOR

Nome	Tipo	Campos
IUQ_MEDIDOR	Índice único	med_parametro
PK_MEDIDOR	PK	med_id

2.5.1.4 Tabela OPERADOR

A tabela OPERADOR armazena dados sobre os funcionários que coletam dados dos medidores.

Por simplicidade, considerou-se apenas 4 atributos. Num banco de dados profissional esta tabela poderia ter dezenas de atributos tais como: data de nascimento, cpf, rg, dados bancários etc.

2.5.1.4.1 Atributos da Tabela OPERADOR

Atributo	Formato	Tamanho	NULL	Descrição
ope_chapa	INTEGER	38	NOT NULL	Chapa do operador.
ope_nome	VARCHAR	200	NOT NULL	Nome do operador
ope_telefone	VARCHAR	200	NOT NULL	Telefone do operador.
ope_endereco	VARCHAR	200	NOT NULL	Endereço do operador.

2.5.1.4.2 Restrições da Tabela OPERADOR

Nome	Tipo	Campos
PK_OPERADOR	PK	ope_chapa

2.5.1.5 Tabela PCD

A tabela PCD armazena dados sobre os Pontos de Coleta de Dados (PCDs).

Cada PCD possui um número de série numérico que o identifica.

Um mesmo PCD pode ser levado de uma localização para outra.

2.5.1.5.1 Atributos da Tabela PCD

Atributo	Formato	Tamanho	NULL	Descrição
pcd_id	INTEGER	38	NOT NULL	Identificador do PCD
pcd_numero_serie	INTEGER	38	NOT NULL	Número de Série do PCD
pcd_data_instalacao	DATE	16	NOT NULL	Data da instalação do PCD na localização dada por pcd_localizacao.
pcd_localizacao	MDSYS.SDO_GEOMETRY	200	NOT NULL	Localização do PCD

Atributo	Formato	Tamanho	NULL	Descrição
				(latitude, longitude e altitude).

2.5.1.5.2 Restrições da Tabela PCD

Nome	Tipo	Campos
IDX_PCD_SPATIAL	Índice Espacial	pcd_localizacao
IUQ_PCD	Índice único	pcd_numero_serie,pcd_data_instalacao
PK_PCD	PK	pcd_id

2.5.1.6 Tabela PERIODO

A tabela PERIODO armazena dados sobre os períodos de tempo aos quais faz sentido se obter ESTATISTICAS.

2.5.1.6.1 Atributos da Tabela PERIODO

Atributo	Formato	Tamanho	NULL	Descrição
per_id	CHAR	1	NOT NULL	Identificador do Período. Exemplo: M para mês, D para dia, A para ano.
per_nome	VARCHAR	200	NOT NULL	Nome do Período. Exemplo: Mês, Dia ou Ano.
per_desc	VARCHAR	200	NOT NULL	Descrição do Período.

2.5.1.6.2 Restrições da Tabela PERIODO

Nome	Tipo	Campos
IUQ_PERIODO	Índice único	per_nome
PK_PERIODO	PK	per_id

2.5.1.7 Tabela SERIE_HISTORICA

A tabela SERIE_HISTORICA armazena as leituras coletadas pelo OPERADOR de um MEDIDOR de um PCD ao longo do tempo.

2.5.1.7.1 Atributos da Tabela SERIE_HISTORICA

Atributo	Formato	Tamanho	NULL	Descrição
ope_chapa	INTEGER	38	NOT NULL	Chapa do operador que fez a leitura.
pcd_id	INTEGER	38	NOT NULL	Identificador do PCD no qual foi feita a

Atributo	Formato	Tamanho	NULL	Descrição
				leitura.
med_id	INTEGER	38	NOT NULL	Identificador do MEDIDOR que forneceu os dados lidos.
shi_data	DATE	18	NOT NULL	Data da leitura.
shi_valor	FLOAT	126	NULL	Valor lido no MEDIDOR do PCD pelo OPERADOR na data shi_data. Se NULL significa que o dado estava indisponível.
shi_outlier	CHAR	1	NULL	Valores possíveis: T: valor de shi_valor é um outlier. F: valor de shi_valor não é um outlier. NULL: não foi feita análise se o valor lido é ou não é um outlier.

2.5.1.7.2 Restrições da Tabela SERIE_HISTORICA

Nome	Tipo	Campos
PK_SERIE_HISTORICA	PK	ope_chapa, pcd_id, med_id, shi_data
FK_SERIE_HISTORICA_OPE_CHAPA	FK OPERADOR (ope_chapa)	ope_chapa
FK_SERIE_HISTORICA_MED_ID	FK MEDIDA (med_id)	med_id
FK_SERIE_HISTORICA_PCD_ID	FK PCD (pcd_id)	pcd_id

2.5.2 Diretório de Dados

O Diretório de Dados deve relacionar informações sobre relacionamentos entre atributos, entidades e usuários.

Em nosso protótipo do Módulo ADF, todos os atributos e entidades são acessados pelos seguintes tipos de usuários:

2.5.2.1 Peopleware:

Administradores de Bancos de Dados (DBAs): podem criar, remover ou alterar tabelas, atributos, índices, funções e procedimentos visando melhorar o desempenho do sistema e/ou adequar os dados a novas regras de negócio.

Desenvolvedores: desenvolvem novos sistemas ou melhoram os sistemas já existentes que fazem a interface entre usuários finais e o banco de dados.

2.5.2.2 Software:

Sistemas de Interface: são operados por usuários autenticados por senha, que consultam os dados armazenados e inserem novos dados. Os Sistemas de Interface zelam para que dados não sejam destruídos por erros de operação dos usuários.

Sistemas automáticos de inserção de dados: coletam dados de estações remotas, fazem depuração e aglutinação, inserindo o resultado final no banco de dados e reportando erros encontrados.

Sistemas para realizar e restaurar cópias de segurança.

2.5.3 Dicionário de Recursos de Dados

Atualmente, os dados estão armazenados na pasta F:\app\vidvoo\ce240\toni10 de um computador IBM PC Intel Core 2 Quad CPU Q6600 @2.4GHz 2.75GB RAM HD 500GB 7500RPM cujo nome na rede interna é TONI10.

Este computador é protegido por um No-Break de 3000 KVA o que lhe dá autonomia para funcionar até 2h. Portanto, seu funcionamento é praticamente 24x7 embora o acesso aos dados seja esporádico (apenas para resolução das ListEx).

O Sistema Operacional é o Windows XP 2002 SP3 em inglês.

O Sistema Gerenciador de Banco de Dados é o Oracle 11g versão 11.1.0.6.0 com bug fix 7389110.

O sistema é acessado via https na URL <https://toni10:1158/em> (ver Anexo I, Figura 5, Pág. 15) ou diretamente via SQL*Plus.

Parte dos scripts de criação das tabelas e triggers foi gerada pelo ERWin 4.0 Build 1338.

Atualmente, o sistema tem uso apenas em rede local. Em breve, o sistema será instalado no Laboratório da FCMF de onde terá acesso remoto com substancial alteração das características acima descritas.

2.5.4 Dicionário de Metadados

O dicionário de metadados foi incorporado ao ERWin 4.0 sendo demasiado extenso para ser descrito completamente nesta ListEx.

Uma vista parcial deste dicionário é mostrada na figura 1.

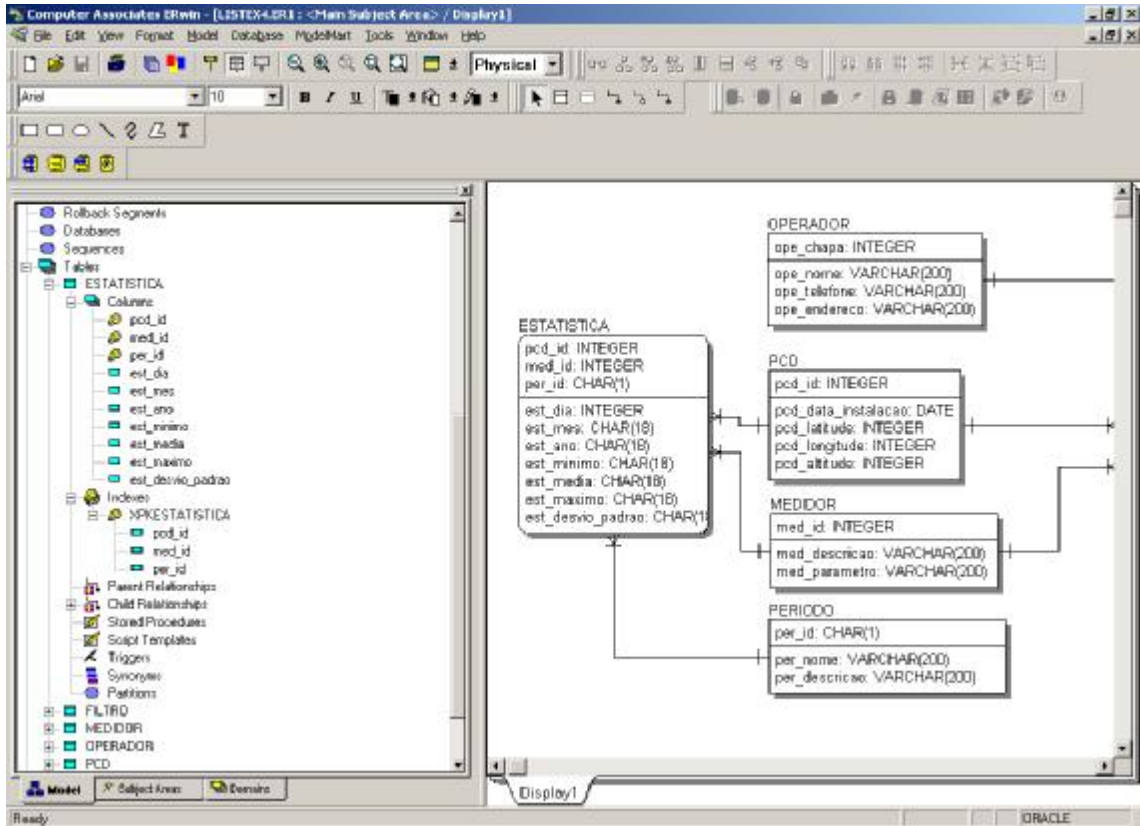


Figura 1: Visão Parcial do Dicionário de Metadados no ERWin 4.0.

2.6 Conversão para o Modelo Hierárquico

No Modelo Hierárquico o acesso a um determinado dado tem que ser precedido pelo acesso ao dado de hierarquia superior.

Deste modo, as entidades são organizadas na forma de árvores nas quais as entidades de menor hierarquia se situam nas folhas.

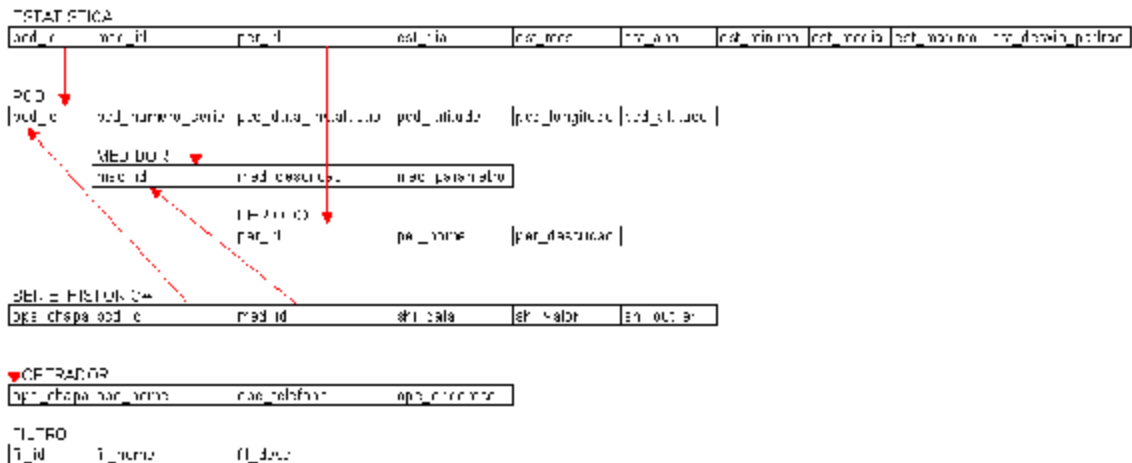


Figura 2: Conversão para o Modelo de Dados Hierárquico.

Na Figura 2, cada atributo é representada por um retângulo. Um conjunto de retângulos contíguos representa uma entidade, cujo nome é realçado em maiúsculas. As setas vermelhas ligam os registros de maior hierarquia aos registros de menor hierarquia.

2.7 Conversão para o Modelo de Rede

O Modelo de Rede é similar ao hierárquico. Porém, enquanto no modelo hierárquico as entidades são organizadas na forma de árvores, no Modelo de Rede é possível ligar uma folha com outra, criando uma “rede” de entidades.

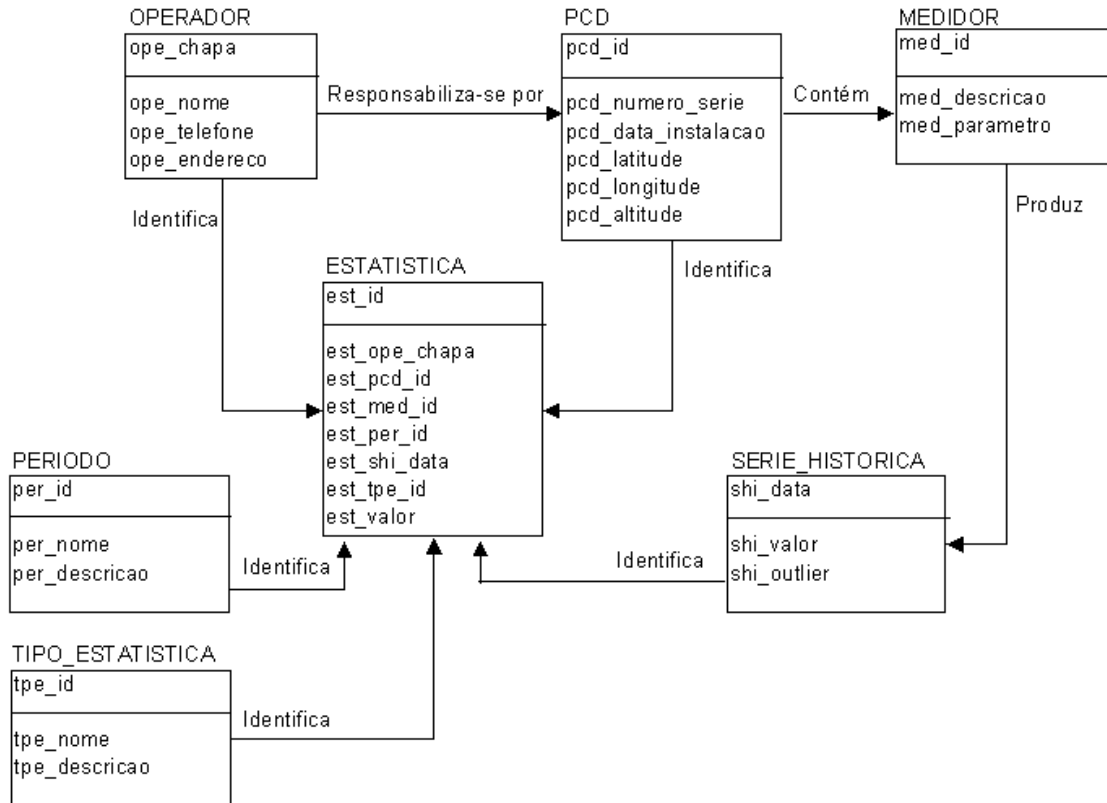


Figura 3: Conversão para o Modelo de Dados em Rede.

2.8 Conversão para o Modelo Orientado a Objetos

No Modelo Orientado a Objetos cada entidade possui, além de seus atributos, métodos que permitem inserir, remover ou alterar dados.

O objeto georeferenciado PCD beneficiou-se deste modelo, pois ganhou o método Mover que facilita a atualização da localização do PCD.

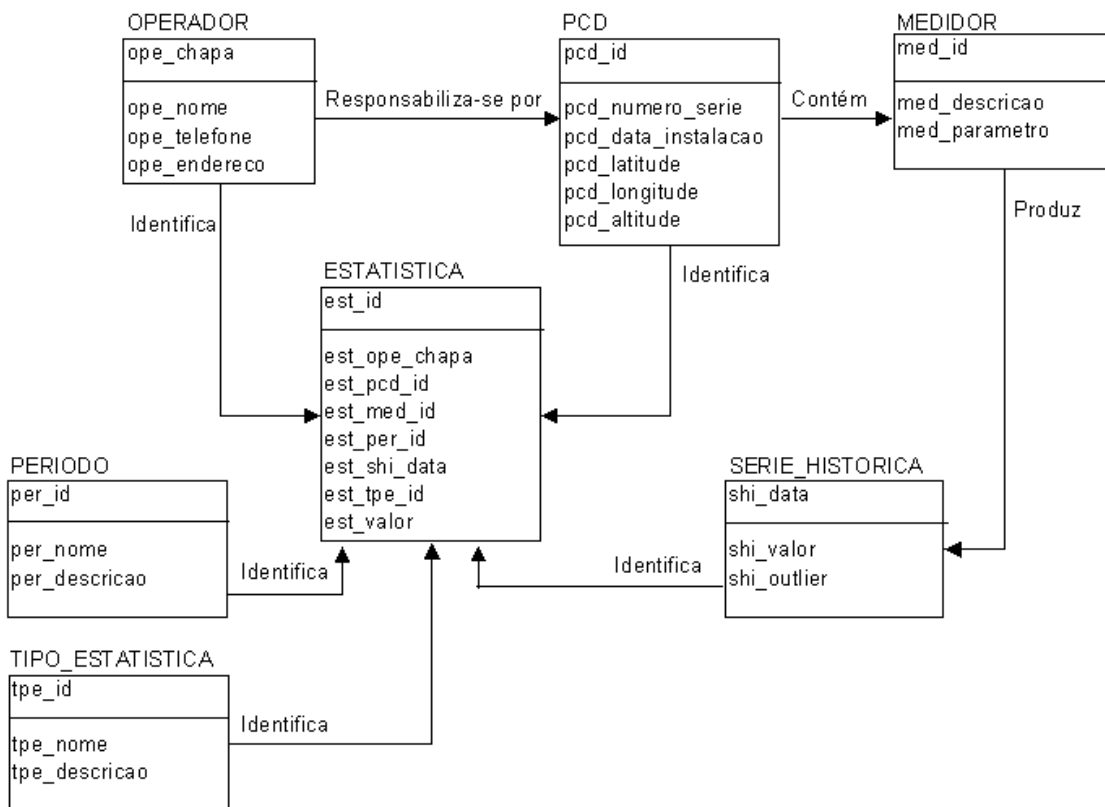


Figura 4: Conversão para o Modelo Orientado a Objetos.

3 Conclusão e Comentários

A resolução da ListEx 4 apresentou um grau de dificuldade muito maior do que as demais ListExs.

Parte da dificuldade foi devido a impossibilidade de se acessar remotamente o Oracle instalado no Laboratório da Fundação Casimiro Montenegro Filho, no ITA. Após várias tentativas infrutíferas de conexão, optamos por baixar o oracle 11g diretamente do site do fabricante.

Como o arquivo de instalação tem 1.8GB, levou-se mais de 24h para baixá-lo. A instalação e a configurações consumiu pelo menos 6h.

Embora na ListEx 3 já tivéssemos desenvolvido o modelo de dados na 3NF, a implementação das tabelas no Oracle 11g nos permitiu aperfeiçoar as tabelas, principalmente no que diz respeito aos campos NULL e NOT NULL que foram reconsiderados.

A implementação de campos georeferenciados foi muito trabalhosa pois não ficou claro a necessidade de se inserir registros na tabela user_sdo_geom_metadata para que fosse possível indexar o campo pcd_localizacao. Foram empregadas muitas horas de pesquisa para resolver este problema.

A criação do Dicionário de Dados consumiu bastante tempo. Se o banco de dados tivesse milhares de tabelas, seria inviável manter uma documentação em separado da linguagem SQL que gera as tabelas, atributos e restrições. Felizmente, na prática, podemos usar uma ferramenta como ERWin ou a própria documentação do Oracle 11g (via comentários nas tabelas e atributos).

Na conversão do modelo relacional para outros modelos, ficou claro que os modelos hierárquico e em rede são inadequados. O modelo orientado a objetos me pareceu inadequado, mas tal conclusão deve estar viesada pela maior experiência que tenho com o modelo relacional.

4 Referências

[1] CUNHA, ADILSON MARQUES DA. Notas de Aula de “CE-240 Projeto de Sistemas de Bancos de Dados” no Primeiro Período de 2009. ITA – Instituto Tecnológico de Aeronáutica. Disponível em: <http://www.comp.ita.br/~cunha>. Acessado em: 02 de maio de 2009.

[2] Oracle Corp. Oracle Database 11g. Disponível em: <http://www.oracle.com/technology/products/database/oracle11g/index.html>. 2009. Visitado em: 02 de maio de 2009.

[3] Oracle Corp. Linear Referencing Example. Disponível em: http://www.oracle.com/technology/sample_code/products/spatial/htdocs/lrs_sample/lrs_samples.html Visitado em: 17 de maio de 2009.

[4] SACRAMENTO, MARCOS COUTO. Como armazenar dados e efetuar pesquisas SQL baseadas em dados espaciais. “V Encontro Nacional de Profissionais Oracle - 2008”. Disponível em: <http://www.enpo.com.br/downloads/v-enpo/v-enpo-oracle-spatial.ppt>

5 Anexos

5.1 Instalação do Oracle 11g

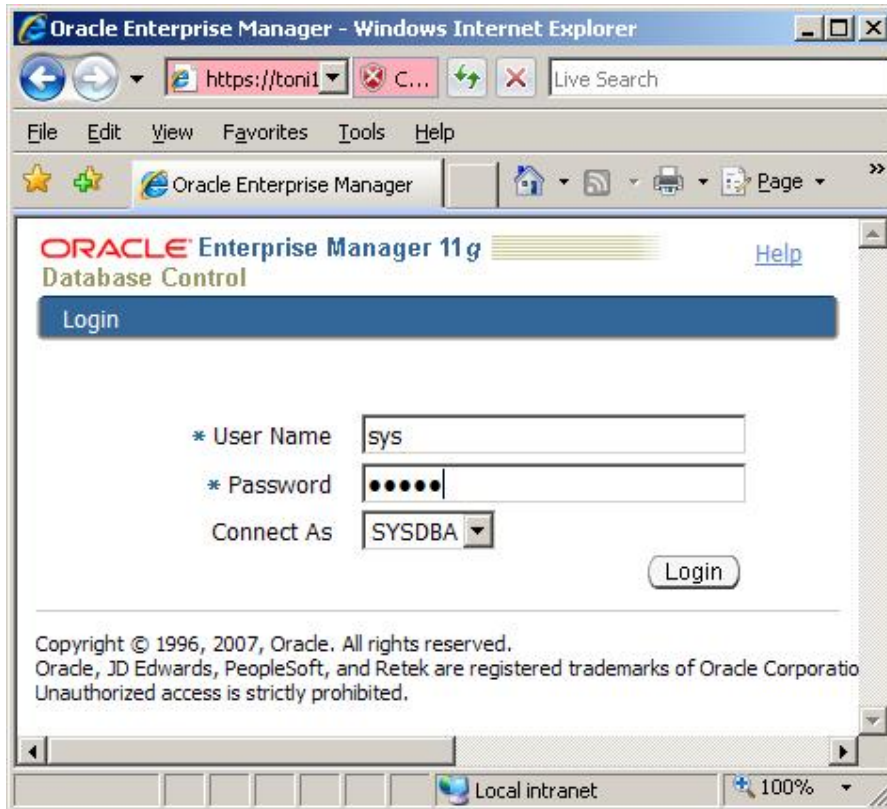


Figura 5: Tela de Login do Oracle Enterprise Manager 11g na máquina TONI10.

5.2 Modelo Físico na 3NF

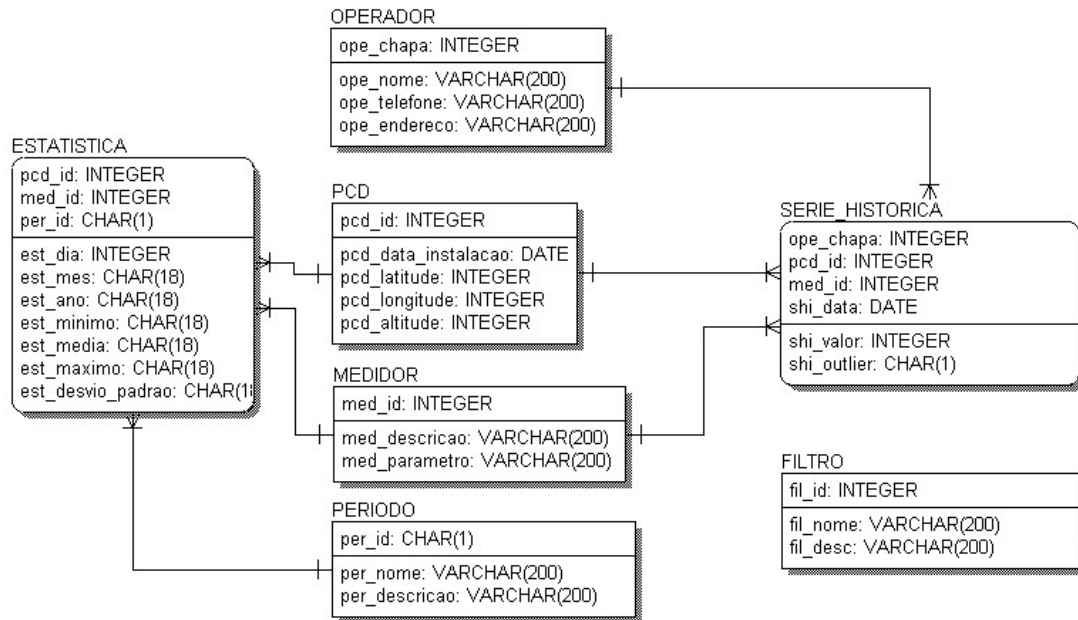


Figura 6: Modelo Físico do Módulo ADE no ERWin 4.0

5.3 Comandos para Criação de Tabelas no Oracle 11g

Estes comandos foram gerados pelo ERWin 4.0 a partir do Modelo Físico:

```
CREATE TABLE ESTATISTICA (
    pcd_id          INTEGER NOT NULL,
    med_id          INTEGER NOT NULL,
    per_id          CHAR(1) NOT NULL,
    est_dia         CHAR(2) NULL,
    est_mes         CHAR(2) NULL,
    est_ano         CHAR(4) NOT NULL,
    est_minimo      FLOAT NULL,
    est_media       FLOAT NULL,
    est_maximo      FLOAT NULL,
    est_desvio_padrao  FLOAT NULL
);

CREATE TABLE FILTRO (
    fil_id          INTEGER NOT NULL,
    fil_nome        VARCHAR(200) NOT NULL,
    fil_desc        VARCHAR(200) NOT NULL
);
```

```

CREATE TABLE MEDIDOR (
    med_id            INTEGER      NOT NULL,
    med_descricao    VARCHAR(200) NOT NULL,
    med_parametro    VARCHAR(200) NOT NULL
);

CREATE TABLE OPERADOR (
    ope_chapa        INTEGER NOT NULL,
    ope_nome         VARCHAR(200) NOT NULL,
    ope_telefone     VARCHAR(200) NULL,
    ope_endereco     VARCHAR(200) NULL
);

CREATE TABLE PCD (
    pcd_id           INTEGER NOT NULL,
    pcd_numero_serie INTEGER NOT NULL,
    pcd_data_instalacao DATE    NOT NULL,
    pcd_latitude     INTEGER NOT NULL,
    pcd_longitude    INTEGER NOT NULL,
    pcd_altitude     INTEGER NOT NULL
);

CREATE TABLE PERIODO (
    per_id           CHAR(1) NOT NULL,
    per_nome         VARCHAR(200) NOT NULL,
    per_descricao    VARCHAR(200) NOT NULL
);

CREATE TABLE SERIE_HISTORICA (
    ope_chapa        INTEGER NOT NULL,
    pcd_id           INTEGER NOT NULL,
    med_id           INTEGER NOT NULL,
    shi_data         DATE    NOT NULL,
    shi_valor        FLOAT    NULL,
    shi_outlier      CHAR(1)   NULL
);

ALTER TABLE ESTATISTICA ADD CONSTRAINT PK_ESTADISTICA PRIMARY KEY (pcd_id,
med_id, per_id);

ALTER TABLE PERIODO ADD CONSTRAINT PK_PERIODO PRIMARY KEY (per_id);

ALTER TABLE PCD ADD CONSTRAINT PK_PCD PRIMARY KEY (pcd_id);

ALTER TABLE FILTRO ADD CONSTRAINT PK_FILTRO PRIMARY KEY (fil_id);

```

```

ALTER TABLE MEDIDOR ADD CONSTRAINT PK_MEDIDOR PRIMARY KEY (med_id);

ALTER TABLE OPERADOR ADD CONSTRAINT PK_OPERADOR PRIMARY KEY (ope_chapa);

ALTER TABLE SERIE_HISTORICA ADD CONSTRAINT PK_SERIE_HISTORICA PRIMARY KEY
(ope_chapa, pcd_id, med_id, shi_data);

ALTER TABLE ESTATISTICA ADD CONSTRAINT FK_ESTADISTICA_PER_ID FOREIGN KEY
(per_id) REFERENCES PERIODO (per_id);

ALTER TABLE ESTATISTICA ADD CONSTRAINT FK_ESTADISTICA_MED_ID FOREIGN KEY
(med_id) REFERENCES MEDIDOR (med_id);

ALTER TABLE ESTATISTICA ADD CONSTRAINT FK_ESTADISTICA_PCD_ID FOREIGN KEY
(pcd_id) REFERENCES PCD (pcd_id);

ALTER TABLE SERIE_HISTORICA ADD CONSTRAINT FK_SERIE_HISTORICA_MED_ID
FOREIGN KEY (med_id) REFERENCES MEDIDOR (med_id);

ALTER TABLE SERIE_HISTORICA ADD CONSTRAINT FK_SERIE_HISTORICA_PCD_ID
FOREIGN KEY (pcd_id) REFERENCES PCD (pcd_id);

ALTER TABLE SERIE_HISTORICA ADD CONSTRAINT FK_SERIE_HISTORICA_OPE_CHAPA
FOREIGN KEY (ope_chapa) REFERENCES OPERADOR (ope_chapa);

CREATE UNIQUE INDEX IUQ_MEDIDOR on MEDIDOR(med_parametro);

CREATE UNIQUE INDEX IUQ_PERIODO on PERIODO(per_nome);

create trigger tI_ESTADISTICA after INSERT on ESTATISTICA for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- INSERT trigger on ESTATISTICA
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 09 08:37:46 2009 */
    /* PERIODO R/6 ESTATISTICA ON CHILD INSERT RESTRICT */
    select count(*) into numrows
    from PERIODO
    where
        /* :new.per_id = PERIODO.per_id */
        :new.per_id = PERIODO.per_id;
    if (
        /* */

```

```

    numrows = 0
)
then
    raise_application_error(
        -20002,
        'Cannot INSERT ESTATISTICA because PERIODO does not exist.'
    );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* MEDIDOR R/5 ESTATISTICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
    from MEDIDOR
    where
        /* :new.med_id = MEDIDOR.med_id */
        :new.med_id = MEDIDOR.med_id;
if (
    /* */

    numrows = 0
)
then
    raise_application_error(
        -20002,
        'Cannot INSERT ESTATISTICA because MEDIDOR does not exist.'
    );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/4 ESTATISTICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
    from PCD
    where
        /* :new.pcd_id = PCD.pcd_id */
        :new.pcd_id = PCD.pcd_id;
if (
    /* */

    numrows = 0
)
then
    raise_application_error(
        -20002,
        'Cannot INSERT ESTATISTICA because PCD does not exist.'
    );
end if;

```

```

        );
    end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tU_ESTADISTICA after UPDATE on ESTADISTICA for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- UPDATE trigger on ESTADISTICA
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 09 08:37:46 2009 */
    /* PERIODO R/6 ESTADISTICA ON CHILD UPDATE RESTRICT */
    select count(*) into numrows
        from PERIODO
        where
            /* :new.per_id = PERIODO.per_id */
            :new.per_id = PERIODO.per_id;
    if (
        /* */

        numrows = 0
    )
    then
        raise_application_error(
            -20007,
            'Cannot UPDATE ESTADISTICA because PERIODO does not exist.'
        );
    end if;

    /* ERwin Builtin Sun May 09 08:37:46 2009 */
    /* MEDIDOR R/5 ESTADISTICA ON CHILD UPDATE RESTRICT */
    select count(*) into numrows
        from MEDIDOR
        where
            /* :new.med_id = MEDIDOR.med_id */
            :new.med_id = MEDIDOR.med_id;
    if (
        /* */

        numrows = 0
    )
    then

```

```

        raise_application_error(
            -20007,
            'Cannot UPDATE ESTATISTICA because MEDIDOR does not exist.'
        );
    end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/4 ESTATISTICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
    from PCD
    where
        /* :new.pcd_id = PCD.pcd_id */
        :new.pcd_id = PCD.pcd_id;
if (
    /* */

    numrows = 0
)
then
    raise_application_error(
        -20007,
        'Cannot UPDATE ESTATISTICA because PCD does not exist.'
    );
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tD_MEDIDOR after DELETE on MEDIDOR for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- DELETE trigger on MEDIDOR
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 09 08:37:46 2009 */
    /* MEDIDOR R/5 ESTATISTICA ON PARENT DELETE RESTRICT */
    select count(*) into numrows
        from ESTATISTICA
        where
            /* ESTATISTICA.med_id = :old.med_id */
            ESTATISTICA.med_id = :old.med_id;
    if (numrows > 0)
    then
        raise_application_error(

```



```

        -20001,
        'Cannot DELETE MEDIDOR because ESTATISTICA exists.'
    );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* MEDIDOR R/3 SERIE_HISTORICA ON PARENT DELETE RESTRICT */
select count(*) into numrows
  from SERIE_HISTORICA
  where
    /* SERIE_HISTORICA.med_id = :old.med_id */
    SERIE_HISTORICA.med_id = :old.med_id;
if (numrows > 0)
then
  raise_application_error(
    -20001,
    'Cannot DELETE MEDIDOR because SERIE_HISTORICA exists.'
  );
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tU_MEDIDOR after UPDATE on MEDIDOR for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- UPDATE trigger on MEDIDOR
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* MEDIDOR R/5 ESTATISTICA ON PARENT UPDATE RESTRICT */
  if
    /* :old.med_id <> :new.med_id */
    :old.med_id <> :new.med_id
  then
    select count(*) into numrows
      from ESTATISTICA
      where
        /* ESTATISTICA.med_id = :old.med_id */
        ESTATISTICA.med_id = :old.med_id;
    if (numrows > 0)
    then
      raise_application_error(
        -20005,

```

```

        'Cannot UPDATE MEDIDOR because ESTATISTICA exists.'
    );
end if;
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* MEDIDOR R/3 SERIE_HISTORICA ON PARENT UPDATE RESTRICT */
if
    /* :old.med_id <> :new.med_id */
    :old.med_id <> :new.med_id
then
    select count(*) into numrows
        from SERIE_HISTORICA
        where
            /* SERIE_HISTORICA.med_id = :old.med_id */
            SERIE_HISTORICA.med_id = :old.med_id;
    if (numrows > 0)
    then
        raise_application_error(
            -20005,
            'Cannot UPDATE MEDIDOR because SERIE_HISTORICA exists.'
        );
    end if;
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tD_OPERADOR after DELETE on OPERADOR for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- DELETE trigger on OPERADOR
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 09 08:37:46 2009 */
    /* OPERADOR R/1 SERIE_HISTORICA ON PARENT DELETE RESTRICT */
    select count(*) into numrows
        from SERIE_HISTORICA
        where
            /* SERIE_HISTORICA.ope_chapa = :old.ope_chapa */
            SERIE_HISTORICA.ope_chapa = :old.ope_chapa;
    if (numrows > 0)
    then
        raise_application_error(

```

```

        -20001,
        'Cannot DELETE OPERADOR because SERIE_HISTORICA exists.'
    );
end if;

-- ERwin Bultin Sun May 09 08:37:46 2009
end;
/

create trigger tU_OPERADOR after UPDATE on OPERADOR for each row
-- ERwin Bultin Sun May 09 08:37:46 2009
-- UPDATE trigger on OPERADOR
declare numrows INTEGER;
begin
    /* ERwin Bultin Sun May 09 08:37:46 2009 */
    /* OPERADOR R/1 SERIE_HISTORICA ON PARENT UPDATE RESTRICT */
    if
        /* :old.ope_chapa <> :new.ope_chapa */
        :old.ope_chapa <> :new.ope_chapa
    then
        select count(*) into numrows
            from SERIE_HISTORICA
            where
                /* SERIE_HISTORICA.ope_chapa = :old.ope_chapa */
                SERIE_HISTORICA.ope_chapa = :old.ope_chapa;
        if (numrows > 0)
        then
            raise_application_error(
                -20005,
                'Cannot UPDATE OPERADOR because SERIE_HISTORICA exists.'
            );
        end if;
    end if;
end if;

-- ERwin Bultin Sun May 09 08:37:46 2009
end;
/

create trigger tD_PCD after DELETE on PCD for each row
-- ERwin Bultin Sun May 09 08:37:46 2009
-- DELETE trigger on PCD
declare numrows INTEGER;
begin

```

```

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/4 ESTATISTICA ON PARENT DELETE RESTRICT */
select count(*) into numrows
  from ESTATISTICA
  where
    /* ESTATISTICA.pcd_id = :old.pcd_id */
    ESTATISTICA.pcd_id = :old.pcd_id;
if (numrows > 0)
then
  raise_application_error(
    -20001,
    'Cannot DELETE PCD because ESTATISTICA exists.'
  );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/2 SERIE_HISTORICA ON PARENT DELETE RESTRICT */
select count(*) into numrows
  from SERIE_HISTORICA
  where
    /* SERIE_HISTORICA.pcd_id = :old.pcd_id */
    SERIE_HISTORICA.pcd_id = :old.pcd_id;
if (numrows > 0)
then
  raise_application_error(
    -20001,
    'Cannot DELETE PCD because SERIE_HISTORICA exists.'
  );
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tU_PCD after UPDATE on PCD for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- UPDATE trigger on PCD
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* PCD R/4 ESTATISTICA ON PARENT UPDATE RESTRICT */
  if
    /* :old.pcd_id <> :new.pcd_id */
    :old.pcd_id <> :new.pcd_id

```

```

then
  select count(*) into numrows
    from ESTATISTICA
    where
      /* ESTATISTICA.pcd_id = :old.pcd_id */
      ESTATISTICA.pcd_id = :old.pcd_id;
  if (numrows > 0)
  then
    raise_application_error(
      -20005,
      'Cannot UPDATE PCD because ESTATISTICA exists.'
    );
  end if;
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/2 SERIE_HISTORICA ON PARENT UPDATE RESTRICT */
if
  /* :old.pcd_id <> :new.pcd_id */
  :old.pcd_id <> :new.pcd_id
then
  select count(*) into numrows
    from SERIE_HISTORICA
    where
      /* SERIE_HISTORICA.pcd_id = :old.pcd_id */
      SERIE_HISTORICA.pcd_id = :old.pcd_id;
  if (numrows > 0)
  then
    raise_application_error(
      -20005,
      'Cannot UPDATE PCD because SERIE_HISTORICA exists.'
    );
  end if;
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tD_PERIODO after DELETE on PERIODO for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- DELETE trigger on PERIODO
declare numrows INTEGER;
begin

```

```

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PERIODO R/6 ESTATISTICA ON PARENT DELETE RESTRICT */
select count(*) into numrows
  from ESTATISTICA
  where
    /* ESTATISTICA.per_id = :old.per_id */
    ESTATISTICA.per_id = :old.per_id;
if (numrows > 0)
then
  raise_application_error(
    -20001,
    'Cannot DELETE PERIODO because ESTATISTICA exists.'
  );
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tU_PERIODO after UPDATE on PERIODO for each row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- UPDATE trigger on PERIODO
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* PERIODO R/6 ESTATISTICA ON PARENT UPDATE RESTRICT */
  if
    /* :old.per_id <> :new.per_id */
    :old.per_id <> :new.per_id
  then
    select count(*) into numrows
      from ESTATISTICA
      where
        /* ESTATISTICA.per_id = :old.per_id */
        ESTATISTICA.per_id = :old.per_id;
    if (numrows > 0)
    then
      raise_application_error(
        -20005,
        'Cannot UPDATE PERIODO because ESTATISTICA exists.'
      );
    end if;
  end if;
end if;

```

```

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tI_SERIE_HISTORICA after INSERT on SERIE_HISTORICA for each
row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- INSERT trigger on SERIE_HISTORICA
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* MEDIDOR R/3 SERIE_HISTORICA ON CHILD INSERT RESTRICT */
  select count(*) into numrows
    from MEDIDOR
   where
      /* :new.med_id = MEDIDOR.med_id */
      :new.med_id = MEDIDOR.med_id;
  if (
    /* */

    numrows = 0
  )
  then
    raise_application_error(
      -20002,
      'Cannot INSERT SERIE_HISTORICA because MEDIDOR does not exist.'
    );
  end if;

  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* PCD R/2 SERIE_HISTORICA ON CHILD INSERT RESTRICT */
  select count(*) into numrows
    from PCD
   where
      /* :new.pcd_id = PCD.pcd_id */
      :new.pcd_id = PCD.pcd_id;
  if (
    /* */

    numrows = 0
  )
  then
    raise_application_error(
      -20002,

```

```

        'Cannot INSERT SERIE_HISTORICA because PCD does not exist.'
    );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* OPERADOR R/1 SERIE_HISTORICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
  from OPERADOR
  where
    /* :new.ope_chapa = OPERADOR.ope_chapa */
    :new.ope_chapa = OPERADOR.ope_chapa;
if (
  /* */

  numrows = 0
)
then
  raise_application_error(
    -20002,
    'Cannot INSERT SERIE_HISTORICA because OPERADOR does not exist.'
  );
end if;

-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/

create trigger tU_SERIE_HISTORICA after UPDATE on SERIE_HISTORICA for each
row
-- ERwin Builtin Sun May 09 08:37:46 2009
-- UPDATE trigger on SERIE_HISTORICA
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 09 08:37:46 2009 */
  /* MEDIDOR R/3 SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
  select count(*) into numrows
    from MEDIDOR
    where
      /* :new.med_id = MEDIDOR.med_id */
      :new.med_id = MEDIDOR.med_id;
  if (
    /* */

    numrows = 0

```



```

)
then
  raise_application_error(
    -20007,
    'Cannot UPDATE SERIE_HISTORICA because MEDIDOR does not exist.'
  );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* PCD R/2 SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
  from PCD
  where
    /* :new.pcd_id = PCD.pcd_id */
    :new.pcd_id = PCD.pcd_id;
if (
  /* */

  numrows = 0
)
then
  raise_application_error(
    -20007,
    'Cannot UPDATE SERIE_HISTORICA because PCD does not exist.'
  );
end if;

/* ERwin Builtin Sun May 09 08:37:46 2009 */
/* OPERADOR R/1 SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
  from OPERADOR
  where
    /* :new.ope_chapa = OPERADOR.ope_chapa */
    :new.ope_chapa = OPERADOR.ope_chapa;
if (
  /* */

  numrows = 0
)
then
  raise_application_error(
    -20007,
    'Cannot UPDATE SERIE_HISTORICA because OPERADOR does not exist.'
  );
end if;

```

```
-- ERwin Builtin Sun May 09 08:37:46 2009
end;
/
```

5.4 Comandos SQL para Alterar a Tabela PCD

Foi necessário alterar a tabela PCD substituindo os campos pcd_altitude, pcd_longitude e pcd_altitude do modelo definido na ListEx 3, pelo campo pcd_localizacao que é georeferenciado.

Foi inserido um registro na tabela user_sdo_geom_metada para permitir indexar espacialmente o campo pcd_localizacao[3].

```
alter table PCD
    drop column pcd_altitude;

alter table PCD
    drop column pcd_longitude;

alter table PCD
    drop column pcd_latitude;

alter table PCD
    add pcd_localizacao MDSYS.SDO_GEOMETRY;

create unique index iuq_pcd on pcd(pcd_numero_serie,pcd_data_instalacao);
insert into user_sdo_geom_metadata (table_name,column_name,diminfo,srid)
values
('PCD', 'PCD_LOCALIZACAO',mdsys.sdo_dim_array(
    mdsys.sdo_dim_element('LONG', -180.0, 180.0, 0.005),
    mdsys.sdo_dim_element('LAT' , -90.0, 90.0, 0.005),
    mdsys.sdo_dim_element('ALT' , -100,+10000, 1.0)
),
8292);

create index idx_pcd_spatial on pcd(pcd_localizacao)
indextype is mdsys.spatial_index
parameters ('sdo_indx_dims=2,layer_gtype=point');
```

5.5 Exemplos de Dados

As tabelas a seguir exibem os exemplos de dados utilizados para “massa de testes” do módulo AFE no Oracle 11g.

Tabela 1: Exemplo de dados para a Tabela OPERADOR.

ope_chapa	ope_nome	ope_telefone	ope_endereco
1	José Silva	55 92 1234 5678	Igarapé, 5 – Lagoa – Manaus, AM
2	Maria Silva	55 68 4321 8765	Rua Riozinho, 123 – Bairro das Enchentes – Bujari, AC
3	Cacique Juruna		Tribo dos Piraraquaras, Alto Xingu, PA

Em nosso modelo, o endereço do operador é uma cadeia de caracteres suficientemente longa para incluir logradouro, número, casa, cidade, estado, cep etc. Num banco de dados profissional, este campo seria substituído por ligações para diversas outras tabelas que validariam estes dados permitindo, por exemplo, que ao digitar o CEP, o usuário já tivesse os valores de logradouro, cidade e estado.

Tabela 2: Exemplo de dados para a Tabela PCD.

pcd_id	pcd_serial_number	pcd_data_inst	pcd_latitude	pcd_longitude	pcd_altitude
1	10	01/01/2000	-22,6625	-43,9567	370
2	10	01/01/2001	-22,5389	-43,7775	389
3	20	01/01/2000	-22,7489	-44,1247	440

Em nosso modelo, um mesmo PCD pode estar em locais diferentes desde que em datas diferentes. É o que acontece, por exemplo, com o PCD cujo número de série é 10 o qual estave em um determinado local em 01/01/2000 e em outro local em 01/01/2001.

Tabela 3: Exemplo de dados para a Tabela MEDIDOR.

med_id	med_descricao	med_parametro
1	XPTO1	Altura pluviométrica (mm)
2	XPTO2	Temperatura (Celsius)
3	XPTO3	Lâmina D' Água (m)

Em nosso modelo, consideramos apenas 3 tipos de medidores. Na prática, haveria dezenas de tipos de medidores sendo talvez necessário especificar melhor cada medidor, armazenando, por exemplo: erro médio esperado, dados do fabricante, dados de calibração etc.

Tabela 4: Exemplo de dados para a Tabela SERIE_HISTORICA.

ope_chapa	pcd_id	med_id	shi_data	shi_valor	shi_outlier
1	1	1	01/01/2000 10:30:35	34	F
1	1	1	01/01/2000 10:32:35	35	F
1	1	1	01/01/2000 10:34:40	42	T
2	2	1	01/01/2000 10:30:35	34	F
2	2	2	01/01/2000 10:30:35	50	F
2	2	3	01/01/2000 10:30:35	1,25	F
3	2	3	01/02/2000 12:30:35	1,35	F

Nota-se que o terceiro registro deste exemplo é um outlier pois seu valor está completamente fora do esperado, sendo pouco provável que a temperatura aumentasse 7 °C em apenas 2min.

Tabela 5: Exemplo de dados para a Tabela PERIODO.

per_id	per_nome	per_descricao
D	Dia	Cálculos estatísticos considerando períodos de 1 dia.
M	Mês	Cálculos estatísticos considerando períodos de 1 mês.
A	Ano	Cálculos estatísticos considerando períodos de 1 ano.

Tabela 6: Exemplo de dados para a Tabela ESTATISTICA.

pcd_id	med_id	per_id	est_dia	est_mes	est_ano	est_minimo	est_media	est_maximo	est_c
1	1	D	01	01	2000	34,0	34,5	35,0	1,0
1	1	M	NULL	01	2000	34,0	34,5	35,0	1,0
1	1	A	NULL	NULL	2000	34,0	34,5	35,0	1,0
2	2	D	01	01	2000	34,0	34,0	34,0	0
2	2	M	NULL	01	2000	34,0	34,0	34,0	0
2	2	A	NULL	NULL	2000	34,0	34,0	34,0	0

Tabela 7: Exemplo de dados para a Tabela FILTRO.

fil_id	fil_nome	fil_descricao
1	Média Móvel	Retorna o valor filtrado como sendo a média aritmética dos N valores anteriormente lidos numa série histórica.
2	Outlier	Preenche o campo outlier para uma determinada série histórica.
3	Mínimos	Retorna o valor filtrado como sendo o valor mínimo dos N valores anteriormente lidos numa série histórica. Valores com outlier=True são ignorados.
4	Máximos	Retorna o valor filtrado como sendo o valor máximo dos N valores anteriormente lidos numa série histórica. Valores com outlier=True são ignorados.

5.6 Comandos SQL para Inserir Exemplos de Dados

```

/*****
/* OPERADOR */
*****/
insert into operador (ope_chapa,ope_nome,ope_telefone,ope_endereco) values
(1,'José Silva','55 92 1234 5678','Igarapé, 5 - Lagoa - Manaus, AM');

insert into operador (ope_chapa,ope_nome,ope_telefone,ope_endereco) values
(2,'Maria Silva','55 68 4321 8765','Rua Riozinho, 123 - Bairro das
Enchentes - Bujari, AC');

insert into operador (ope_chapa,ope_nome,ope_telefone,ope_endereco) values
(3,'Cacique Juruna',NULL,'Tribo dos Piraraquaras, Alto Xingu, PA');
commit;

/*****
/* PCD */
*****/

insert into
pcd(pcd_id,pcd_numero_serie,pcd_data_instalacao,pcd_localizacao) values
(
  1,
  10,
  to_date('2000-01-01','yyyy-mm-dd'),
  mdsys.sdo_geometry(3001,8292,mdsys.sdo_point_type(-43.9567, -22.6625,
370),null,null)
);

insert into
pcd(pcd_id,pcd_numero_serie,pcd_data_instalacao,pcd_localizacao) values
(
  2,
  10,
  to_date('2001-01-01','yyyy-mm-dd'),
  mdsys.sdo_geometry(3001,8292,mdsys.sdo_point_type(-43.7775, -22.5389,
389),null,null)
);

```

```

insert into
pcd(pcd_id,pcd_numero_serie,pcd_data_instalacao,pcd_localizacao) values
(
  3,
  20,
  to_date('2000-01-01','yyyy-mm-dd'),
  mdsys.sdo_geometry(3001,8292,mdsys.sdo_point_type(-44.1247, -22.7489,
440),null,null)
);
commit;

```

```

/*****/
/* MEDIDOR */
/*****/

```

```

insert into medidor(med_id,med_descricao,med_parametro) values
(1,'XPTO 1','Altura Pluviométrica (mm)');
insert into medidor(med_id,med_descricao,med_parametro) values
(2,'XPTO 2','Temperatura (Celsius)');
insert into medidor(med_id,med_descricao,med_parametro) values
(3,'XPTO 3','Lâmina D' 'Água (m)');
commit;

```

```

/*****/
/* PERIODO */
/*****/

```

```

insert into periodo(per_id,per_nome,per_descricao) values
('D','Dia','Cálculos estatísticos considerando períodos de 1 dia. ');
insert into periodo(per_id,per_nome,per_descricao) values
('M','Mês','Cálculos estatísticos considerando períodos de 1 mês. ');
insert into periodo(per_id,per_nome,per_descricao) values
('A','Ano','Cálculos estatísticos considerando períodos de 1 ano. ');
commit;

```

```

/*****/
/* FILTRO */
/*****/

```

```

insert into filtro(fil_id,fil_nome,fil_desc) values
(1,'Média Móvel','Retorna o valor filtrado como sendo a média aritmética
dos N valores anteriormente lidos numa série histórica. ');
insert into filtro(fil_id,fil_nome,fil_desc) values
(2,'Outlier','Preenche o campo outlier para uma determinada série
histórica. ');
insert into filtro(fil_id,fil_nome,fil_desc) values
(3,'Mínimos','Retorna o valor filtrado como sendo o valor mínimo dos N
valores anteriormente lidos numa série histórica. Valores com outlier=True
são ignorados. ');
insert into filtro(fil_id,fil_nome,fil_desc) values
(4,'Máximos','Retorna o valor filtrado como sendo o valor máximo dos N
valores anteriormente lidos numa série histórica. Valores com outlier=True
são ignorados. ');
commit;

```

```

/*****/
/* SERIE_HISTORICA */
/*****/

```

```

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(1,1,1,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),34,'F');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(1,1,1,to_date('01/01/2000 10:32:35','DD/MM/YYYY HH24:MI:SS'),35,'F');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(1,1,1,to_date('01/01/2000 10:34:40','DD/MM/YYYY HH24:MI:SS'),42,'T');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(2,2,1,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),34,'F');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(2,2,2,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),50,'F');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(2,2,3,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),1.25,'F');

insert into
serie_historica(ope_chapa,pcd_id,med_id,shi_data,shi_valor,shi_outlier)
values
(2,2,3,to_date('01/01/2000 12:30:35','DD/MM/YYYY HH24:MI:SS'),1.35,'F');

commit;

/*****
/* ESTATISTICA */
*****/

insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(1,1,'D',01,01,2000,34.0,34.5,35.0,1.0);

insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(1,1,'M',NULL,01,2000,34.0,34.5,35.0,1.0);

insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(1,1,'A',NULL,NULL,2000,34.0,34.5,35.0,1.0);

insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(2,2,'D',01,01,2000,34.0,34.0,34.0,0);

```

```
insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(2,2,'M',NULL,01,2000,34.0,34.0,34.0,0);

insert into
estatistica(pcd_id,med_id,per_id,est_dia,est_mes,est_ano,est_minimo,est_med
ia,est_maximo,est_desvio_padrao) values
(2,2,'A',NULL,NULL,2000,34.0,34.0,34.0,0);

commit;
```