



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

Versão 2 - 25 de maio de 2009

Sumário

1	Introdução	1
1.1	Título	1
1.2	Motivação	1
1.3	Objetivos.....	1
2	Conteúdo.....	2
2.1	Instalação do Oracle 11g	2
2.2	Implementação do Módulo AFE.....	2
2.3	Georeferenciamento da Tabela T_PCD	2
2.4	Inserção de Dados no Módulo AFE.....	2
2.5	Realização de Queries no Banco de Dados	3
2.5.1	Query com Uma Tabela.....	3
2.5.2	Query com Duas Tabelas	3
2.5.3	Query com Três Tabelas	3
2.5.4	Query Georeferenciada	4
2.6	Dicionário de Dados.....	4
2.6.1	Dicionário de Dados Propriamente Dito.....	4
2.6.1.1	Tabela T_ESTATISTICA	5
2.6.1.2	Tabela T_FILTRO	7
2.6.1.3	Tabela T_MEDIDOR.....	7
2.6.1.4	Tabela T_OPERADOR	8
2.6.1.5	Tabela T_PCD	8
2.6.1.6	Tabela T_PERIODO	9
2.6.1.7	Tabela T_SERIE_HISTORICA.....	10
2.6.2	Diretório de Dados	11
2.6.2.1	Peopleware:	11
2.6.2.2	Software:.....	12
2.6.3	Dicionário de Recursos de Dados	12
2.6.4	Dicionário de Metadados.....	12
2.7	Conversão para o Modelo Hierárquico	13
2.8	Conversão para o Modelo de Rede	14
2.9	Conversão para o Modelo Orientado a Objetos.....	15
2.10	Definição de Qual Modelo Utilizar.....	15
3	Conclusão e Comentários.....	16
4	Referências	16
5	Anexos.....	18
5.1	Instalação do Oracle 11g	18
5.2	Modelo Físico na 3NF da ListEx 3	19
5.3	Modelo Físico na 3NF da ListEx 4	19

5.4	Comandos para Criação de Tabelas no Oracle 11g	20
5.5	Georeferenciamento da Tabela T_PCD	36
5.6	Exemplos de Dados na ListEx 3	37
5.7	Comandos SQL para Inserir Exemplos de Dados	39

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 realizar, na prática, a implementação das tabelas definidas de modo teórico na ListEx 3. É 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 Objetivos

- Implementar, em Oracle 11g, 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.
- Comparar a implementação em modelo relacional com possíveis implementações em modelos rede, hierárquico e orientado a objeto visando identificar vantagens e desvantagens de cada modelo.

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. 18).

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. 19).

Foi feita a revisão completa do Modelo Físico apresentado na ListEx 3, alterando-se a nomenclatura das entidades e dos atributos de acordo com as “Regras e Padrões para Nomenclatura de Objetos” [2]. O resultado é mostrado no Anexo I, Figura 7, Pág. 19.

A partir deste modelo, utilizou-se o ERWin para gerar comandos em SQL para criar tabelas no Oracle 11g. Em seguida, o código foi revisto e comentado, gerando o arquivo `listex4.SQL` cujo conteúdo é mostrado no Anexo I, Seção 5.4, Pág. 19.

Para criar as tabelas, o arquivo `listex4.SQL` foi executado no aplicativo SQL*Developer¹ 1.1.3 Build MAIN-27.69 do Oracle 11g.

2.3 Georeferenciamento da Tabela T_PCD

Para permitir consultas georeferenciadas na tabela T_PCD foi necessário incluir um registro na tabela de sistema `user_sdo_geom_metadata` e criar um índice espacial.

Os comandos que executam estas alterações estão no arquivo `listex4_pcd.sql` cujo conteúdo é mostrado no Anexo I, Seção 5.5, Pág. 36.

2.4 Inserção de Dados no Módulo AFE

Como “massa de dados para teste” foram usados os dados de exemplo da ListEx 3, que é reproduzida no Anexo I, Seção 5.6. Pág. 37.

¹ Para mais informações sobre o SQL*Developer, consultar:
http://www.oracle.com/technology/products/database/sql_developer/index.html

Os comandos SQL para inserir dados foram digitados no script `listex4_dados.SQL` e estão reproduzidos no Anexo I, Seção 5.7. Pág. 39.

2.5 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.

2.5.1 Query com Uma Tabela

Listar nome e telefone dos operadores.

```
SELECT ope_nm_nome as Nome,  
       ope_tx_telefone as Telefone  
FROM t_operador  
ORDER BY 1,2;
```

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

2.5.2 Query com Duas Tabelas

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

```
SELECT DISTINCT ope_nm_nome as Nome,  
               ope_tx_telefone as Telefone  
FROM t_operador o  
JOIN t_serie_historica sh on sh.shi_ope_mt_chapa = o.ope_mt_chapa  
WHERE shi_il_outlier = 'T'  
ORDER BY 1,2;
```

Nome	Telefone
José Silva	55 92 1234 5678

2.5.3 Query com Três Tabelas

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

```
SELECT ope_nm_nome as Nome,  
       ope_tx_telefone as Telefone,  
       med_nm_parametro as Parametro,  
       shi_dt_data as Data,  
       shi_md_valor as Valor  
FROM t_operador o  
JOIN t_serie_historica sh on sh.shi_ope_mt_chapa = o.ope_mt_chapa  
JOIN t_medidor m on m.med_id = sh.shi_med_id  
WHERE shi_il_outlier = 'T'  
ORDER BY 1,2,3,4;
```

Nome	Telefone	Parametro	Data	Valor
José Silva	55 92 1234 5678	ALTPLU	01-JAN-00	42

2.5.4 Query Georeferenciada

Listar a identificação do PCD, o nome do respectivo operador e a distância, em metros, do respectivo PCD até o PCD cujo identificador é 1.

```
select DISTINCT
    p1.pcd_id pcd,
    ope_nm_nome operador,
MDSYS.SDO_GEOM.SDO_DISTANCE(p1.pcd_cg_localizacao,p2.pcd_cg_localizacao,0.0
05) distancia
from t_operador o
join t_serie_historica sh on sh.shi_ope_mt_chapa = o.ope_mt_chapa
join t_pcd p1 on p1.pcd_id = sh.shi_pcd_id,
    t_pcd p2
where p2.pcd_id=1;
```

pcd	operador	distancia
1	José Silva	0
2	Maria Silva	22953.1902551594

2.6 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.

2.6.1 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 **INTEGRIDADE** 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.

Nas tabelas do dicionário considerou-se:

1. A coluna “Nulos” está preenchida com **Sim** quando o campo pode conter valores nulos e com **Não** quando o campo não pode conter valores nulos.
2. A coluna “PK” (de *Primary Key* ou Chave Primária) está preenchido com **Sim** quando o campo faz parte da chave primária da tabela e com **Não** quando o campo não faz parte da chave primária da tabela.
3. A coluna “FK” (de *Foreign Key* ou Chave Estrangeira) está preenchido com **Sim** quando o campo é uma chave estrangeira e com **Não** quando o campo não é chave estrangeira.

O relacionamento entre as entidade pode ser visto no Anexo I, Figura 6, Pág. 19.

2.6.1.1 Tabela T_ESTATISTICA

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

2.6.1.1.1 Atributos da Tabela T_ESTATISTICA

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
EST_PCD_ID	INTEGER	38	Não	Sim	Sim	Identificador do PCD
EST_MED_ID	INTEGER	38	Não	Sim	Sim	Identificador do MEDIDOR
EST_PER_ID	CHAR	1	Não	Sim	Sim	Identificador do PERIODO
EST_DD_DIA	CHAR	2	Não	Sim	Não	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. Valores válidos 01 a 31 (sempre com dois dígitos). Valor máximo de acordo com o valor de est_mes.

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
EST_MM_MES	CHAR	2	Sim	Não	Não	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. Valores válidos 01 a 12 (sempre com dois dígitos).
EST_AA_ANO	CHAR	4	Não	Não	Não	Ano ao qual se refere o PERIODO em que a estatística foi efetuada. Valores válidos 1900 a 2050 (sempre com quatro dígitos).
EST_NU_MINIMO	FLOAT	126	Sim	Não	Não	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. Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).
EST_NU_MEDIA	FLOAT	126	Sim	Não	Não	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. Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no período considerado).
EST_NU_MAXIMO	FLOAT	126	Sim	Não	Não	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. Preenchido com NULL caso não possa ser calculado (devido, por exemplo, à ausência de amostras válidas no

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
						período considerado.
EST_NU_DESVIO_PADRAO	FLOAT	126	Sim	Não	Não	Desvio-padrão da medida med_id do PCD pcd_id no período per_id e na data 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.

2.6.1.1.2 Restrições da Tabela T_ESTATISTICA

Nome	Tipo	Campos
PK_T_ESTATISTICA	PK	EST_PCD_ID, EST_MED_ID, EST_PER_ID
FK_EST_PER	FK PERIODO (PER_ID)	EST_PER_ID
FK_EST_MED	FK MEDIDA (MED_ID)	EST_MED_ID
FK_EST_PCD	FK PCD (PCD_ID)	EST_PCD_ID

2.6.1.2 Tabela T_FILTRO

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

2.6.1.2.1 Atributos da Tabela T_FILTRO

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
FIL_ID	INTEGER	38	Não	Sim	Não	Identificador do Filtro
FIL_NM_NOME	VARCHAR	200	Não	Não	Não	Nome do Filtro
FIL_TX_DESCRICA0	VARCHAR	200	Não	Não	Não	Descrição do Filtro.

2.6.1.2.2 Restrições da Tabela T_FILTRO

Nome	Tipo	Campos
PK_T_FILTRO	PK	FIL_ID

2.6.1.3 Tabela T_MEDIDOR

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

A leitura destes medidores é armazenada na tabela T_SERIE_HISTORICA.

Estatísticas dos dados destes medidores são armazenadas na tabela T_ESTATISTICA.

2.6.1.3.1 Atributos da Tabela T_MEDIDOR

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
MED_ID	INTEGER	38	Não	Sim	Não	Identificador do Medidor
MED_NM_PARAMETRO	VARCHAR	200	Não	Não	Não	Nome do parâmetro medido.
MED_TX_DESCRICAÇÃO	VARCHAR	200	Não	Não	Não	Descrição do parâmetro medido.

2.6.1.3.2 Restrições da Tabela T_MEDIDOR

Nome	Tipo	Campos
IUQ_MEDIDOR	Índice único	MED_NM_PARAMETRO
PK_MEDIDOR	PK	MED_ID

2.6.1.4 Tabela T_OPERADOR

A tabela T_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.

2.6.1.4.1 Atributos da Tabela T_OPERADOR

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
OPE_MT_CHAPA	INTEGER	38	Não	Sim	Não	Chapa do operador.
OPE_NM_NOME	VARCHAR	200	Não	Não	Não	Nome do operador
OPE_TX_TELEFONE	VARCHAR	200	Sim	Não	Não	Telefone do operador.
OPE_TX_ENDERECO	VARCHAR	200	Sim	Não	Não	Endereço do operador.

2.6.1.4.2 Restrições da Tabela T_OPERADOR

Nome	Tipo	Campos
PK_T_OPERADOR	PK	OPE_MT_CHAPA

2.6.1.5 Tabela T_PCD

A tabela georeferenciada T_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.

2.6.1.5.1 Atributos da Tabela T_PCD

Atributo	Formato	Tamanho	Nulo	PK	FK	Descrição
PCD_ID	INTEGER	38	Não	Sim	Não	Identificador do PCD
PCD_NM_NUMERO_SERIE	INTEGER	38	Não	Não	Não	Número de Série do PCD
PCD_DT_INSTALACAO	DATE	16	Não	Não	Não	Data da instalação do PCD na localização dada por pcd_localizacao.
PCD_CG_LOCALIZACAO	MDSYS.SDO_GEOMETRY	200	Não	Não	Não	Localização do PCD (latitude, longitude e altitude).

2.6.1.5.2 Restrições da Tabela T_PCD

Nome	Tipo	Campos
IDX_T_PCD_SPATIAL	Índice Espacial	PCD_CG_LOCALIZACAO
IUQ_T_PCD	Índice único	PCD_NM_NUMERO_SERIE, PCD_DT_INSTALACAO
PK_T_PCD	PK	PCD_ID

2.6.1.6 Tabela T_PERIODO

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

2.6.1.6.1 Atributos da Tabela T_PERIODO

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
PER_ID	CHAR	1	Não	Sim	Não	Identificador do Período. Exemplo: M para mês, D para dia, A para ano.
PER_NM_NOME	VARCHAR	200	Não	Não	Não	Nome do Período. Exemplo: Mês, Dia ou Ano.
PER_TX_DESCRICA0	VARCHAR	200	Não	Não	Não	Descrição do Período.

2.6.1.6.2 Restrições da Tabela T_PERIODO

Nome	Tipo	Campos
IUQ_T_PERIODO	Índice único	PER_NM_NOME
PK_T_PERIODO	PK	PER_ID

2.6.1.7 Tabela T_SERIE_HISTORICA

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

2.6.1.7.1 Atributos da Tabela T_SERIE_HISTORICA

Atributo	Formato	Tamanho	Nulos	PK	FK	Descrição
SHI_OPE_MT_CHAPA	INTEGER	38	Não	Sim	Sim	Chapa do operador que fez a leitura.
SHI_PCD_ID	INTEGER	38	Não	Sim	Sim	Identificador do PCD no qual foi feita a leitura.
SHI_MED_ID	INTEGER	38	Não	Sim	Sim	Identificador do MEDIDOR que forneceu os dados lidos.
SHI_DT_DATA	DATE	18	Não	Sim	Não	Data da leitura.
SHI_MD_VALOR	FLOAT	126	Não	Não	Não	Valor lido no MEDIDOR do PCD pelo OPERADOR na data shi_data. Se NULL significa que o dado estava indisponível.
SHI_IL_OUTLIER	CHAR	1	Não	Não	Não	Valores possíveis: "T": valor de shi_valor é um outlier. "F": valor de shi_valor não é um outlier. NULL: não se sabe se o valor lido é ou não é um outlier.

2.6.1.7.2 Restrições da Tabela T_SERIE_HISTORICA

Nome	Tipo	Campos
PK_T_SERIE_HISTORICA	PK	SHI_OPE_MT_CHAPA, SHI_PCD_ID, SHI_MED_ID, SHI_DATA
FK_SHI_OPE	FK OPERADOR (OPE_MT_CHAPA)	SHI_OPE_CHAPA
FK_SHI_MED	FK MEDIDA (MED_ID)	SHI_MED_ID
FK_SHI_PCD	FK PCD (PCD_ID)	SHI_PCD_ID

2.6.2 Diretório de Dados

O Diretório de Dados descreve processos associados às entidades.

Entidade	Processo
T_ESTATISTICA	A tabela T_ESTATISTICA é preenchida por Stored Procedures que, consultando os dados em T_SERIE_HISTORICA, produzem as estatísticas de acordo com o período e os filtros selecionados pelo usuário.
T_FILTRO	A tabela T_FILTROS é preenchida com os filtros mais comuns utilizados pelo usuário. Para cada filtro, há uma stored procedure que o implementa.
T_MEDIDOR	A tabela T_MEDIDOR é preenchida a partir de formulários coletados em campo os quais descrevem um PCD (Posto de Coleta de Dados) e seus respectivos medidores. Portanto, os dados desta tabela são preenchidos simultaneamente com os dados da T_PCD.
T_OPERADOR	A tabela T_OPERADOR é preenchida a partir de formulários coletados em campo. Logo após serem contratados os operadores ganham um número de Chapa e são cadastrados em T_OPERADOR.
T_PCD	A tabela T_MEDIDOR é preenchida a partir de formulários coletados em campo os quais descrevem um PCD (Posto de Coleta de Dados) e seus respectivos medidores. Portanto, os dados desta tabela são preenchidos simultaneamente com os dados da T_MEDIDOR.
T_PERIODO	Esta é uma tabela auxiliar do sistema. Pode-se considerar que seus dados somente serão modificados em futuras expansões do módulo.
T_SERIE_HISTORICA	T_SERIE_HISTORICA armazena os dados efetivamente lidos em campo. Estes dados podem ser enviados via satélite, via links de radiofrequência ou coletados manualmente (via formulários preenchidos em campo). Antes de serem inseridos no banco, os dados passam por um processo de depuração.

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

2.6.2.1 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.

2.6.2.2 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.

2.6.3 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. 18) 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.

2.6.4 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, a seguir.

O relacionamento entre as entidades está no Anexo I, Seção 5.3, Pág. 19.

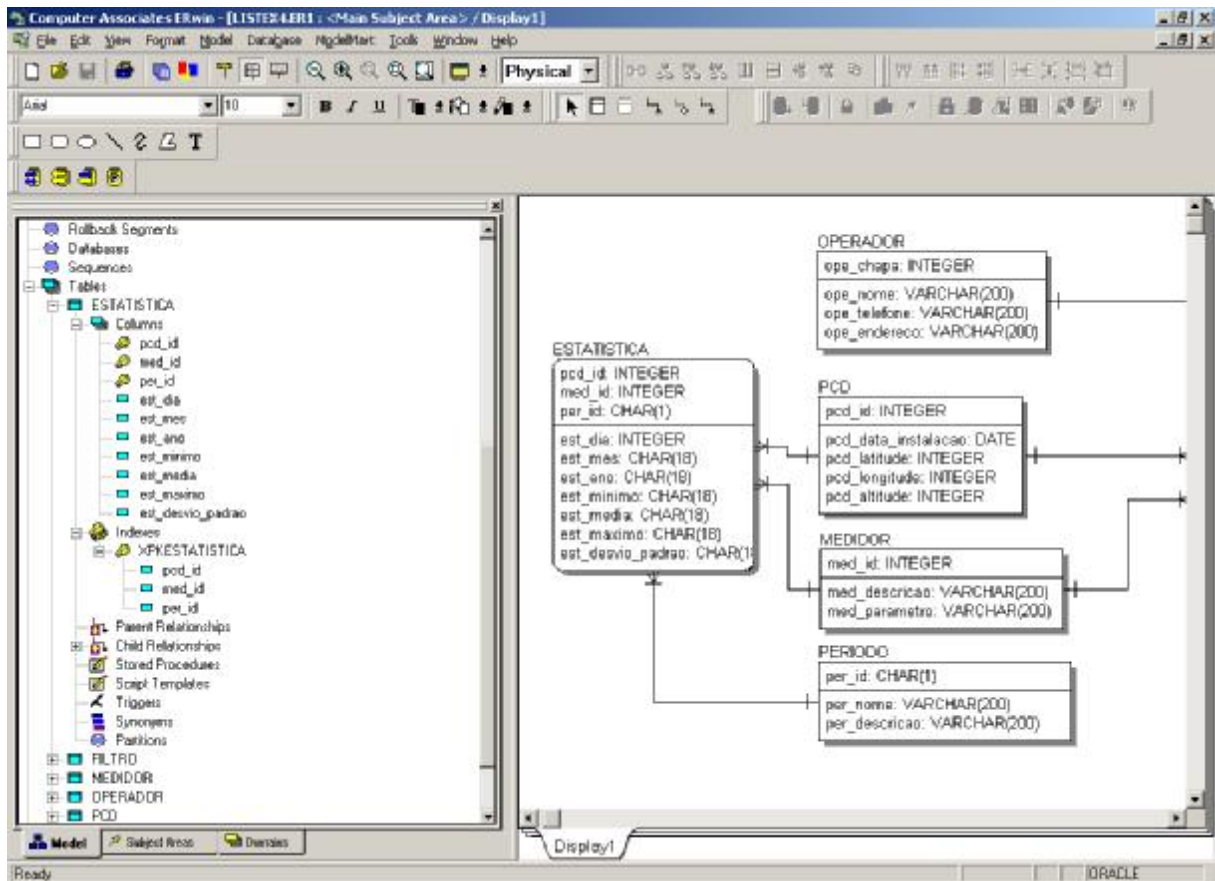


Figure 1: Visão Parcial do Dicionário de Metadados no ERWin 4.0.

2.7 Conversão para o Modelo Hierárquico

No Modelo Hierárquico [6], o acesso a uma determinada entidade tem que ser precedido pelo acesso à uma entidade de hierarquia superior. Logo, temos que organizar hierarquicamente as entidades criando uma árvore na qual a entidade de maior hierarquia está na raiz e as de menor hierarquia se situam nas folhas.

Consideramos três bases hierárquicas:

- **Dados Medidos:** organiza hierarquicamente T_OPERADOR, T_PCD, T_MEDIDOR, T_SERIE_HISTORICA.
- **Estatísticas:** T_PERIODO e T_ESTATISTICA.
- **Filtros:** T_FILTRO.

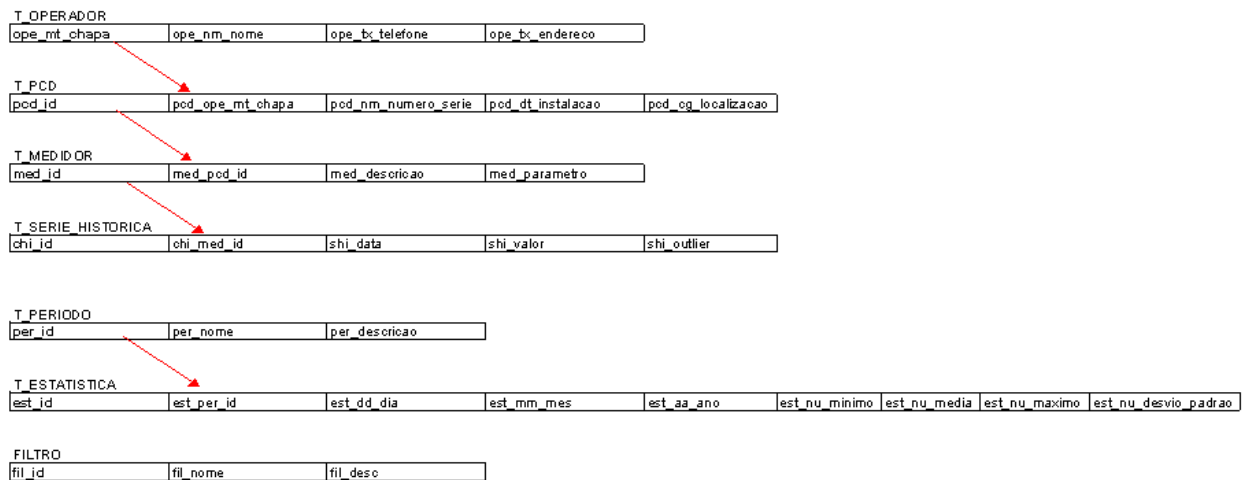


Figura 2: Conversão para o Modelo de Dados Hierárquico.

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.

2.8 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.

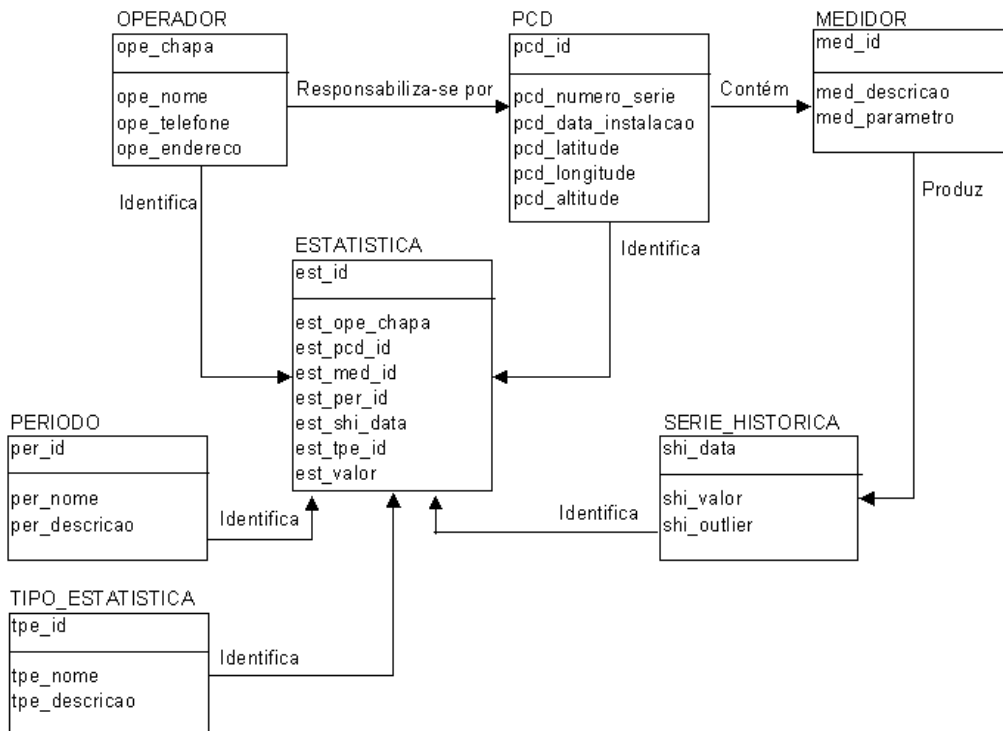


Figura 3: Conversão para o Modelo de Dados em Rede.

2.9 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.

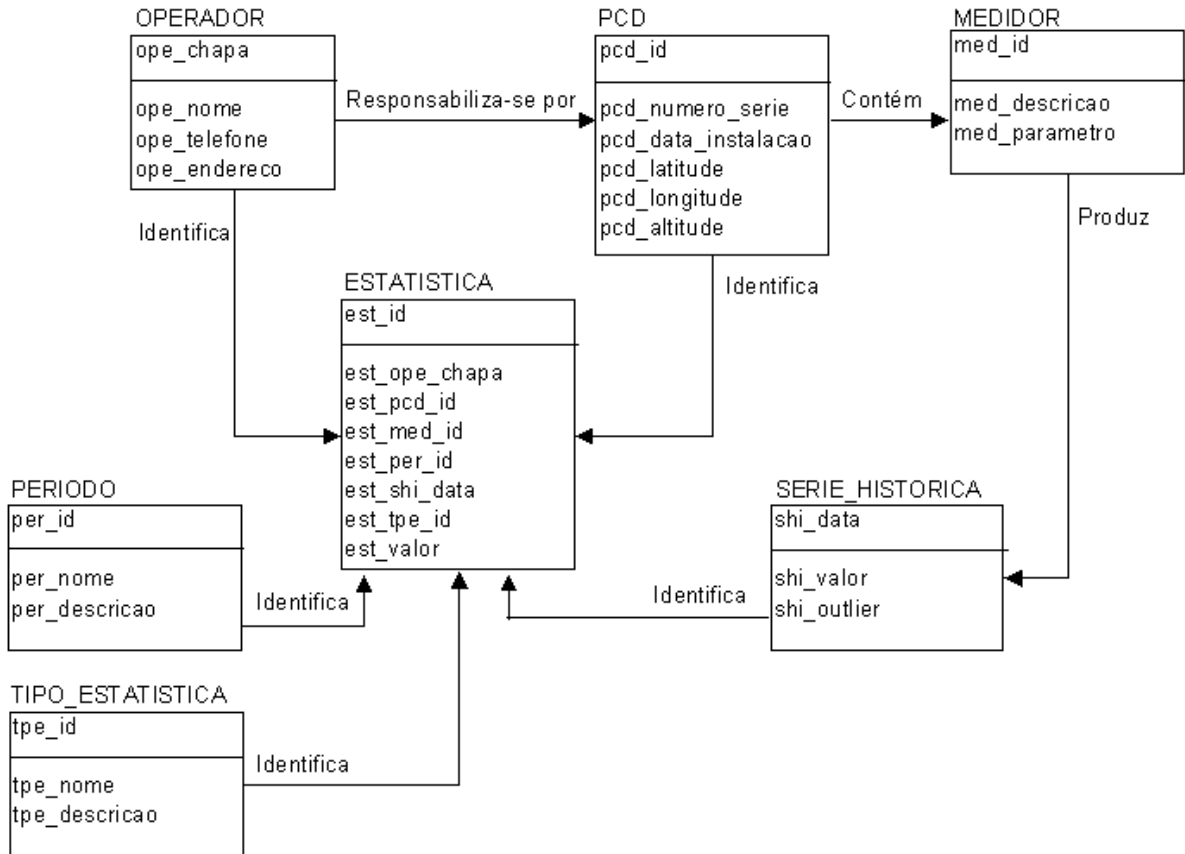


Figura 4: Conversão para o Modelo Orientado a Objetos.

2.10 Definição de Qual Modelo Utilizar

O modelo hierárquico apresenta baixo desempenho pois, para se ler um determinado dado faz-se sempre necessário percorrer a hierarquia de entidades.

O modelo rede é um pouco melhor, mas ainda carece da flexibilidade do modelo relacional.

O modelo de dados relacional, é mais adequado pois pretendemos usar dados georeferenciados em Oracle 11g o qual já possui tipos.

Embora o modelo orientado a objetos seja mais atual, a implementação do modelo relacional será mais rápida, devido a maior experiência dos alunos com este modelo.

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 à nomenclatura das tabelas e atributos.

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_cg_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.

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] CUNHA, ADILSON MARQUES DA. Regras e Padrões para Nomenclatura de Objetos - Versão 9.0. Notas da Aula 03 do curso “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>. Arquivo Aula03.4aCe24009m(Padr|oObjetosDeBD-Prof.CunhaProfa.DanielaGláuciaElton).pdf. Acessado em: 02 de maio de 2009.

[3] 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.

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

[5] 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>

[6] WIKIMEDIA FOUNDATION, INC. Wikipedia (*Hierarchical Model*). Disponível em: http://en.wikipedia.org/wiki/Hierarchical_model. 2009. Visitado em: 23 de maio de 2009.

5 Anexos

5.1 Instalação do Oracle 11g

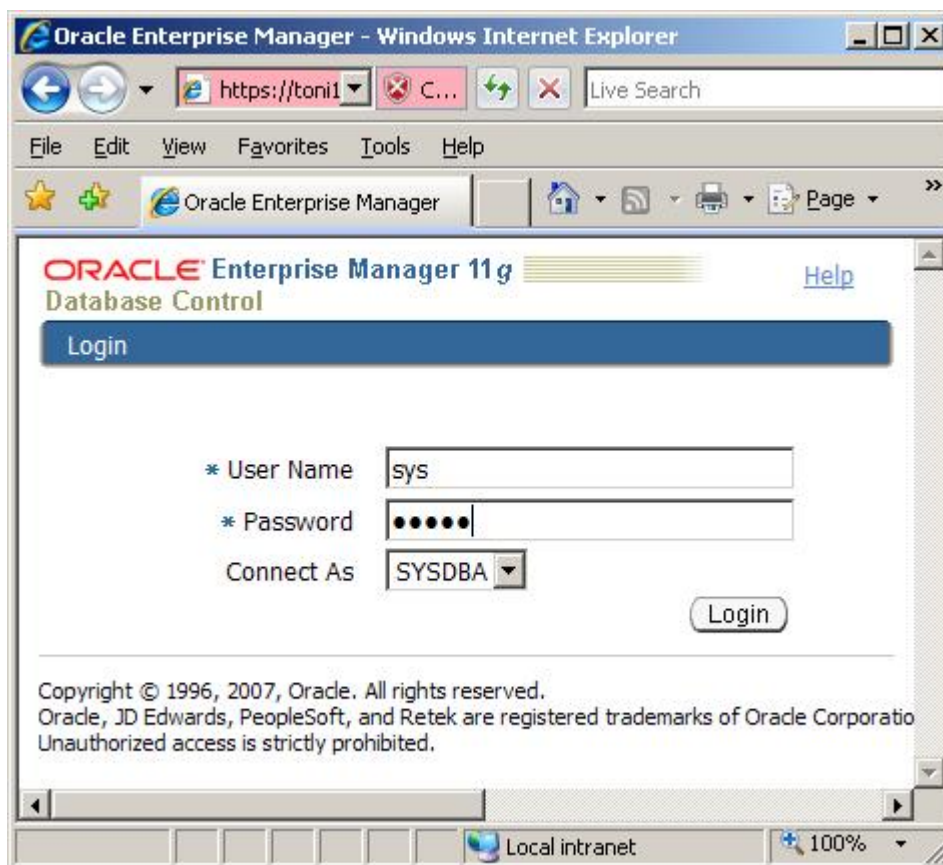


Figura 5: Tela de Login do Oracle Enterprise Manager 11g na máquina TONI10.

5.2 Modelo Físico na 3NF da ListEx 3

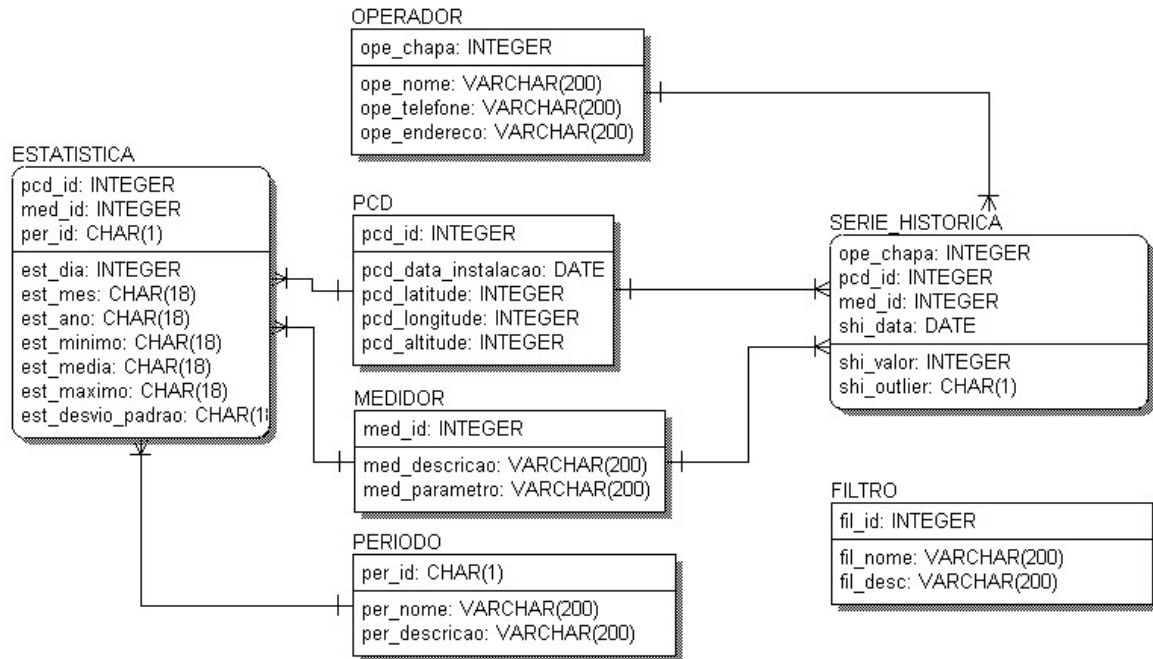


Figura 6: Modelo Físico do Módulo AFE no ERWin 4.0 de acordo com a ListEx 3.

5.3 Modelo Físico na 3NF da ListEx 4

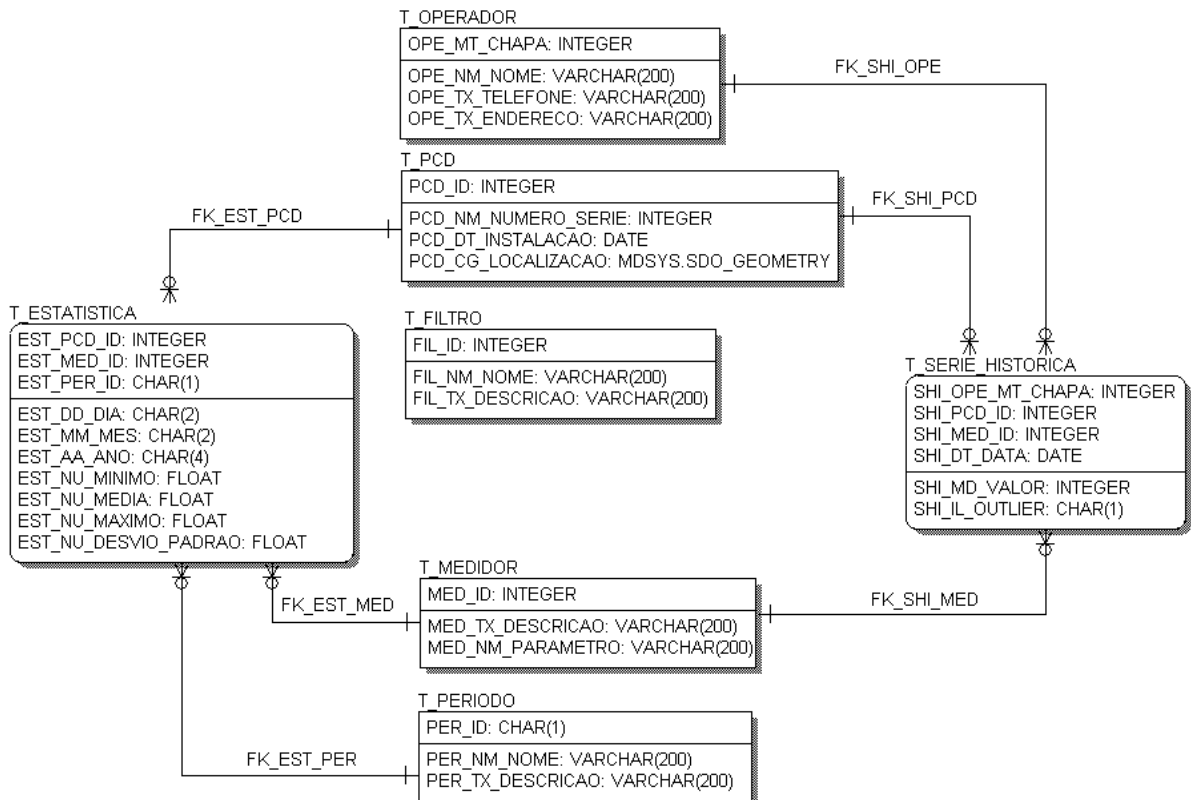


Figura 7: Modelo Físico do Módulo AFE no ERWin 4.0 de acordo com a ListEx 4.

5.4 Comandos para Criação de Tabelas no Oracle 11g

Os comandos a seguir foram obtidos a partir do código gerado pelo ERWin 4.0.

Após ter sido gerado pelo ERWin, o código sofreu algumas alterações:

- Redefinição dos nomes das chaves primárias e chaves estrangeiras visando adequá-las às regras de nomenclatura..
- Inclusão de alguns índices visando garantir a unicidade de dados que não fazem parte da chave primária.
- Acréscimo de comentários visando facilitar a leitura do código.

```

/*****
/* INICIO DO SCRIPT LISTEX4.SQL
/*****

/*****
/* TABELAS
/*****

/*****
CREATE TABLE T_ESTADISTICA (
    EST_PCD_ID          INTEGER NOT NULL,
    EST_MED_ID          INTEGER NOT NULL,
    EST_PER_ID          CHAR(1) NOT NULL,
    EST_DD_DIA          CHAR(2) NULL,
    EST_MM_MES          CHAR(2) NULL,
    EST_AA_ANO          CHAR(4) NOT NULL,
    EST_NU_MINIMO       FLOAT NULL,
    EST_NU_MEDIA        FLOAT NULL,
    EST_NU_MAXIMO       FLOAT NULL,
    EST_NU_DESVIO_PADRAO FLOAT NULL
);

ALTER TABLE T_ESTADISTICA
    ADD CONSTRAINT PK_T_ESTADISTICA
    PRIMARY KEY (EST_PCD_ID, EST_MED_ID, EST_PER_ID);

/*****
CREATE TABLE T_FILTRO (
    FIL_ID              INTEGER NOT NULL,
    FIL_NM_NOME         VARCHAR(200) NOT NULL,
    FIL_TX_DESCRICAÇÃO  VARCHAR(200) NOT NULL
);

```

```
ALTER TABLE T_FILTRO
ADD CONSTRAINT PK_T_FILTRO
PRIMARY KEY (FIL_ID);
```

```
CREATE UNIQUE INDEX IUQ_T_FILTRO on T_FILTRO(FIL_NM_NOME);
```

```
/*
CREATE TABLE T_MEDIDOR (
MED_ID                INTEGER NOT NULL,
MED_TX_DESCRICAO     VARCHAR(200) NOT NULL,
MED_NM_PARAMETRO     VARCHAR(200) NOT NULL
);
```

```
ALTER TABLE T_MEDIDOR
ADD CONSTRAINT PK_T_MEDIDOR
PRIMARY KEY (MED_ID);
```

```
CREATE          UNIQUE          INDEX          IUQ_T_MEDIDOR          on
T_MEDIDOR(MED_TX_DESCRICAO,MED_NM_PARAMETRO);
```

```
/*
CREATE TABLE T_OPERADOR (
OPE_MT_CHAPA        INTEGER NOT NULL,
OPE_NM_NOME         VARCHAR(200) NOT NULL,
OPE_TX_TELEFONE     VARCHAR(200) NULL,
OPE_TX_ENDERECO     VARCHAR(200) NULL
);
```

```
ALTER TABLE T_OPERADOR
ADD CONSTRAINT PK_T_OPERADOR
PRIMARY KEY (OPE_MT_CHAPA);
```

```
/*
CREATE TABLE T_PCD (
PCD_ID              INTEGER NOT NULL,
PCD_NM_NUMERO_SERIE INTEGER NOT NULL,
PCD_DT_INSTALACAO  DATE NOT NULL,
PCD_CG_LOCALIZACAO MDSYS.SDO_GEOMETRY NOT NULL
);
```



```

ALTER TABLE T_PCD
  ADD CONSTRAINT PK_T_PCD
  PRIMARY KEY (PCD_ID);

CREATE          UNIQUE          INDEX          IUQ_T_PCD          on
T_PCD(PCD_NM_NUMERO_SERIE,PCD_DT_INSTALACAO);

/*****/
CREATE TABLE T_PERIODO (
  PER_ID          CHAR(1) NOT NULL,
  PER_NM_NOME    VARCHAR(200) NOT NULL,
  PER_TX_DESCRICA0  VARCHAR(200) NOT NULL
);

ALTER TABLE T_PERIODO
  ADD CONSTRAINT PK_T_PERIODO
  PRIMARY KEY (PER_ID);

CREATE UNIQUE INDEX IUQ_T_PERIODO on T_PERIODO(PER_NM_NOME);

/*****/
CREATE TABLE T_SERIE_HISTORICA (
  SHI_OPE_MT_CHAPA    INTEGER NOT NULL,
  SHI_PCD_ID          INTEGER NOT NULL,
  SHI_MED_ID          INTEGER NOT NULL,
  SHI_DATA             DATE NOT NULL,
  SHI_VALOR            INTEGER NULL,
  SHI_OUTLIER          CHAR(1) NOT NULL
);

ALTER TABLE T_SERIE_HISTORICA
  ADD CONSTRAINT PK_T_SERIE_HISTORICA
  PRIMARY KEY (SHI_OPE_MT_CHAPA, SHI_PCD_ID, SHI_MED_ID,SHI_DATA);

/*****/
/* FOREIGN KEYS */
/*****/
ALTER TABLE T_ESTADISTICA
  ADD CONSTRAINT FK_EST_PER
  FOREIGN KEY (EST_PER_ID)
  REFERENCES T_PERIODO (PER_ID);

```

```
ALTER TABLE T_ESTADISTICA
ADD CONSTRAINT FK_EST_MED
FOREIGN KEY (EST_MED_ID)
REFERENCES T_MEDIDOR (MED_ID);
```

```
ALTER TABLE T_ESTADISTICA
ADD CONSTRAINT FK_EST_PCD
FOREIGN KEY (EST_PCD_ID)
REFERENCES T_PCD (PCD_ID);
```

```
ALTER TABLE T_SERIE_HISTORICA
ADD CONSTRAINT FK_SHI_MED
FOREIGN KEY (SHI_MED_ID)
REFERENCES T_MEDIDOR (MED_ID) ;
```

```
ALTER TABLE T_SERIE_HISTORICA
ADD CONSTRAINT FK_SHI_PCD
FOREIGN KEY (SHI_PCD_ID)
REFERENCES T_PCD (PCD_ID);
```

```
ALTER TABLE T_SERIE_HISTORICA
ADD CONSTRAINT FK_SHI_OPE
FOREIGN KEY (SHI_OPE_MT_CHAPA)
REFERENCES T_OPERADOR (OPE_MT_CHAPA);
```

```
/*
/* TRIGGERS
/*
```

```
create trigger tI_T_ESTADISTICA after INSERT on T_ESTADISTICA for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- INSERT trigger on T_ESTADISTICA
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PERIODO R/6 T_ESTADISTICA ON CHILD INSERT RESTRICT */
  select count(*) into numrows
  from T_PERIODO
  where
    /* :new.EST_PER_ID = T_PERIODO.PER_ID */
    :new.EST_PER_ID = T_PERIODO.PER_ID;
  if (
    /* */

    numrows = 0
```

```

)
then
  raise_application_error(
    -20002,
    'Cannot INSERT T_ESTADISTICA because T_PERIODO does not exist.'
  );
end if;

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_MEDIDOR R/5 T_ESTADISTICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
  from T_MEDIDOR
  where
    /* :new.EST_MED_ID = T_MEDIDOR.MED_ID */
    :new.EST_MED_ID = T_MEDIDOR.MED_ID;
if (
  /* */

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

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_PCD R/4 T_ESTADISTICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
  from T_PCD
  where
    /* :new.EST_PCD_ID = T_PCD.PCD_ID */
    :new.EST_PCD_ID = T_PCD.PCD_ID;
if (
  /* */

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

```

```

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tU_T_ESTADISTICA after UPDATE on T_ESTADISTICA for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_ESTADISTICA
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PERIODO R/6 T_ESTADISTICA ON CHILD UPDATE RESTRICT */
  select count(*) into numrows
    from T_PERIODO
   where
     /* :new.EST_PER_ID = T_PERIODO.PER_ID */
     :new.EST_PER_ID = T_PERIODO.PER_ID;
  if (
    /* */

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

  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_MEDIDOR R/5 T_ESTADISTICA ON CHILD UPDATE RESTRICT */
  select count(*) into numrows
    from T_MEDIDOR
   where
     /* :new.EST_MED_ID = T_MEDIDOR.MED_ID */
     :new.EST_MED_ID = T_MEDIDOR.MED_ID;
  if (
    /* */

    numrows = 0
  )
  then
    raise_application_error(

```

```

        -20007,
        'Cannot UPDATE T_ESTADISTICA because T_MEDIDOR does not exist.'
    );
end if;

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_PCD R/4 T_ESTADISTICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
  from T_PCD
  where
    /* :new.EST_PCD_ID = T_PCD.PCD_ID */
    :new.EST_PCD_ID = T_PCD.PCD_ID;
if (
  /* */

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

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger td_T_MEDIDOR after DELETE on T_MEDIDOR for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- DELETE trigger on T_MEDIDOR
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_MEDIDOR R/5 T_ESTADISTICA ON PARENT DELETE RESTRICT */
  select count(*) into numrows
    from T_ESTADISTICA
    where
      /* T_ESTADISTICA.EST_MED_ID = :old.MED_ID */
      T_ESTADISTICA.EST_MED_ID = :old.MED_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20001,

```

```

        'Cannot DELETE T_MEDIDOR because T_ESTADISTICA exists.'
    );
end if;

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_MEDIDOR R/3 T_SERIE_HISTORICA ON PARENT DELETE RESTRICT */
select count(*) into numrows
  from T_SERIE_HISTORICA
  where
    /* T_SERIE_HISTORICA.SHI_MED_ID = :old.MED_ID */
    T_SERIE_HISTORICA.SHI_MED_ID = :old.MED_ID;
if (numrows > 0)
then
  raise_application_error(
    -20001,
    'Cannot DELETE T_MEDIDOR because T_SERIE_HISTORICA exists.'
  );
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tU_T_MEDIDOR after UPDATE on T_MEDIDOR for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_MEDIDOR
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_MEDIDOR R/5 T_ESTADISTICA ON PARENT UPDATE RESTRICT */
  if
    /* :old.MED_ID <> :new.MED_ID */
    :old.MED_ID <> :new.MED_ID
  then
    select count(*) into numrows
      from T_ESTADISTICA
      where
        /* T_ESTADISTICA.EST_MED_ID = :old.MED_ID */
        T_ESTADISTICA.EST_MED_ID = :old.MED_ID;
    if (numrows > 0)
    then
      raise_application_error(
        -20005,

```

```

        'Cannot UPDATE T_MEDIDOR because T_ESTADISTICA exists.'
    );
end if;
end if;

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_MEDIDOR R/3 T_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 T_SERIE_HISTORICA
    where
        /* T_SERIE_HISTORICA.SHI_MED_ID = :old.MED_ID */
        T_SERIE_HISTORICA.SHI_MED_ID = :old.MED_ID;
    if (numrows > 0)
    then
        raise_application_error(
            -20005,
            'Cannot UPDATE T_MEDIDOR because T_SERIE_HISTORICA exists.'
        );
    end if;
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tD_T_OPERADOR after DELETE on T_OPERADOR for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- DELETE trigger on T_OPERADOR
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 24 11:23:20 2009 */
    /* T_OPERADOR R/1 T_SERIE_HISTORICA ON PARENT DELETE RESTRICT */
    select count(*) into numrows
    from T_SERIE_HISTORICA
    where
        /* T_SERIE_HISTORICA.SHI_OPE_MT_CHAPA = :old.OPE_MT_CHAPA */
        T_SERIE_HISTORICA.SHI_OPE_MT_CHAPA = :old.OPE_MT_CHAPA;
    if (numrows > 0)
    then

```

```

        raise_application_error(
            -20001,
            'Cannot DELETE T_OPERADOR because T_SERIE_HISTORICA exists.'
        );
    end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tU_T_OPERADOR after UPDATE on T_OPERADOR for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_OPERADOR
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 24 11:23:20 2009 */
    /* T_OPERADOR R/1 T_SERIE_HISTORICA ON PARENT UPDATE RESTRICT */
    if
        /* :old.OPE_MT_CHAPA <> :new.OPE_MT_CHAPA */
        :old.OPE_MT_CHAPA <> :new.OPE_MT_CHAPA
    then
        select count(*) into numrows
        from T_SERIE_HISTORICA
        where
            /* T_SERIE_HISTORICA.SHI_OPE_MT_CHAPA = :old.OPE_MT_CHAPA */
            T_SERIE_HISTORICA.SHI_OPE_MT_CHAPA = :old.OPE_MT_CHAPA;
        if (numrows > 0)
        then
            raise_application_error(
                -20005,
                'Cannot UPDATE T_OPERADOR because T_SERIE_HISTORICA exists.'
            );
        end if;
    end if;
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tD_T_PCD after DELETE on T_PCD for each row
-- ERwin Builtin Sun May 24 11:23:20 2009

```



```

-- DELETE trigger on T_PCD
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PCD R/4 T_ESTADISTICA ON PARENT DELETE RESTRICT */
  select count(*) into numrows
    from T_ESTADISTICA
    where
      /* T_ESTADISTICA.EST_PCD_ID = :old.PCD_ID */
      T_ESTADISTICA.EST_PCD_ID = :old.PCD_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20001,
      'Cannot DELETE T_PCD because T_ESTADISTICA exists.'
    );
  end if;

  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PCD R/2 T_SERIE_HISTORICA ON PARENT DELETE RESTRICT */
  select count(*) into numrows
    from T_SERIE_HISTORICA
    where
      /* T_SERIE_HISTORICA.SHI_PCD_ID = :old.PCD_ID */
      T_SERIE_HISTORICA.SHI_PCD_ID = :old.PCD_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20001,
      'Cannot DELETE T_PCD because T_SERIE_HISTORICA exists.'
    );
  end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tU_T_PCD after UPDATE on T_PCD for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_PCD
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */

```

```

/* T_PCD R/4 T_ESTADISTICA ON PARENT UPDATE RESTRICT */
if
  /* :old.PCD_ID <> :new.PCD_ID */
  :old.PCD_ID <> :new.PCD_ID
then
  select count(*) into numrows
  from T_ESTADISTICA
  where
    /* T_ESTADISTICA.EST_PCD_ID = :old.PCD_ID */
    T_ESTADISTICA.EST_PCD_ID = :old.PCD_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20005,
      'Cannot UPDATE T_PCD because T_ESTADISTICA exists.'
    );
  end if;
end if;

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_PCD R/2 T_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 T_SERIE_HISTORICA
  where
    /* T_SERIE_HISTORICA.SHI_PCD_ID = :old.PCD_ID */
    T_SERIE_HISTORICA.SHI_PCD_ID = :old.PCD_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20005,
      'Cannot UPDATE T_PCD because T_SERIE_HISTORICA exists.'
    );
  end if;
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/

```

```

create trigger tD_T_PERIODO after DELETE on T_PERIODO for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- DELETE trigger on T_PERIODO
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PERIODO R/6 T_ESTADISTICA ON PARENT DELETE RESTRICT */
  select count(*) into numrows
    from T_ESTADISTICA
   where
     /* T_ESTADISTICA.EST_PER_ID = :old.PER_ID */
     T_ESTADISTICA.EST_PER_ID = :old.PER_ID;
  if (numrows > 0)
  then
    raise_application_error(
      -20001,
      'Cannot DELETE T_PERIODO because T_ESTADISTICA exists.'
    );
  end if;

```

```

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

```

```

/*****/
create trigger tU_T_PERIODO after UPDATE on T_PERIODO for each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_PERIODO
declare numrows INTEGER;
begin
  /* ERwin Builtin Sun May 24 11:23:20 2009 */
  /* T_PERIODO R/6 T_ESTADISTICA ON PARENT UPDATE RESTRICT */
  if
    /* :old.PER_ID <> :new.PER_ID */
    :old.PER_ID <> :new.PER_ID
  then
    select count(*) into numrows
      from T_ESTADISTICA
     where
       /* T_ESTADISTICA.EST_PER_ID = :old.PER_ID */
       T_ESTADISTICA.EST_PER_ID = :old.PER_ID;
    if (numrows > 0)
    then
      raise_application_error(

```

```

        -20005,
        'Cannot UPDATE T_PERIODO because T_ESTADISTICA exists.'
    );
end if;
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tI_T_SERIE_HISTORICA after INSERT on T_SERIE_HISTORICA for
each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- INSERT trigger on T_SERIE_HISTORICA
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 24 11:23:20 2009 */
    /* T_MEDIDOR R/3 T_SERIE_HISTORICA ON CHILD INSERT RESTRICT */
    select count(*) into numrows
    from T_MEDIDOR
    where
        /* :new.SHI_MED_ID = T_MEDIDOR.MED_ID */
        :new.SHI_MED_ID = T_MEDIDOR.MED_ID;
    if (
        /* */

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

    /* ERwin Builtin Sun May 24 11:23:20 2009 */
    /* T_PCD R/2 T_SERIE_HISTORICA ON CHILD INSERT RESTRICT */
    select count(*) into numrows
    from T_PCD
    where
        /* :new.SHI_PCD_ID = T_PCD.PCD_ID */
        :new.SHI_PCD_ID = T_PCD.PCD_ID;
    if (

```

```

        /* */

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

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_OPERADOR R/1 T_SERIE_HISTORICA ON CHILD INSERT RESTRICT */
select count(*) into numrows
from T_OPERADOR
where
    /* :new.SHI_OPE_MT_CHAPA = T_OPERADOR.OPE_MT_CHAPA */
    :new.SHI_OPE_MT_CHAPA = T_OPERADOR.OPE_MT_CHAPA;
if (
    /* */

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

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

/*****/
create trigger tU_T_SERIE_HISTORICA after UPDATE on T_SERIE_HISTORICA for
each row
-- ERwin Builtin Sun May 24 11:23:20 2009
-- UPDATE trigger on T_SERIE_HISTORICA
declare numrows INTEGER;
begin
    /* ERwin Builtin Sun May 24 11:23:20 2009 */
    /* T_MEDIDOR R/3 T_SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
    select count(*) into numrows

```

```

from T_MEDIDOR
where
  /* :new.SHI_MED_ID = T_MEDIDOR.MED_ID */
  :new.SHI_MED_ID = T_MEDIDOR.MED_ID;
if (
  /* */

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

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_PCD R/2 T_SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
  from T_PCD
  where
    /* :new.SHI_PCD_ID = T_PCD.PCD_ID */
    :new.SHI_PCD_ID = T_PCD.PCD_ID;
if (
  /* */

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

/* ERwin Builtin Sun May 24 11:23:20 2009 */
/* T_OPERADOR R/1 T_SERIE_HISTORICA ON CHILD UPDATE RESTRICT */
select count(*) into numrows
  from T_OPERADOR
  where
    /* :new.SHI_OPE_MT_CHAPA = T_OPERADOR.OPE_MT_CHAPA */
    :new.SHI_OPE_MT_CHAPA = T_OPERADOR.OPE_MT_CHAPA;
if (
  /* */

```

```

        numRows = 0
    )
then
    raise_application_error(
        -20007,
        'Cannot UPDATE T_SERIE_HISTORICA because T_OPERADOR does not exist.'
    );
end if;

-- ERwin Builtin Sun May 24 11:23:20 2009
end;
/

```

```

/*****
/* FIM DO SCRIPT LISTEX4.SQL
/*****

```

5.5 Georeferenciamento da Tabela T_PCD

Foi necessário executar alguns comandos [4] de configuração do Oracle visando executar queries georeferenciadas na tabela T_PCD.

```

/*****
/* INICIO DO SCRIPT LISTEX4_PCD.SQL
/*****

```

```

delete user_sdo_geom_metadata
where table_name = 'T_PCD'
and column_name = 'PCD_CG_LOCALIZACAO';

insert into user_sdo_geom_metadata (table_name,column_name,diminfo,srid)
values
('T_PCD','PCD_CG_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_T_PCD_SPATIAL on T_PCD(PCD_CG_LOCALIZACAO)
indextype is mdsys.spatial_index
parameters ('sdo_indx_dims=2 layer_gtype=point');

```

```

/*****

```

```

/* FIM DO SCRIPT LISTEX4_PCD.SQL */
/*****

```

5.6 Exemplos de Dados na ListEx 3

As tabelas a seguir, transcritas da ListEx 3, 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_parametro	med_descricao
1	ALTPLU	Altura pluviométrica (mm)
2	TEMPCEL	Temperatura (Celsius)
3	LAMDAG	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_desvio_padrao
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.

5.7 Comandos SQL para Inserir Exemplos de Dados

A partir dos exemplos de dados da ListEx 3 foram criados os comandos para inserção de dados no Modelo Físico definido na ListEx 4. Os comandos foram gravados no arquivo `listex4_dados.sql` e reproduzidos abaixo:

```

/*****/
/* INICIO DO LISTEX4_DADOS.SQL */
/*****/

/*****/
/* T_OPERADOR */
/*****/

insert                               into                               T_OPERADOR
(OPE_MT_CHAPA,OPE_NM_NOME,OPE_TX_TELEFONE,OPE_TX_ENDERECO) values
(1,'José Silva','55 92 1234 5678','Igarapé, 5 - Lagoa - Manaus, AM');

insert                               into                               T_OPERADOR
(OPE_MT_CHAPA,OPE_NM_NOME,OPE_TX_TELEFONE,OPE_TX_ENDERECO) values
(2,'Maria Silva','55 68 4321 8765','Rua Riozinho, 123 - Bairro das
Enchentes - Bujari, AC');

insert                               into                               T_OPERADOR
(OPE_MT_CHAPA,OPE_NM_NOME,OPE_TX_TELEFONE,OPE_TX_ENDERECO) values

```

```

(3,'Cacique Juruna',NULL,'Tribo dos Piraraquaras, Alto Xingu, PA');
commit;

/*****
/* T_PCD
*****/

insert into T_PCD(PCD_ID, PCD_NM_NUMERO_SERIE, PCD_DT_INSTALACAO,
PCD_CG_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 T_PCD(PCD_ID, PCD_NM_NUMERO_SERIE, PCD_DT_INSTALACAO,
PCD_CG_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 T_PCD(PCD_ID, PCD_NM_NUMERO_SERIE, PCD_DT_INSTALACAO,
PCD_CG_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;

/*****
/* T_MEDIDOR
*****/

insert into T_MEDIDOR(MED_ID,MED_TX_DESCRICAO,MED_NM_PARAMETRO) values
(1,'Altura Pluviométrica (mm)','ALTPLU');

```

```

insert into T_MEDIDOR(MED_ID,MED_TX_DESCRICAO,MED_NM_PARAMETRO) values
(2,'Temperatura (Celsius)', 'TEMPCEL');
insert into T_MEDIDOR(MED_ID,MED_TX_DESCRICAO,MED_NM_PARAMETRO) values
(3,'Lâmina D' 'Água (m)', 'LAMDAG');
commit;

```

```

/*****/
/* T_PERIODO          */
/*****/

```

```

insert into T_PERIODO(PER_ID,PER_NM_NOME,PER_TX_DESCRICAO) values
('D','Dia','Cálculos estatísticos considerando períodos de 1 dia. ');
insert into T_PERIODO(PER_ID,PER_NM_NOME,PER_TX_DESCRICAO) values
('M','Mês','Cálculos estatísticos considerando períodos de 1 mês. ');
insert into T_PERIODO(PER_ID,PER_NM_NOME,PER_TX_DESCRICAO) values
('A','Ano','Cálculos estatísticos considerando períodos de 1 ano. ');
commit;

```

```

/*****/
/* T_FILTRO          */
/*****/

```

```

insert into T_FILTRO(FIL_ID,FIL_NM_NOME,FIL_TX_DESCRICAO) values
(1,'Média Móvel','Retorna o valor filtrado como sendo a média aritmética
dos N valores anteriormente lidos numa série histórica. ');
insert into T_FILTRO(FIL_ID,FIL_NM_NOME,FIL_TX_DESCRICAO) values
(2,'Outlier','Preenche o campo outlier para uma determinada série
histórica. ');
insert into T_FILTRO(FIL_ID,FIL_NM_NOME,FIL_TX_DESCRICAO) 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 T_FILTRO(FIL_ID,FIL_NM_NOME,FIL_TX_DESCRICAO) 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;

```

```

/*****/
/* T_SERIE_HISTORICA  */
/*****/

```

```

insert
into
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD
_VALOR,SHI_IL_OUTLIER) values

```

```
(1,1,1,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),34,'F');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(1,1,1,to_date('01/01/2000 10:32:35','DD/MM/YYYY HH24:MI:SS'),35,'F');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(1,1,1,to_date('01/01/2000 10:34:40','DD/MM/YYYY HH24:MI:SS'),42,'T');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(2,2,1,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),34,'F');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(2,2,2,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),50,'F');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(2,2,3,to_date('01/01/2000 10:30:35','DD/MM/YYYY HH24:MI:SS'),1.25,'F');
```

```
insert                                                    into  
T_SERIE_HISTORICA(SHI_OPE_MT_CHAPA,SHI_PCD_ID,SHI_MED_ID,SHI_DT_DATA,SHI_MD  
_VALOR,SHI_IL_OUTLIER) values
```

```
(2,2,3,to_date('01/01/2000 12:30:35','DD/MM/YYYY HH24:MI:SS'),1.35,'F');
```

```
commit;
```

```
/*  
*****  
/* T_ESTADISTICA          */  
*****  
*/
```

```
insert                                                    into  
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA  
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
```

```
(1,1,'D',01,01,2000,34.0,34.5,35.0,1.0);
```

```
insert                                                    into  
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA  
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
```

```
(1,1,'M',NULL,01,2000,34.0,34.5,35.0,1.0);
```

```
insert                                                    into
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
(1,1,'A',NULL,NULL,2000,34.0,34.5,35.0,1.0);
```

```
insert                                                    into
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
(2,2,'D',01,01,2000,34.0,34.0,34.0,0);
```

```
insert                                                    into
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
(2,2,'M',NULL,01,2000,34.0,34.0,34.0,0);
```

```
insert                                                    into
T_ESTADISTICA(EST_PCD_ID,EST_MED_ID,EST_PER_ID,EST_DD_DIA,EST_MM_MES,EST_AA
_ANO,EST_NU_MINIMO,EST_NU_MEDIA,EST_NU_MAXIMO,EST_NU_DESVIO_PADRAO) values
(2,2,'A',NULL,NULL,2000,34.0,34.0,34.0,0);
```

```
commit;
```

```
/*
*****
/* FIM DO LISTEX4_DADOS.SQL */
*****
```