

Bancos de Dados Geográficos

Projeto Físico de BDG

Clodoveu Davis

Mapeamento lógico-físico

- Apesar da padronização da codificação da geometria dos objetos, os desenvolvedores de SIG implementam modelos físicos diferentes e inflexíveis, o que dificulta a interoperabilidade
- Existe uma tendência para o aperfeiçoamento desse ponto em futuras versões

Mapeamento lógico-físico

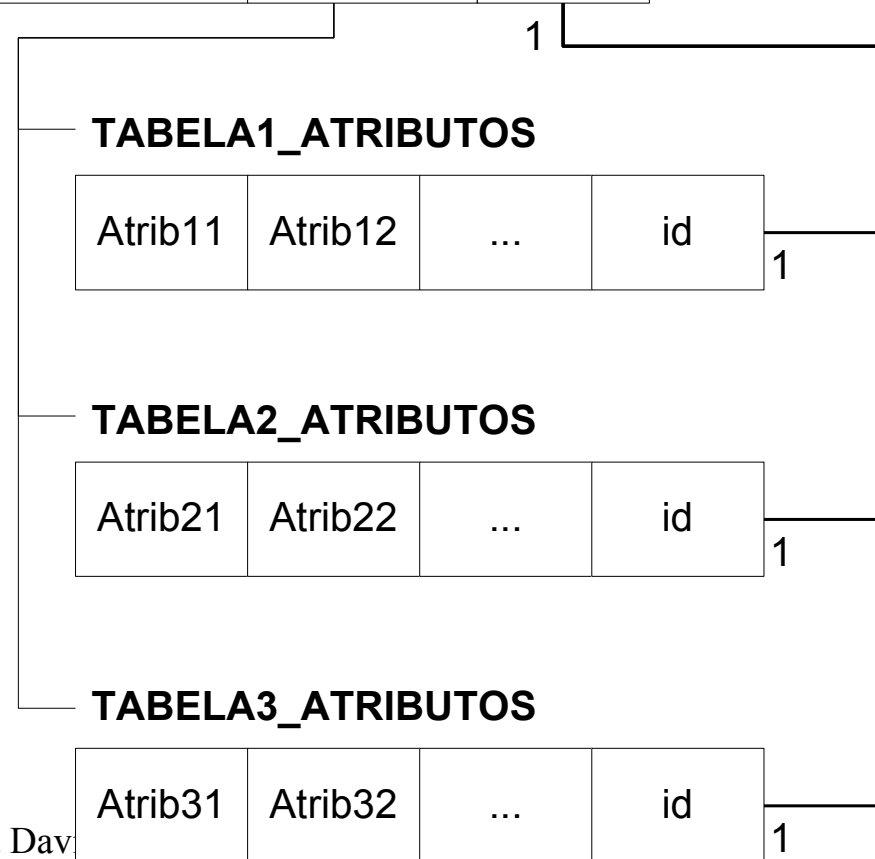
- As estruturas lógicas produzidas a partir do mapeamento dos esquemas OMT-G podem ser implementadas de maneiras diversas, dependendo da arquitetura computacional usada
- Se a aplicação for de fato baseada em um BD objeto-relacional dentro do padrão OGC, o mapeamento lógico-físico depende principalmente de decidir como tratar os dados geográficos

Mapeamento lógico-físico

- Alternativa 1: geometrias concentradas em uma única tabela

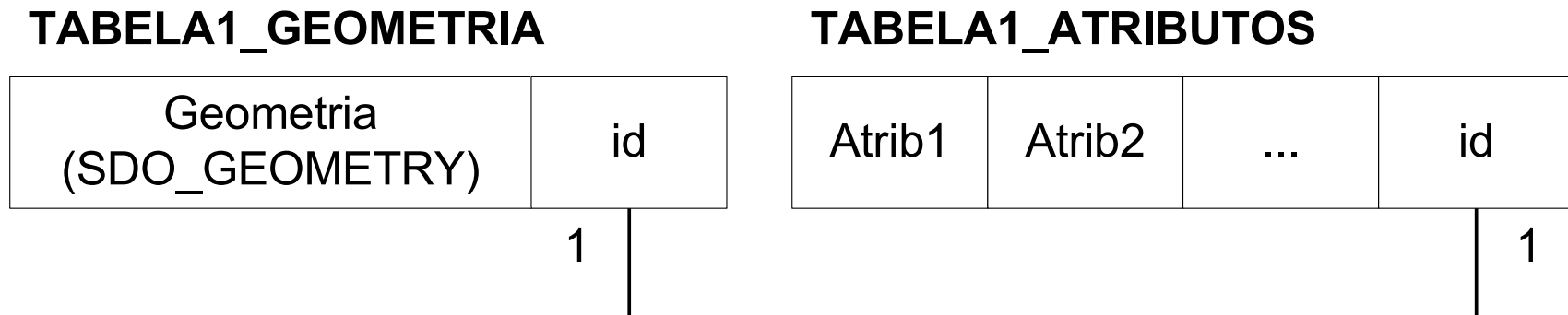
GEOMETRIA_TOTAL

Geometria (SDO_GEOMETRY)	Nome da Tabela	id
-----------------------------	-------------------	----



Mapeamento lógico-físico

- Alternativa 2: um par de tabelas relacionadas para cada classe georreferenciada



Mapeamento lógico-físico

- Alternativa 3: geometria e atributos na mesma tabela

TABELA1_GORREF

Geometria (SDO_GEOMETRY)	Atrib1	Atrib2	...	id
-----------------------------	--------	--------	-----	----

Mapeamento lógico-físico

- Comparação entre alternativas
 - Alternativa 1:
 - produz um desequilíbrio no SGBD, concentrando acessos a uma única tabela, que se torna crítica
 - indexação espacial e operações topológicas entre classes de objetos são teoricamente beneficiadas
 - alivia as tabelas alfanuméricas

Mapeamento lógico-físico

– Alternativa 2:

- desvantagem: exige *join* entre tabelas para se recuperar a estrutura completa do objeto
- vantagem: flexibilidade na possível associação entre vários conjuntos de dados alfanuméricos e uma única representação espacial, ou entre várias representações e um mesmo conjunto de dados alfanuméricos
- exige a garantia de integridade referencial entre a tabela que contém a parte gráfica e a tabela alfanumérica, algo trivial em SGBD convencionais mas que nem sempre é implementado nos bancos gerados por SIG

Mapeamento lógico-físico

– Alternativa 3

- mais semelhante ao modelo conceitual de objetos, como o definido pelo OMT-G
 - provoca um aumento no volume de dados em uma tabela, forçando aplicações convencionais a acessar mais dados do que requerem para operar
 - integridade referencial intrinsecamente garantida
 - relação 1:1 entre parte gráfica e parte alfanumérica
- É possível implementar um mapeamento entre cada par de alternativas

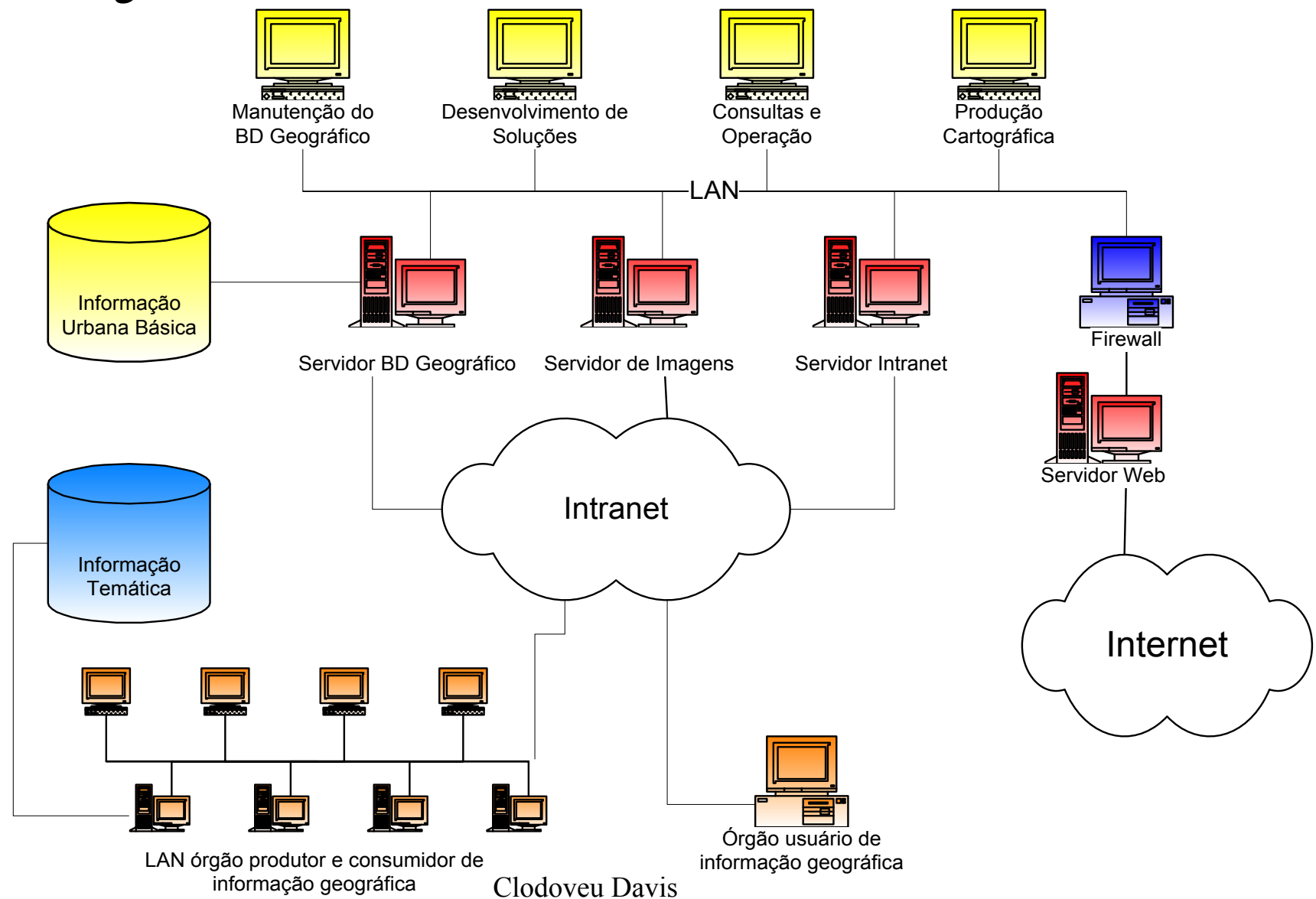
Mapeamento lógico-físico

- Caso o mapeamento seja direcionado a um produto SIG cujo banco de dados esteja embutido no próprio produto, o mapeamento deve ser planejado pensando nas eventuais limitações do BD
 - Promover a equivalência entre tabelas lógicas e *features* (*layers*, *coverages*, etc.), com seus respectivos atributos
 - Verificar a possibilidade de implementação das restrições de integridade
 - Definir as transformações modeladas para execução manual ou por meio de desenvolvimento de software específico
 - Implementar os esquemas de apresentação, usando os recursos do produto

OpenGIS: interoperabilidade em SIG

- Arquitetura de software para acesso distribuído a dados geo-espaciais e recursos de geoprocessamento em geral
 - Modelo de dados (*Open Geodata Model*, OGM)
 - Modelo de serviços de acesso (*OGIS Services Model*)
 - Comunidades de informação (*Information Communities Model*)
- O primeiro resultado expressivo do OpenGIS é a *Simple Features Specification*, que viabilizou o Oracle Spatial

SIG objeto-relacional



SIG objeto-relacional interoperável

- Liberdade de escolha de software para cada função (manutenção da base, consulta, navegação, produção cartográfica, etc.), e conseqüente possibilidade de adequação de custos
- Maiores possibilidades de integração aos sistemas corporativos convencionais
- Maior capilaridade
- Possibilidade de implementação de aplicativos geográficos sem a necessidade de aquisição de software SIG específico

Objetos Geográficos em BD objeto-relacional

- O uso de SGBD relacionais em associação ao SIG vem sendo feito há muito tempo
 - Objetivo: possibilitar algum nível de integração entre o SIG e os sistemas de informação convencionais (legados)
- Em geral, os dados geométricos são armazenados em estruturas externas ao SGBD
 - Objetivos: melhorar o desempenho e ocultar as estruturas de dados usadas da concorrência
 - Conseqüência: ambientes com dados em formato proprietário

Objetos Geográficos em BD objeto-relacional

- Bancos de dados relacionais são baseados em *tabelas*
- Em geral, as tabelas são formadas por colunas (atributos) contendo dados de apenas um tipo (numérico, *string*, data, lógico, etc.) e com uma largura definida
- Tentativas de codificar os dados geométricos em tabelas convencionais foram feitas
 - Problemas: tamanho do banco e desempenho
 - Vantagem: formato aberto e livremente legível

Objetos Geográficos em BD objeto-relacional

- Avanços recentes na tecnologia de bancos de dados permitiram a inclusão de campos de tamanho variável, denominados BLObs (*Binary Large Objects*)
- Os BLObs não são a melhor solução para o armazenamento de objetos geográficos, mas como suportam uma quantidade variável de dados, foram usados em algumas implementações
 - Problema: formato proprietário dos dados (binário) impossibilita o compartilhamento

Objetos Geográficos em BD objeto-relacional

- Mais recentemente (1999), os principais gerenciadores de bancos de dados passaram a trabalhar com *objetos* armazenados em colunas especiais, convivendo com colunas convencionais
- A estrutura interna de cada objeto é previamente definida pelo usuário na modelagem física dos dados
- A Oracle usou esse recurso para criar um modelo de objeto geográfico de acordo com os padrões estipulados pelo Consórcio OpenGIS em suas *Simple Feature Specifications*

Objetos Geográficos em BD objeto-relacional

- O formato do objeto geográfico do Oracle Spatial é aberto e documentado
 - Problemas (iniciais):
 - ausência de recursos intrínsecos para codificação de redes (incorporado na versão 10g)
 - ausência de suporte para geo-campos, inclusive imagens (incorporado na versão 10g, para imagens)
 - Vantagens:
 - interoperabilidade: suporte por vários desenvolvedores de SIG cliente
 - possibilidade de acesso por parte de aplicações desenvolvidas pelo próprio usuário, ou por módulos integrados aos sistemas convencionais “expandidos”

Oracle Spatial

Clodoveu Davis

Objetos Geográficos em BD objeto-relacional

- Estrutura dos objetos do tipo SDO_GEOMETRY :

```
CREATE TYPE SDO_GEOMETRY AS OBJECT {  
  SDO_GTYPE          NUMBER,  
  SDO_SRID           NUMBER,  
  SDO_POINT          SDO_POINT_TYPE,  
  SDO_ELEM_INFO     SDO_ELEM_INFO_ARRAY,  
  SDO_ORDINATES     SDO_ORDINATE_ARRAY };
```

Objetos Geográficos em BD objeto-relacional

- **SDO_GTYPE**: tipo de geometria do objeto
 - Previsão para 3D, mas falta um tipo para superfícies

Valor de <i>SDO_GTYPE</i>	Tipo de geometria
d000	Desconhecido
d001	Ponto
d002	Poligonal
d003	Polígono (são permitidos buracos)
d004	Coleção heterogênea de elementos
d005	Múltiplos pontos
d006	Múltiplas poligonais
d007	Múltiplos polígonos (ilhas)

Objetos Geográficos em BD objeto-relacional

- SDO_SRID: identificação do sistema de coordenadas usado *na codificação do objeto*
 - Podem existir na mesma tabela objetos cujas coordenadas estão expressos em sistemas diferentes
 - É tarefa do SIG (interface) interpretar e projetar essas coordenadas sobre a tela no momento da visualização
 - O valor do campo é numérico, e serve de chave para outra tabela em que os parâmetros das diversas projeções cartográficas estão codificados
 - Ex: SAD-69, hemisfério Sul, fuso 23 (45° WGr): código 83701

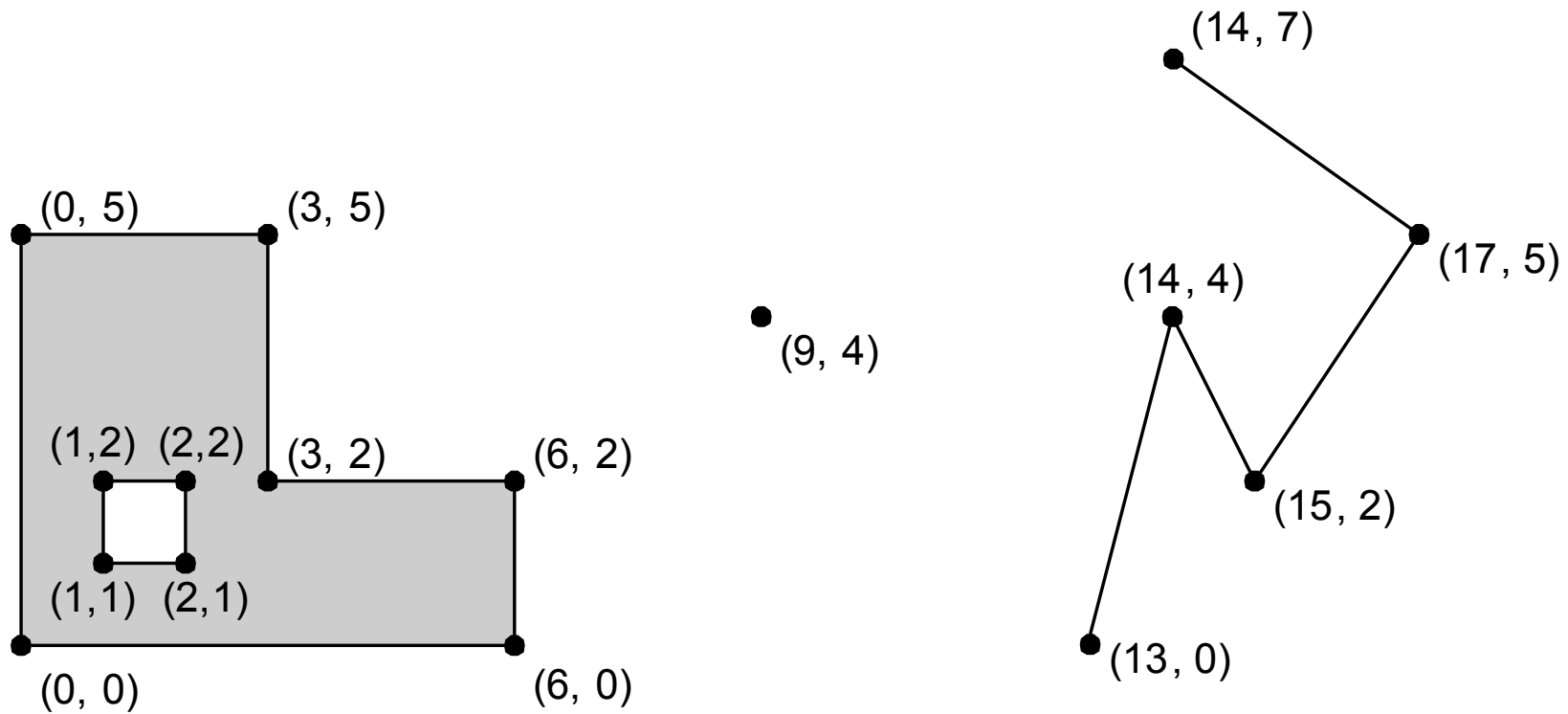
Objetos Geográficos em BD objeto-relacional

- SDO_POINT: coordenadas de um vértice tridimensional (variável do tipo SDO_POINT_TYPE)
 - Se o objeto for do tipo ponto, toda a geometria está definida
 - Se o objeto for de outro tipo, este campo é ignorado
 - Recurso válido para acelerar a seleção e apresentação de objetos do tipo ponto, muito comuns nas aplicações

Objetos Geográficos em BD objeto-relacional

- `SDO_ELEM_INFO` e `SDO_ORDINATES`: dois vetores coordenados que codificam a geometria de objetos lineares e poligonais
 - `SDO_ORDINATES`: uma seqüência de conjuntos de coordenadas 3D
 - `SDO_ELEM_INFO`: separação das coordenadas em grupos, dando a cada grupo uma definição de comportamento

Objetos Geográficos em BD objeto-relacional



Objeto	Campo	Valor	Comentário
1	SDO_GTYPE	2003	Polígono simples, 2D
1	SDO_SRID	83201	UTM, Fuso 23, hemisfério sul, SAD-69
1	SDO_POINT	<nulo>	Desnecessário
1	SDO_ELEM_INFO	(1, 1003, 1, 8, 2003, 1)	Começando do par de coordenadas número 1, trata-se de um polígono externo (1003), com arestas retas (1); do par 8 em diante, é um polígono interno ou buraco (2003), com arestas retas (1).
1	SDO_ORDINATES	(0, 0, 6, 0, 6, 2, 3, 2, 3, 5, 0, 5, 0, 0, 1, 1, 1, 2, 2, 2, 2, 1, 1, 1)	Seqüência x, y em sentido anti-horário no polígono externo e em sentido horário para o polígono interno (indica buraco); o último ponto repete o primeiro nos dois casos.
2	SDO_GTYPE	2001	Ponto 2D
2	SDO_SRID	82301	UTM, Fuso 23, hemisfério sul, SAD-69
2	SDO_POINT	(9, 4, 0)	Coordenadas do ponto
2	SDO_ELEM_INFO	<nulo>	Desnecessário
2	SDO_ORDINATES	<nulo>	Desnecessário
3	SDO_GTYPE	2002	Poligonal 2D
3	SDO_SRID	82301	UTM, Fuso 23, hemisfério sul, SAD-69
3	SDO_POINT	<nulo>	Desnecessário
3	SDO_ELEM_INFO	(1, 2, 1)	Começando do par de coordenadas número 1, trata-se de uma poligonal aberta (2), com arestas retas (1).
3	SDO_ORDINATES	(13, 0, 14, 4, 15, 2, 17, 5, 14, 7)	Seqüência x, y.

Objetos Geográficos em BD objeto-relacional

- Não existe na estrutura de armazenamento do Oracle nenhuma informação sobre apresentação
 - Isso é deixado a cargo do SIG (cliente)
 - Benefício: isolamento completo entre a representação e a apresentação de cada objeto
- Não existe nenhum tipo de codificação explícita da topologia
 - Relacionamentos topológicos são definidos dinamicamente, no momento do processamento de uma consulta ou da realização de uma operação

Metadados no Oracle

- Acessados como visões do dicionário principal
- USER_SDO_GEOM_METADATA
 - Orienta o Oracle nas funções de validação, criação de índices e consulta

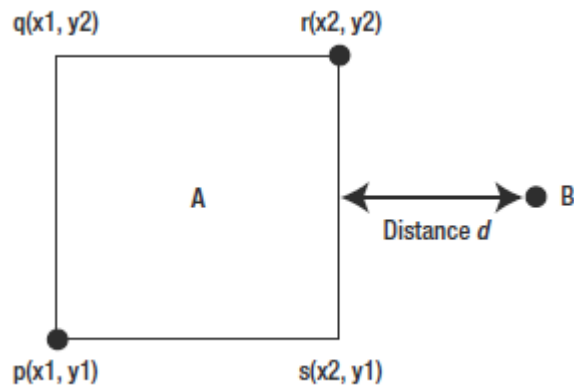
USER_SDO_GEOM_METADATA

- Atributos:
 - TABLE_NAME varchar2(32)
 - COLUMN_NAME varchar2(1024)
 - DIMINFO MDSYS.SDO_DIM_ARRAY
 - SRID number

USER_SDO_GEOM_METADATA

- Atributos de DIMINFO (MDSYS.SDO_DIM_ELEMENT):
 - SDO_DIMNAME varchar2(64)
 - SDO_LB number
 - SDO_UB number
 - SDO_TOLERANCE number
 - Grau de precisão dos dados espaciais: menor distância de separação entre dois objetos para que possam ser considerados disjuntos

Tolerância



- Se $d < \text{tol}$ então B está na fronteira de A
- Se $d \geq \text{tol}$, A e B são considerados disjuntos
- Se $(x_2 - x_1) < \text{tol}$, então os vértices p e s (e q e r) são considerados duplicados, e a geometria é considerada inválida

Tolerância

Table 3-1. *Suggested Values for Tolerance Based on SRID for Applications*

Coordinate System	SRID Values	Tolerance	Units
Geodetic coordinate system (such as 8265, 8307)	Select SRID from MDSYS.CS_SRS, where WKTEXT is like 'GEOGCS%'.	0.5 (should not be less than 0.05)	Meters for tolerance; degrees for longitude, latitude dimensions.
Projected coordinate system (such as 32774)	Select SRID from MDSYS.CS_SRS, where WKTEXT is like 'PROJCS%'.	Half of the smallest difference between any two values in a dimension	Units for tolerance are the same as the units for the ordinates in the dimensions.
Local coordinate system	Select SRID from MDSYS.CS_SRS, where WKTEXT is like 'LOCAL_CS%'.	Half of the smallest difference between any two values in a dimension	Units for tolerance are the same as the units for the ordinates in the dimensions.
No specific coordinate syst	NULL.	Half of the smallest difference between any two values in a dimension	Units for tolerance are the same as the units for the ordinates in the dimensions.

Mais Metadados

- `USER_SDO_INDEX_METADATA_VIEW`
 - Parâmetros dos índices, usado em tuning
- `USER_SDO_STYLES`, `USER_SDO_THEMES`,
`USER_SDO_MAPS`
 - Usados pelo MapViewer
- `USER_SDO_NETWORK_METADATA`
 - Usado pelo módulo de rede

Validação de geometrias

```
SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT  
(geom SDO_GEOMETRY,  
tolerance NUMBER) return varchar2
```

Ou

```
SDO_GEOM.VALIDATE_GEOMETRY_WITH_CONTEXT  
(geom SDO_GEOMETRY,  
SDO_DIM_ARRAY DIMINFO) return varchar2
```

Validação de geometrias

- ```
SELECT
SDO_GEOM.VALIDATE_GEOMETRY_
WITH_CONTEXT(a.geom, 0.5)
FROM tabela a
WHERE a.id = 10000
```

# Validação de layers

```
SDO_GEOM.VALIDATE_LAYER_WITH_CONTEXT
(varchar2 TABLE_NAME,
varchar2 COLUMN_NAME,
varchar2 RESULT_TABLE,
[number COMMIT_INTERVAL])
```

- Result\_table: uma coluna para o SDO\_ROWID, e outra para o status (varchar2(2000))

# Validação e correção

- Funções SDO\_UTIL
  - REMOVE\_DUPLICATE\_VERTICES
  - EXTRACT
  - APPEND
  - SDO\_CONCAT\_LINES
  - SDO\_REVERSE\_LINESTRING
  - SDO\_POLYGONTOLINE

# Objetos Geográficos em BD objeto-relacional

- Além da codificação dos objetos geográficos, o SGBD objeto-relacional implementa uma série de recursos geográficos:
  - Indexação espacial por quad-tree e R-tree
  - Funções de relacionamento espacial que podem compor predicados para pesquisa no banco usando a linguagem SQL

# Objetos Geográficos em BD objeto-relacional

- Ainda falta muito para que o SGBD O-R espacial se torne um recurso genérico para SIG
  - Aumento do suporte e adoção pelos desenvolvedores de SIG, com integral implementação do padrão
  - Suporte integral a bancos de dados distribuídos
  - Implementação de restrições de integridade espaciais no próprio banco, como critérios de garantia da integridade espacial
  - Implementação de controle de versões para objetos geográficos
  - Implementação de recursos para a representação de variáveis temporais

# Esquema de BD

The screenshot displays the Oracle DBA Studio interface. On the left, a tree view shows the database structure under 'MUB.PBH - SYSTEM'. The 'MUB' schema is expanded, showing various object types like Clusters, Packages, and Tables. The 'Tabelas' (Tables) folder is highlighted. On the right, a table list window shows the following data:

| Tabela                 | Tablespace | Particionada | Linhas |
|------------------------|------------|--------------|--------|
| ARTICUL1               | DADOS      | No           | 1360   |
| ARTICUL1_SX_FL6\$      | DADOS      | No           | 1370   |
| ARTICUL5               | DADOS      | No           | 68     |
| ARTICUL5_SX_FL6\$      | DADOS      | No           | 70     |
| BAIRROP                | DADOS      | No           |        |
| BAIRROPSI1_FL1\$       | DADOS      | No           |        |
| CEMITER                | DADOS      | No           |        |
| CEMITERIO              | DADOS      | No           |        |
| CEMITERIO_QTREE_FL28\$ | DADOS      | No           |        |
| CEMITER1               | DADOS      | No           |        |
| CEMITER1SI1_FL1\$      | DADOS      | No           |        |
| ColumnMetadata         | DADOS      | No           |        |
| CURVAMES               | DADOS      | No           | 7335   |
| CURVAMES_SX_FL8\$      | DADOS      | No           | 7352   |
| ENDERECO               | DADOS      | No           |        |
| ENDERECO_SX_FL4\$      | DADOS      | No           | 0      |
| FERROVIA               | DADOS      | No           |        |
| FERROVIASI1_FL1\$      | DADOS      | No           |        |
| GAliasTable            | DADOS      | No           |        |
| LIMITEBH               | DADOS      | No           |        |
| LIMITEBHSI1_FL1\$      | DADOS      | No           |        |
| OEM\$SDO\$SAMPLE21     | DADOS      | No           |        |
| REGIONAL               | DADOS      | No           |        |
| REGIONALS11_FL1\$      | DADOS      | No           |        |
| TESTE                  | DADOS      | No           |        |
| TESTE1                 | DADOS      | No           |        |
| TESTE1_RTREE_RT\$      | DADOS      | No           |        |

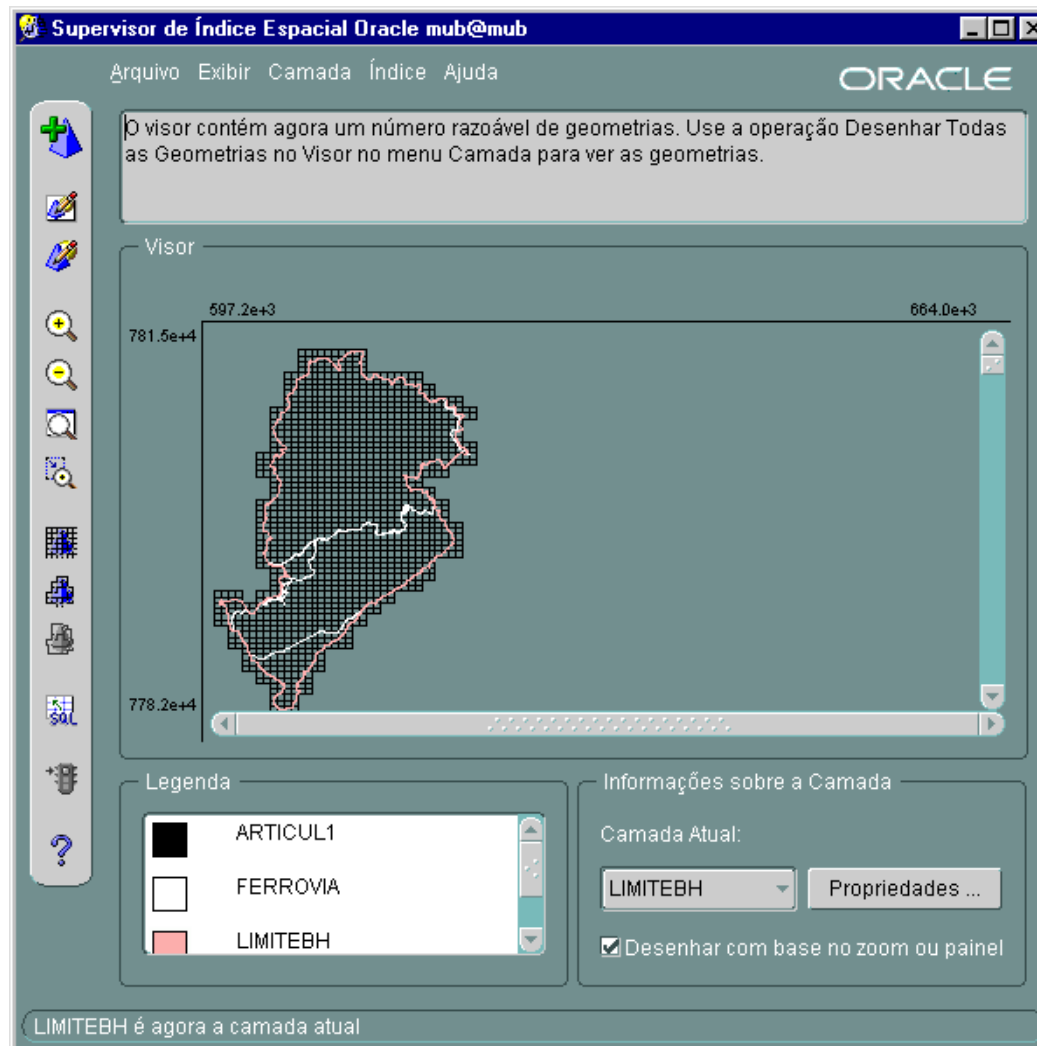


# Esquema de BD

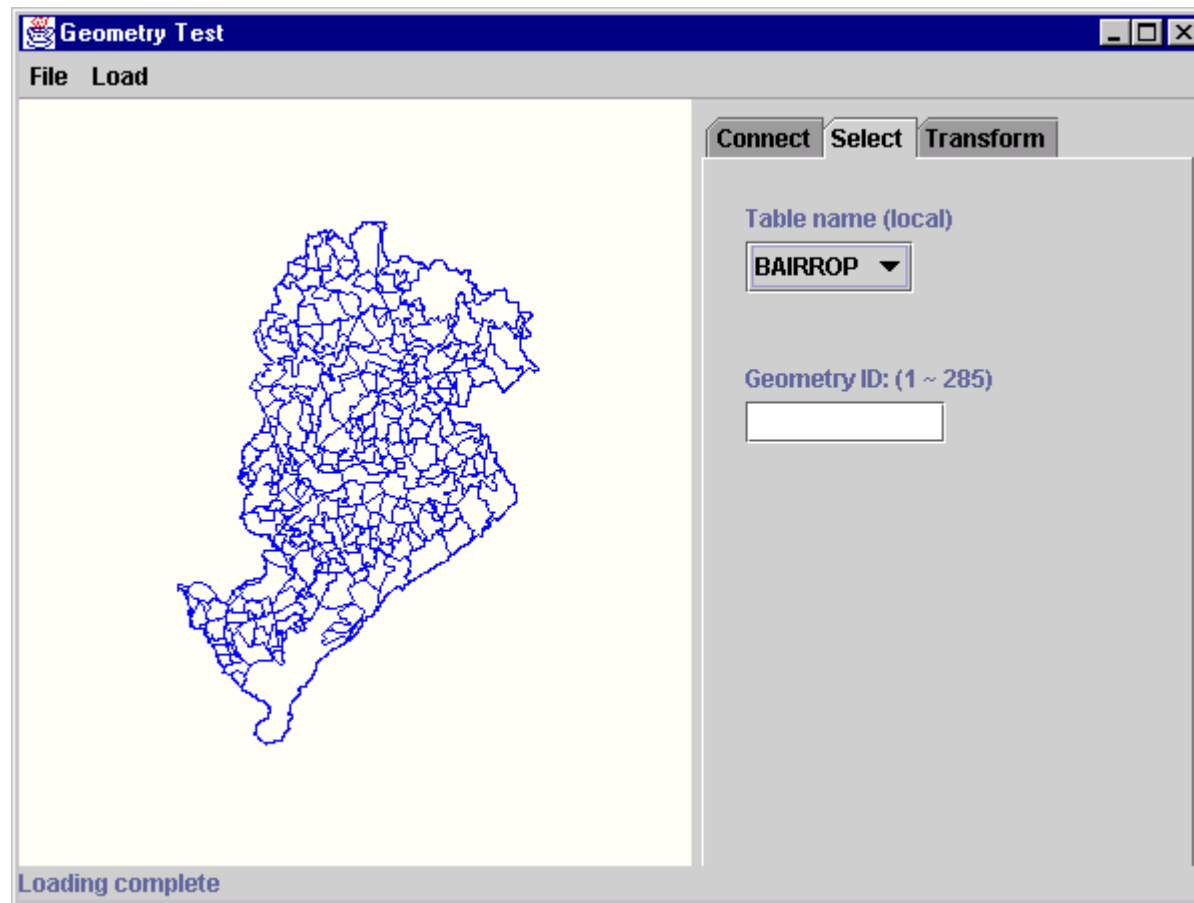
The screenshot displays the Oracle DBA Studio interface. On the left, a tree view shows the database structure under the 'MUB' schema, with 'Tabelas' (Tables) expanded to show 'BAIRROP' selected. The main pane on the right shows the 'Restrições' (Constraints) tab for the 'BAIRROP' table. The table is located in the 'MUB' schema and 'DADOS' tablespace. The table is organized by index (IOT). The columns are listed in the following table:

| Nome          | Esquema  | Tipo de Dados | Tama... | Escala | F |
|---------------|----------|---------------|---------|--------|---|
| NUBAIPOP      | <Nenhum> | VARCHAR2      | 4       |        |   |
| NOBAIPOP      | <Nenhum> | VARCHAR2      | 30      |        |   |
| TPBAIPOP      | <Nenhum> | VARCHAR2      | 25      |        |   |
| MI_PRINX      | <Nenhum> | FLOAT         | 126     |        |   |
| GMIPRIMARYKEY | <Nenhum> | NUMBER        | 0       | 0      |   |
| GEOMETRY      | MDSYS    | SDO_GEOMETRY  |         |        |   |

# Visualização Oracle



# Visualização Java



Clodoveu Davis

# Consulta ao BD via SQL

```
Oracle SQL*Plus
Arquivo Editar Procurar Opções Ajuda
SQL> desc regional;
Nome Nulo? Tipo

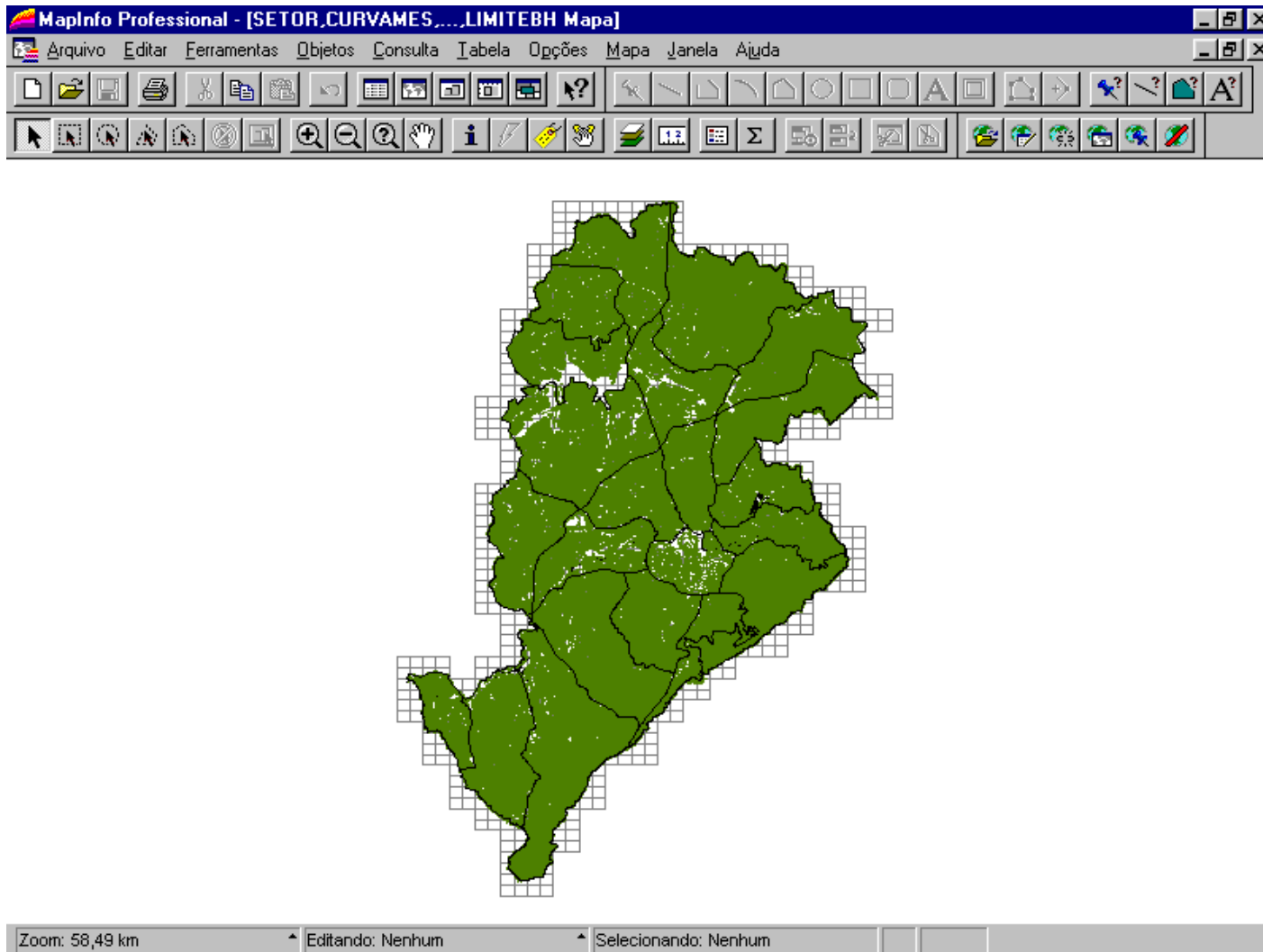
NUREG UARCHAR2(2)
NOREG UARCHAR2(10)
QTMAS FLOAT(126)
QTFEM FLOAT(126)
QTRESOCP FLOAT(126)
QTRESOCA FLOAT(126)
QTRESCOL FLOAT(126)
QTRESFCH FLOAT(126)
QTRESVAG FLOAT(126)
QTNREOCP FLOAT(126)
QTNREVAG FLOAT(126)
QTATUECO FLOAT(126)
QTATUAGR FLOAT(126)
MI_PRINX FLOAT(126)
GMIPRIMARYKEY NOT NULL NUMBER(38)
GEOMETRY MDSYS.SDO_GEOMETRY

SQL> select geometry from regional where nureg=18;

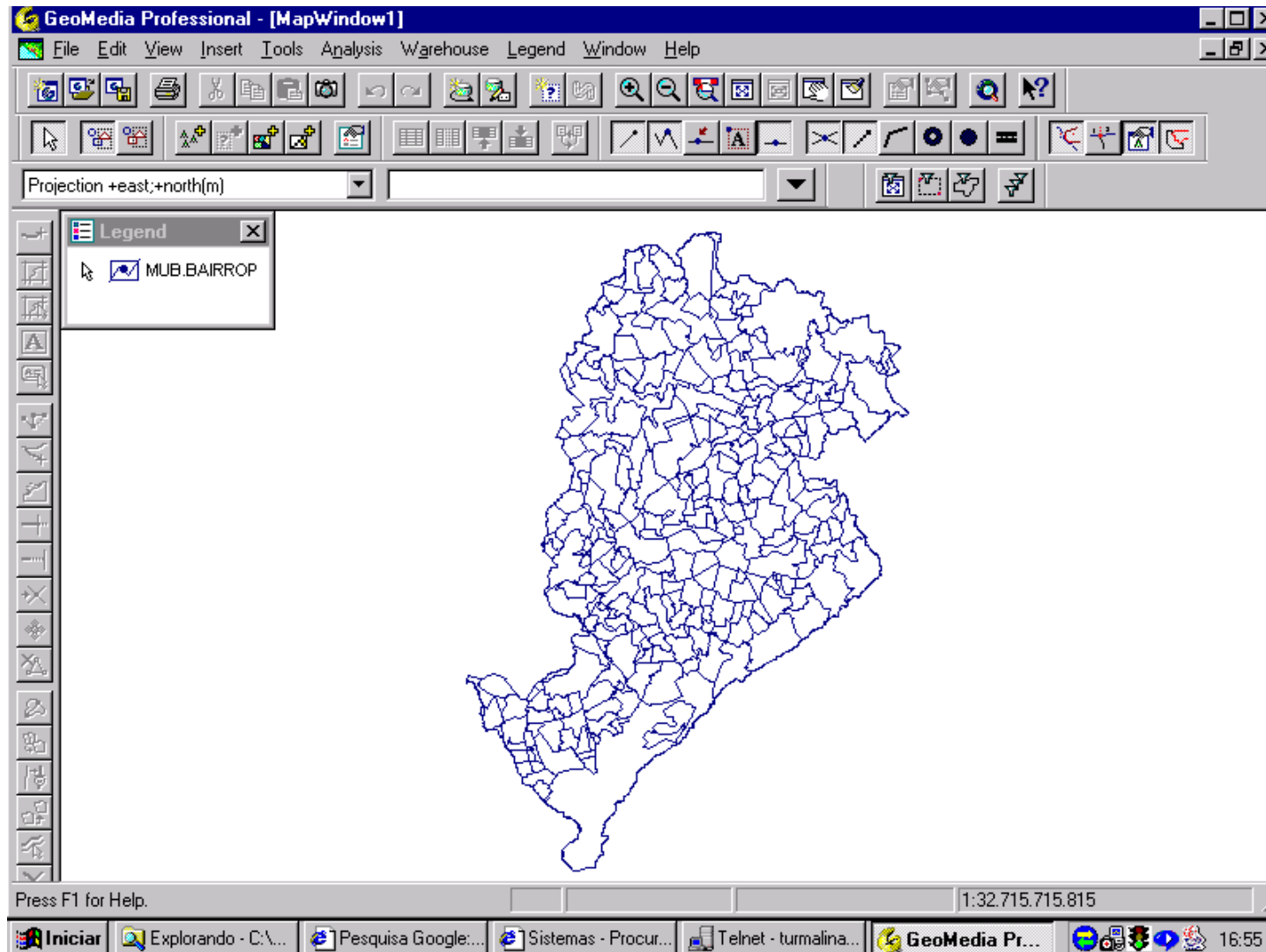
GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT(X, Y, Z), SDO_ELEM_INFO, SDO_ORDINATES)

SDO_GEOMETRY(3003, NULL, NULL, SDO_ELEM_INFO_ARRAY(1, 1003, 1), SDO_ORDINATE_ARR
AY(605225,587, 7784438,7, 0, 605318,831, 7784524,81, 0, 605338,052, 7784639,17,
0, 605396,218, 7784738,24, 0, 605400,597, 7784816,67, 0, 605388,624, 7784851,54,
0, 605342,109, 7784901,4, 0, 605327,299, 7784934,98, 0, 605355,253, 7784965,29,
0, 605357,521, 7784989,17, 0, 605381,12, 7785014,98, 0, 605403,046, 7785019,97,
0, 605447,483, 7785047,1, 0, 605500,619, 7785034,11, 0, 605604,699, 7785024,82,
0, 605695,197, 7785072,58, 0, 605723,901, 7785136,56, 0, 605781,837, 7785155,83
, 0, 605791,847, 7785201,44, 0, 605788,038, 7785242,24, 0, 605796,399, 7785301,0
5, 0, 605862,366, 7785361,92, 0, 605965,159, 7785413,54, 0, 605991,15, 7785447,0
9, 0, 606066,971, 7785590,88, 0, 606079,678, 7785711,97, 0, 606145,472, 7785779,
28, 0, 606187,253, 7785805,13, 0, 606204,256, 7785826,27, 0, 606218,612, 7785864
GEOMETRY(SDO_GTYPE, SDO_SRID, SDO_POINT(X, Y, Z), SDO_ELEM_INFO, SDO_ORDINATES)
```

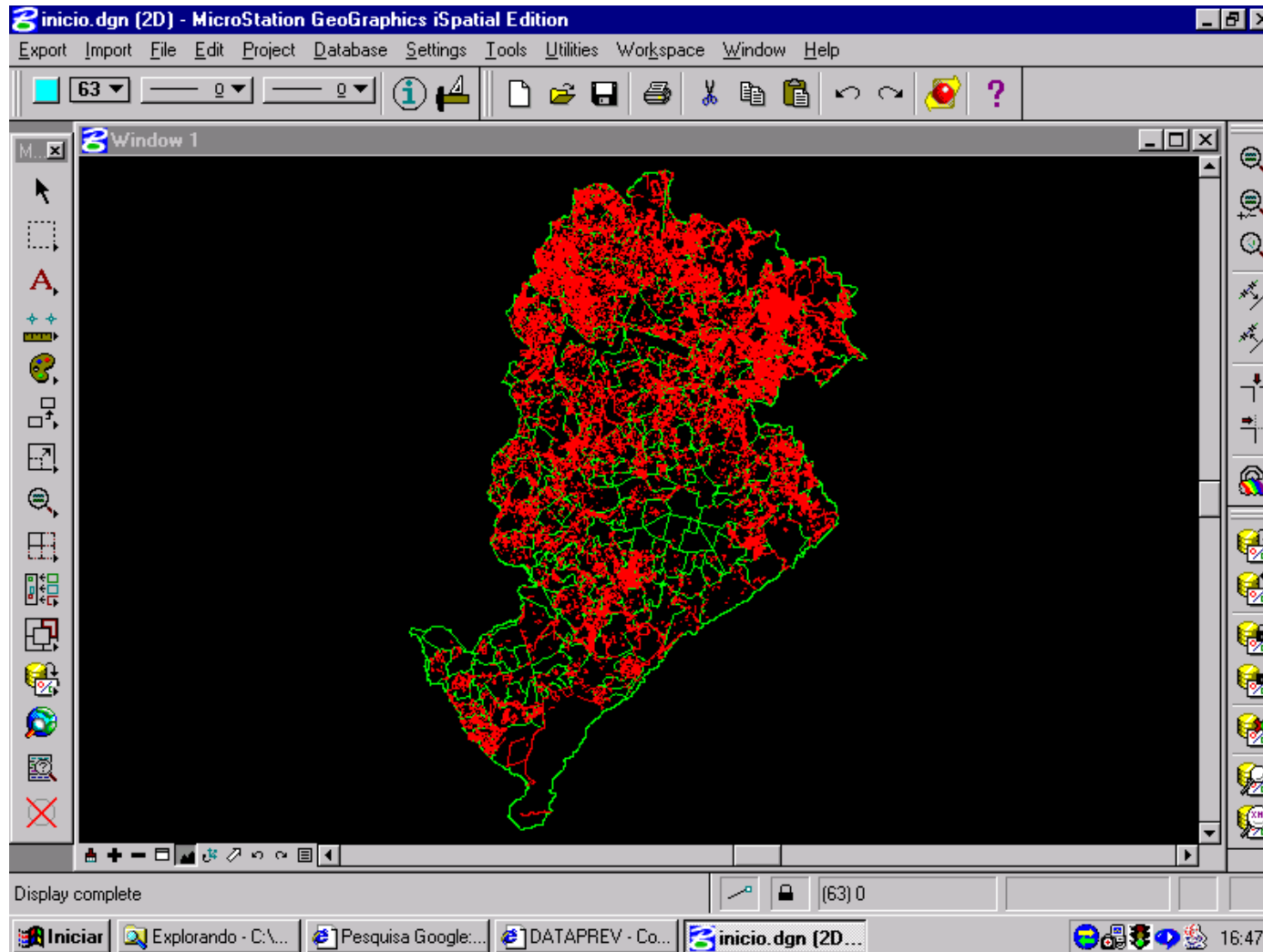
# Visualização MapInfo



# Visualização Geomedia



# Visualização Microstation



# PostGIS

Clodoveu Davis



# PostGIS

- O PostGIS é uma extensão geográfica desenvolvida pela empresa canadense Refrations Research sobre o SGBD objeto-relacional PostgreSQL
- O PostGIS acrescenta ao PostgreSQL a possibilidade de codificação e armazenamento de objetos geográficos dentro do padrão OpenGIS e índices espaciais GiST (Generalized Search Tree), além de funções de análise e relacionamento geográfico entre objetos

# PostGIS

- Como é um projeto de software livre, o PostGIS reúne módulos de diversas origens:
  - PostGreSQL ([www.postgresql.org](http://www.postgresql.org))
  - Índices GiST (<http://www.sai.msu.su/~megera/postgres/gist>)
  - Biblioteca PROJ4 para projeções cartográficas (<http://www.remotesensing.org/proj>)
  - Biblioteca de funções geométricas GEOS (<http://geos.refrations.net>)
- O PostGIS tem também uma interface bem ajustada com o software MapServer, usado na publicação de mapas na Web (<http://mapserver.gis.umn.edu>)

# PostGIS

- Tipos geométricos suportados
  - Ponto
  - Linha
  - Polígono
  - Multipontos
  - Multilinhas
  - Multipolígonos
  - Coleções de geometrias

# PostGIS

- Tipos geométricos
  - São especificados usando o formato WKT (Well Known Text)
    - Exemplo: `INSERT INTO gtest (ID, NAME, GEOM) VALUES (1, 'First Geometry', GeomFromText('LINESTRING(23,4 5,6 5,7 8)', -1));`
    - O conteúdo do comando `LINESTRING` especifica uma seqüência de pares de coordenadas

# PostGIS

- A criação de uma tabela com atributo geográfico é feita em duas etapas
  - Criar a tabela:  

```
CREATE TABLE gtest (ID int4, NAME varchar(20));
```
  - Adicionar a coluna de geometria:  

```
SELECT AddGeometryColumn(' ', ' gtest', 'geom',
-1, 'LINESTRING', 2);
```
- Versões anteriores permitiam criar a tabela em um só passo, mas o procedimento acima é exigido para o atendimento aos padrões da OGC

# PostGIS

- Representações WKT
  - Ponto: POINT(x y)
  - LINESTRING (x1 y1, x2 y2, ..., xn yn)
  - POLYGON (x1 y1, x2 y2, ..., xn yn)
    - Obs: o último vértice não precisa repetir o primeiro
  - MULTIPOINT (x1 y1, x2 y2, ..., xn yn)
  - MULTIPOLYGON ((polygon1), (polygon2), ..., (polygonn))
  - GEOMETRYCOLLECTION(POINT(0 0), POLYGON(1 1, 2 2, 3 3)...) )
- Existe uma variação mais compacta: WKB (Well-known binary)

# PostGIS

- Criação de geometria a partir de WKT/  
WKB

```
geometry = GeomFromWKB(bytea WKB, SRID);
```

```
geometry = GeometryFromText(text WKT, SRID);
```

- Conversão da geometria a WKT/WKB

```
bytea WKB = asBinary(geometry);
```

```
text WKT = asText(geometry);
```

# PostGIS

- Inserção de objetos geográficos em tabelas

```
INSERT INTO SPATIALTABLE (
 THE_GEOM,
 THE_NAME
)
VALUES (
 GeomFromText(' POINT(-126.4 45.32)' , 312),
 ' A Place'
)
```



# PostGIS

- Cada objeto, assim como no caso do Oracle, possui um atributo que indica o sistema de referência espacial (SRID)
- O PostGIS inclui uma tabela contendo as características dos SRID suportados

```
CREATE TABLE SPATIAL_REF_SYS (
 SRID INTEGER NOT NULL PRIMARY KEY, //Número do SRID
 AUTH_NAME VARCHAR(256), //Authority
 AUTH_SRID INTEGER, //ID do SRID segundo a autoridade
 SRTEXT VARCHAR(2048), //Representacao WKT do SRID
 PROJ4TEXT VARCHAR(2048)); //definicao do SRID cf. PROJ4
```

# PostGIS

- Exemplo de carga

```
BEGIN;
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(1,GeomFromText ('LINESTRING(191232 243118,191108 243242)' , -1) , 'Jeff Rd');
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(2,GeomFromText ('LINESTRING(189141 244158,189265 244817)' , -1) , 'Geordie Rd');
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(3,GeomFromText ('LINESTRING(192783 228138,192612 229814)' , -1) , 'Paul St');
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(4,GeomFromText ('LINESTRING(189412 252431,189631 259122)' , -1) , 'Graeme Ave');
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(5,GeomFromText ('LINESTRING(190131 224148,190871 228134)' , -1) , 'Phil Tce');
INSERT INTO ROADS_GEOG (ID,GEOM,NAME) VALUES
(6,GeomFromText ('LINESTRING(198231 263418,198213 268322)' , -1) , 'Dave Cres');
COMMIT;
```

# PostGIS

- Exemplo de consulta textual

```
db=# SELECT id, AsText(geom) AS geom, name FROM ROADS_GEOM;
```

| id | geom                                    | name       |
|----|-----------------------------------------|------------|
| 1  | LINESTRING(191232 243118,191108 243242) | Jeff Rd    |
| 2  | LINESTRING(189141 244158,189265 244817) | Geordie Rd |
| 3  | LINESTRING(192783 228138,192612 229814) | Paul St    |
| 4  | LINESTRING(189412 252431,189631 259122) | Graeme Ave |
| 5  | LINESTRING(190131 224148,190871 228134) | Phil Tce   |
| 6  | LINESTRING(198231 263418,198213 268322) | Dave Cres  |
| 7  | LINESTRING(218421 284121,224123 241231) | Chris Way  |

(6 rows)

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R1 – Restrição de preenchimento do plano
  - Estabelecer uma tabela contendo polígonos para servir como delimitação da área de interesse
  - Comparar o retângulo envolvente mínimo agregado dessa tabela com o retângulo envolvente de cada tabela correspondente a geocampo
  - Esta restrição pode ser relaxada

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R2 – Isolinhas
  - Gatilho: uma isolinha não pode ter interseção com outras
  - A restrição de linha pode ser implementada com CHECK (vide R9)

```

CREATE OR REPLACE FUNCTION VerificaIntersecao()
RETURNS TRIGGER AS
$BODY$
DECLARE
RESULT BOOLEAN;
BEGIN
 RESULT := SELECT ST_INTERSECTS(L.GEOM, T.GEOM)
 FROM NEW L, TabIsolinha T
IF RESULTADO THEN
 RAISE EXCEPTION 'Uma isolinha não pode interceptar outras';
ELSE
 RETURN NEW;
END IF
END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER VerificaIsolinha
 BEFORE INSERT ON tab_isolinha
 FOR EACH ROW EXECUTE PROCEDURE Verifica_intersecao()

```

Clodoveu Davis

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R3 – Tesselação
  - Não existe o tipo GeoRaster no PostGIS
    - Existem projetos em andamento para incorporar este tipo: WKTRaster, PGRaster
  - Pode-se testar os limites geográficos de uma imagem em relação ao retângulo envolvente da área de interesse, mas isso é a R1
  - Por natureza, uma imagem digital ou um grid regular atendem à restrição

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R4 – Subdivisão planar
  - R4a: relacionamento topológico entre polígonos é TOUCHES ou DISJOINT apenas
  - R4b: relacionamento topológico COVERS ou WITHIN entre a delimitação da área de interesse e cada polígono
  - R4c: relacionamento topológico EQUALS entre a união de todos os polígonos e a tabela de delimitação da área de interesse (se existir)
  - Implementar usando gatilhos
  - Alto custo computacional na verificação da integridade: comparação de todos os polígonos entre si (esp. R4c)



```

CREATE OR REPLACE FUNCTION Verifica_Regiao_Municipio_R4a()
RETURNS VOID AS
$BODY$
DECLARE
RESULTADO BOOLEAN;
BEGIN
RESULTADO := SELECT (ST_TOUCHES(M1.GEOM, M2.GEOM) OR ST_DISJOINT(M1.GEOM, M2.GEOM))
 FROM MUN M1, MUN M2
 WHERE M1.ID > M2.ID

IF RESULTADO THEN
 RAISE EXCEPTION 'Geometria inconsistente entre municipios para subdivis o planar'
ELSE
 RETURN NOTICE 'Subdivis o planar de Regi o em Munic pio est  correta.';
END IF

END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER Verifica_Subdiv_Reg_Mun
 BEFORE INSERT
 ON OCORRENCIA
 FOR EACH ROW
 EXECUTE PROCEDURE Verifica_Regiao_Municipio_4a();

```

Clodoveu Davis

```

CREATE OR REPLACE FUNCTION Verifica_Regiao_Municipio_R4b()
RETURNS VOID AS
$BODY$
DECLARE
RESULTADO BOOLEAN;
BEGIN
RESULTADO := SELECT (ST_TOUCHES(M1.GEOM, M2.GEOM) OR ST_DISJOINT(M1.GEOM, M2.GEOM))
 FROM MUN M1, MUN M2
 WHERE M1.ID > M2.ID

IF RESULTADO THEN
 RAISE EXCEPTION 'Geometria inconsistente entre municipios para subdivis o planar'
ELSE
 RETURN NOTICE 'Subdivis o planar de Regi o em Munic pio est  correta.';
END IF

END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER Verifica_Subdiv_Reg_Mun
 BEFORE INSERT
 ON OCORRENCIA
 FOR EACH ROW
 EXECUTE PROCEDURE Verifica_Regiao_Municipio_4a();

```

Clodoveu Davis

```

CREATE OR REPLACE FUNCTION Verifica_Regiao_Municipio_R4b()
RETURNS VOID AS
$BODY$
DECLARE
RESULTADO BOOLEAN;
BEGIN
RESULTADO := SELECT ST_EQUALS (R.GEOM, ST_UNION (M.GEOM))
 FROM REGIAO R, MUNICIPIO M
 WHERE ST_COVERS (R.GEOM, M.GEOM)
 GROUP BY R.GEOM

IF RESULTADO THEN
 RAISE EXCEPTION 'Geometria inconsistente entre municipios para subdivisão planar'
ELSE
 RETURN NOTICE 'Subdivisão planar de Região em Município está correta.';
END IF

END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER Verifica_Subdiv_Reg_Mun
 BEFORE INSERT
 ON OCORRENCIA
 FOR EACH ROW
 EXECUTE PROCEDURE Verifica_Regiao_Municipio_4b();

```

Clodoveu Davis

```

CREATE OR REPLACE FUNCTION Verifica_Regiao_Municipio_R4c()
RETURNS VOID AS
$BODY$
DECLARE
RESULTADO BOOLEAN;
BEGIN
RESULTADO := SELECT ST_EQUALS (R.GEOM, ST_UNION(M.GEOM))
 FROM MUN M, REGIAO R
 GROUP BY R.GEOM

IF RESULTADO THEN
 RETURN NOTICE 'Subdivisão planar de Região em Município está correta.';
ELSE
 RAISE EXCEPTION 'Geometria inconsistente para subdivisão planar em municípios'
END IF

END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER Verifica_Subdiv_Reg_Mun
 BEFORE INSERT
 ON OCORRENCIA
 FOR EACH ROW
 EXECUTE PROCEDURE Verifica_Regiao_Municipio_4c();

```

Clodoveu Davis

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R5 – Triangulação
  - Os triângulos são indicados em tuplas contendo a identificação de três vértices, chaves estrangeiras para uma outra tabela, contendo os vértices propriamente ditos
  - As arestas não são explicitadas
  - R5a: arestas de triângulos não podem ter interseção, exceto nos vértices; testar com funções topológicas
  - R5b: cada vértice precisa estar conectado a pelo menos duas arestas

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R6 – Rede arco-nó
  - Esta restrição é implementada usando duas restrições de integridade referenciais entre a tabela de arcos e a tabela de nós

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R7 – Rede arco-arco
  - Esta restrição de integridade é implementada usando uma restrição de integridade referencial na tabela de arcos
  - Como não se sabe o número de arcos aos quais um arco qualquer estará relacionado, é necessário criar uma tabela de relacionamento, com duas colunas para identificadores de arcos e duas restrições de integridade reerenciais

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R8 – Agregação espacial
  - Vide R4 – subdivisão planar



# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R9 – Linhas
  - Usar CHECK com `IsSimple`

```
ALTER TABLE tab
ADD CONSTRAINT ck_lin
CHECK(IsSimple (geom))
```

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R10 – Polígonos
  - Usar CHECK com IsValid

```
ALTER TABLE tab
ADD CONSTRAINT ck_pol
CHECK (IsValid (geom))
```

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- R11 – Regiões poligonais
  - MULTIPOLYGON
  - Usar CHECK com IsValid

```
ALTER TABLE tab
```

```
ADD CONSTRAINT ck_mpol
CHECK (IsValid (geom))
```

# Restrições de Integridade Espaciais no Esquema Físico PostGIS

- RT – Topológicas
  - Criar gatilhos
  - Em alguns casos é possível criar checks, dependendo da complexidade da restrição

```

CREATE OR REPLACE FUNCTION Verifica_Ocorrencia_in_CompanhiaPM()
RETURNS TRIGGER AS
$BODY$
DECLARE
RESULTADO BOOLEAN;
BEGIN
RESULTADO := SELECT ST_WITHIN(B.GEOM, M.GEOM)
 FROM NEW B, COMPANHIAPM M

IF RESULTADO THEN
 RETURN NEW;
ELSE
 RAISE EXCEPTION 'Ocorrência (%) não está contido em uma Companhia da PM',
 new.nomeocorrencia;
END IF

END;
$BODY$
LANGUAGE 'plpgsql'

CREATE TRIGGER Verifica_Ocorrencia_in_CompanhiaPM
 BEFORE INSERT
 ON OCORRENCIA
 FOR EACH ROW
 EXECUTE PROCEDURE Verifica_Ocorrencia_in_CompanhiaPM();

```

Clodoveu Davis

Clodoveu Davis