

Cenário 14 – Mostre todos os municípios de Pernambuco que não são vizinhos de Recife.

SQL em Oracle Spatial:

```

SELECT m1.nomemunicip
FROM ibge_municipios ne m1, ibge_municipios ne m2
WHERE NOT (SDO_ANYINTERACT(m1.geometria,m2.geometria)='TRUE')
AND m1.nomeuf='PERNAMBUCO'
AND m2.nomemunicip='RECIFE'
AND m1.nomemunicip<>'RECIFE'
ORDER BY m1.nomemunicip;

```

Nomemunicip
ABREU E LIMA
AFOGADOS DA INGAZEIRA
AFRANIO
AGRESTINA
AGUA PRETA
AGUAS BELAS

SQL em PostGIS:

```

1 SELECT m1.nomemunicip
2 FROM ibge_municipios ne m1, ibge_municipios ne m2
3 WHERE ST_DISJOINT(m1.the_geom,m2.the_geom)='T'
4 AND m1.nomeuf='PERNAMBUCO'
5 AND m2.nomemunicip='RECIFE'
6 AND m1.nomemunicip<>'RECIFE'
7 ORDER BY m1.nomemunicip;

```

nomemunicip character varying(40)
1 ABREU E LIMA
2 AFOGADOS DA INGAZEIRA
3 AFRANIO
4 AGRESTINA

Cenário 15 – Encontrar as cidades cujas geometrias sobrepõem a geometria de um buffer de 100KM na LT 04S1-AGL/RIB. Para esta consulta deve-se utilizar o operador OVERLAPS.

SQL em Oracle Spatial:

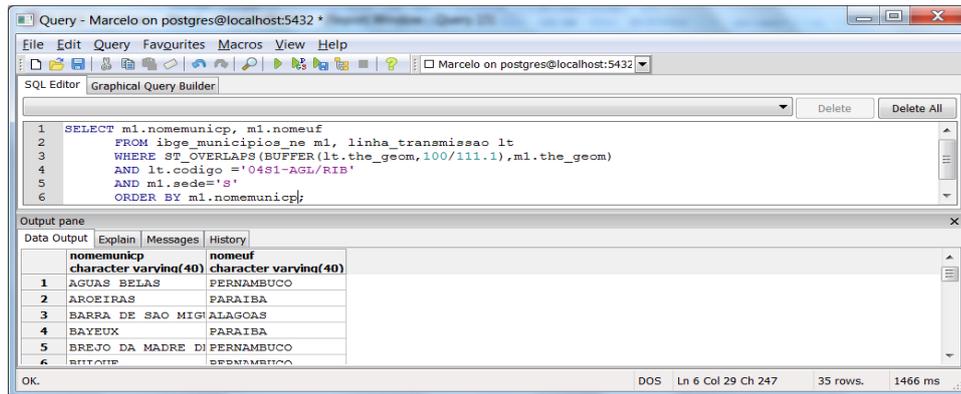
```

SELECT m1.nomemunicip, m1.nomeuf
FROM ibge_municipios ne m1, linha_transmissao lt
WHERE SDO_OVERLAPS(m1.geometria,SDO_GEOM.SDO_BUFFER(lt.geometria,100,
0.5,'UNIT=km')) = 'TRUE'
AND lt.codigo ='04S1-AGL/RIB'
AND m1.sede='s'
ORDER BY m1.nomemunicip;

```

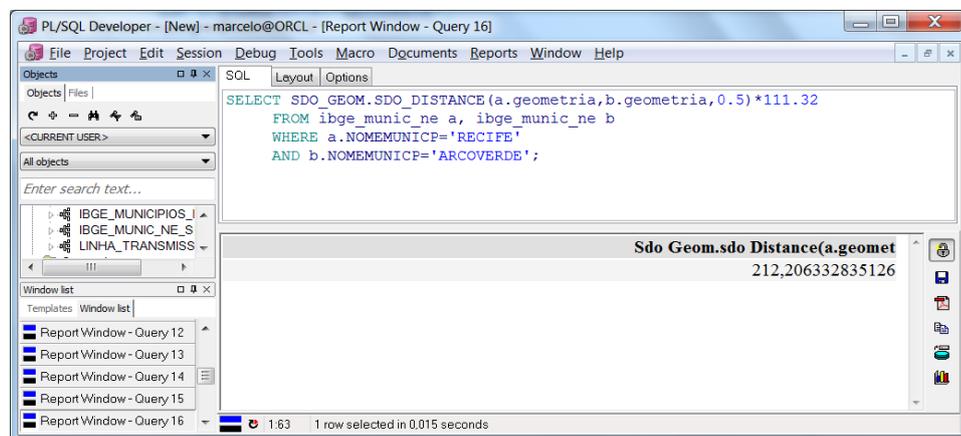
Nomemunicip	Nomeuf
AGUAS BELAS	PERNAMBUCO
AROEIRAS	PARAIBA
BARRA DE SAO MIGUEL	ALAGOAS
BAYEUX	PARAIBA
BREJO DA MADRE DE DEUS	PERNAMBUCO
BUIQUE	PERNAMBUCO

SQL em PostGIS:

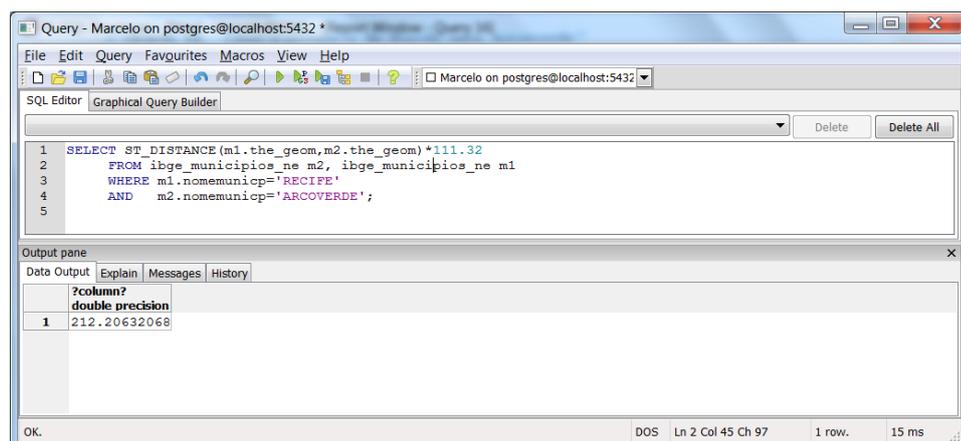


Cenário 16 – Qual a distância de Recife para Arcoverde?

SQL em Oracle Spatial:

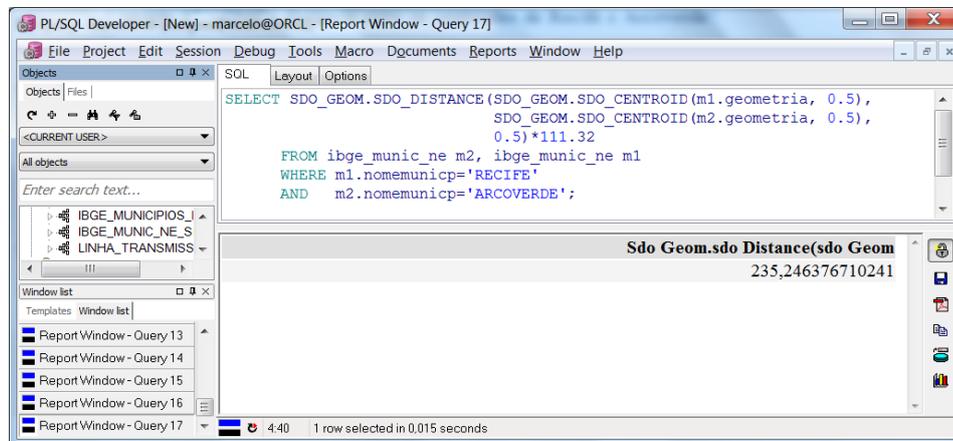


SQL em PostGIS:

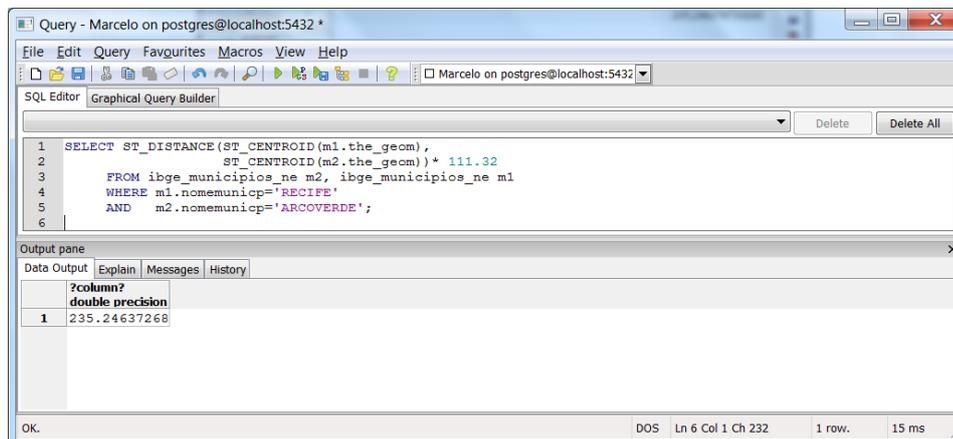


Cenário 17 – Qual a distância entre os centróides de Recife e Arcoverde?

SQL em Oracle Spatial:



SQL em PostGIS:



3.2.1 Resultados e Análises

Os gráficos seguintes foram construídos, utilizando-se os valores de tempo obtidos nos resultados da execução das consultas espaciais dos cenários de 1 a 17 propostos na seção 3.2. Cada consulta foi executada diversas vezes para obtenção do tempo médio em segundos.

O SRID usado para o PostGIS foi o 29101 (SAD69 Brazil Policonic) e o SRID usado para o Oracle Spatial foi o 8292 (SAD69 Oracle).

Para os Cenários 1,3,7,8,16,17 foi usado nas consultas o SRID 29101 (Brazil Policonic) tanto para o PostGIS quanto para o Oracle, pois quando utilizado o SRID 8292 para o Oracle Spatial os resultados divergiram.

No Cenário 1 (Gráfico 1) observou-se que o tempo de consulta no Oracle Spatial foi bastante elevado em relação ao PostGIS. Sendo a consulta executada várias vezes tanto no Oracle como no PostGIS, os tempos médios obtidos ficaram em torno de 2 seg. (Oracle

Spatial) e 0,05 seg. (PostGIS). Entretanto a execução das consultas dos Cenários 16 e 17, que utilizam o mesmo operador (SDO_DISTANCE no Oracle Spatial), apresentou um tempo médio mais satisfatório coincidindo com o tempo médio do PostGIS. Para esse fato não foi encontrada uma explicação.

No Cenário 7 foram empregados para a consulta no Oracle Spatial os operadores SDO_RELATE (usando a máscara ANYINTERACT) e SDO_BUFFER. Para a consulta no PostGIS foram empregados os operadores ST_INTERSECTS e BUFFER. Nesse caso, observa-se no Gráfico 2 que o tempo médio de consulta para o PostGIS (aprox. 1,2 seg.) foi bastante elevado mostrando um desempenho menor que o do Oracle Spatial (aprox. 0,08 seg.).

No Cenário 9 foram utilizados os mesmos operadores SDO_RELATE com máscara ANYINTERACT sem o SDO_BUFFER para o Oracle Spatial e o ST_INTERSECTS também sem o BUFFER para o PostGIS e como observado no Gráfico 3, o tempo médio de consulta para o Oracle Spatial (aprox. 1,6 seg.) foi bastante elevado em relação ao tempo médio do PostGIS (aprox. 0,02 seg.). Neste caso o PostGIS mostrou-se mais eficiente. Vale ressaltar que estas consultas, tanto no Oracle quanto no PostGIS, foram executadas em colunas espaciais indexadas.

No Cenário 15 foram empregados para a consulta no Oracle Spatial os operadores SDO_OVERLAPS e o SDO_BUFFER. Para a consulta no PostGIS foram empregados os operadores ST_OVERLAPS e o BUFFER. Nesse caso, observa-se no Gráfico 4 que o tempo médio de consulta para o PostGIS (aprox. 1,5 seg.) foi muito alto apresentando um desempenho menor que o Oracle Spatial (aprox. 0.5 seg.).

Nas demais consultas não foram apresentadas grandes discrepâncias, conforme se pode observar nos Gráficos de 1 a 4. Observa-se também que, na maioria dos Cenários apresentados, o PostGIS executou as consultas em menor tempo médio em relação ao Oracle Spatial.

Com relação à indexação espacial, todas as tabelas foram indexadas nas suas colunas espaciais. Os operadores espaciais utilizados pelo Oracle Spatial tais como, SDO_RELATE e SDO_inside, só podem ser executados em colunas espaciais indexadas. Funções tais como SDO_DISTANCE e SDO_UNION, podem ser executadas em colunas não indexadas. As funções do PostGIS fizeram uso automaticamente dos índices disponíveis nas geometrias.

TEMPO DE EXECUÇÃO DAS QUERIES

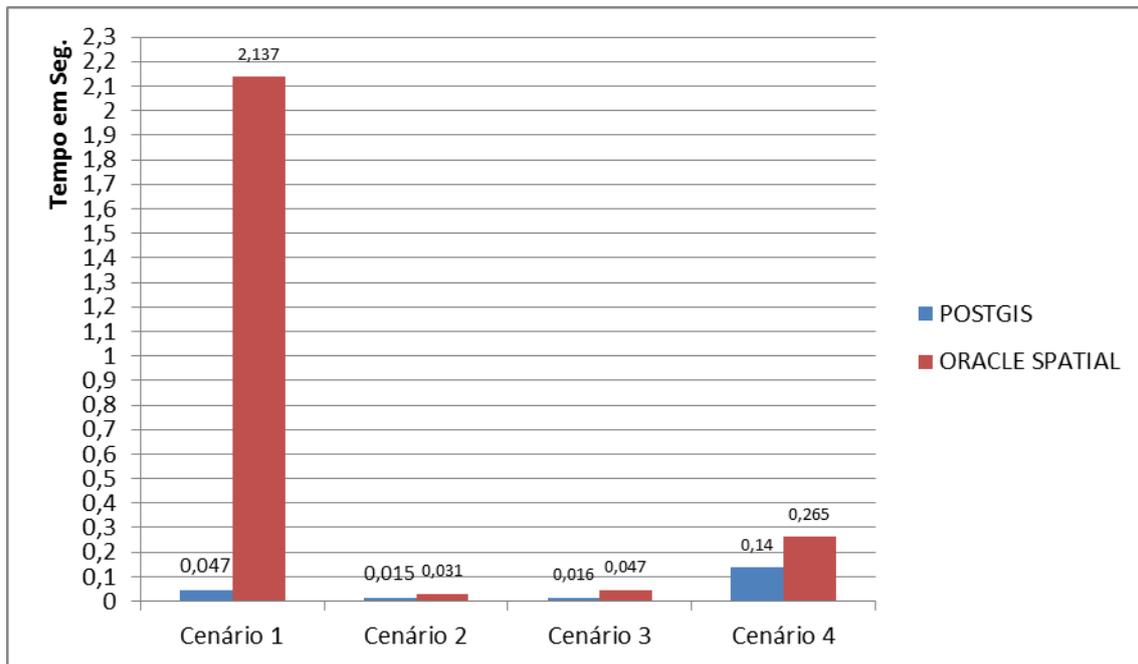


GRÁFICO 1 - CENÁRIOS 1 A 4

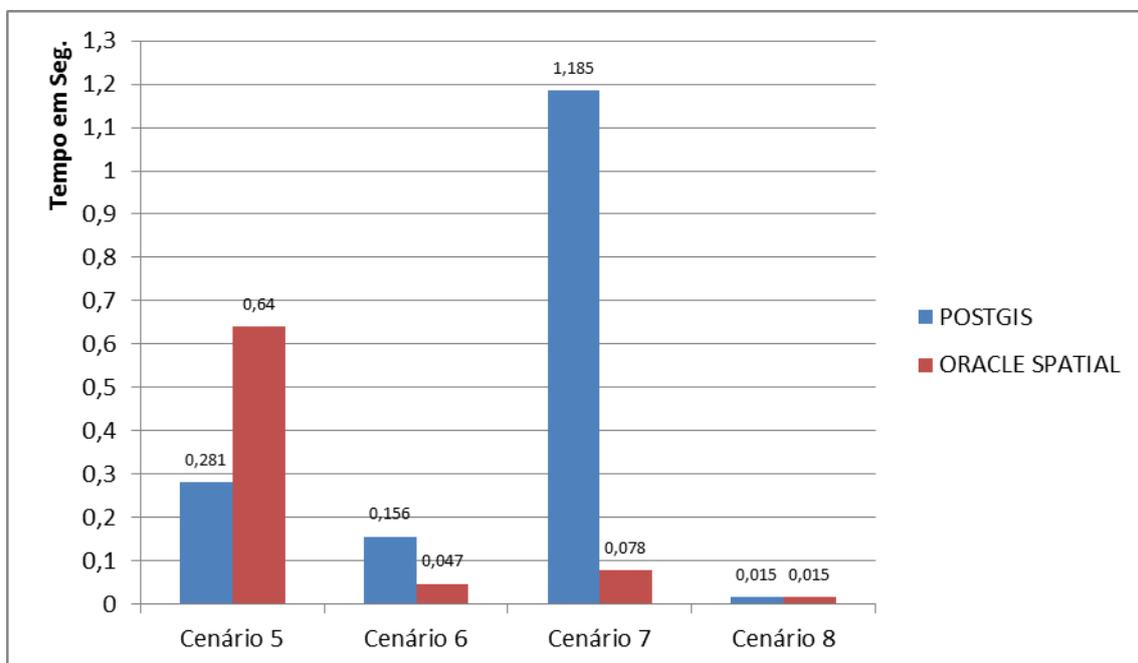


GRÁFICO 2 - CENÁRIOS 5 A 8

TEMPO DE EXECUÇÃO DAS QUERIES

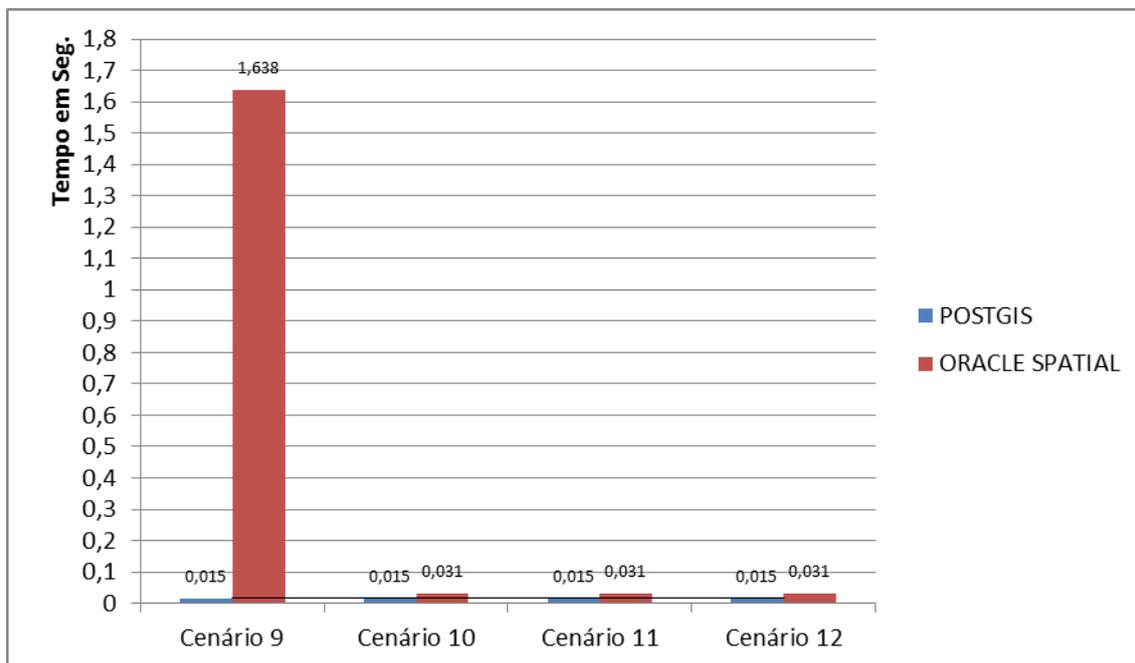


GRÁFICO 3 - CENÁRIOS 9 A 12

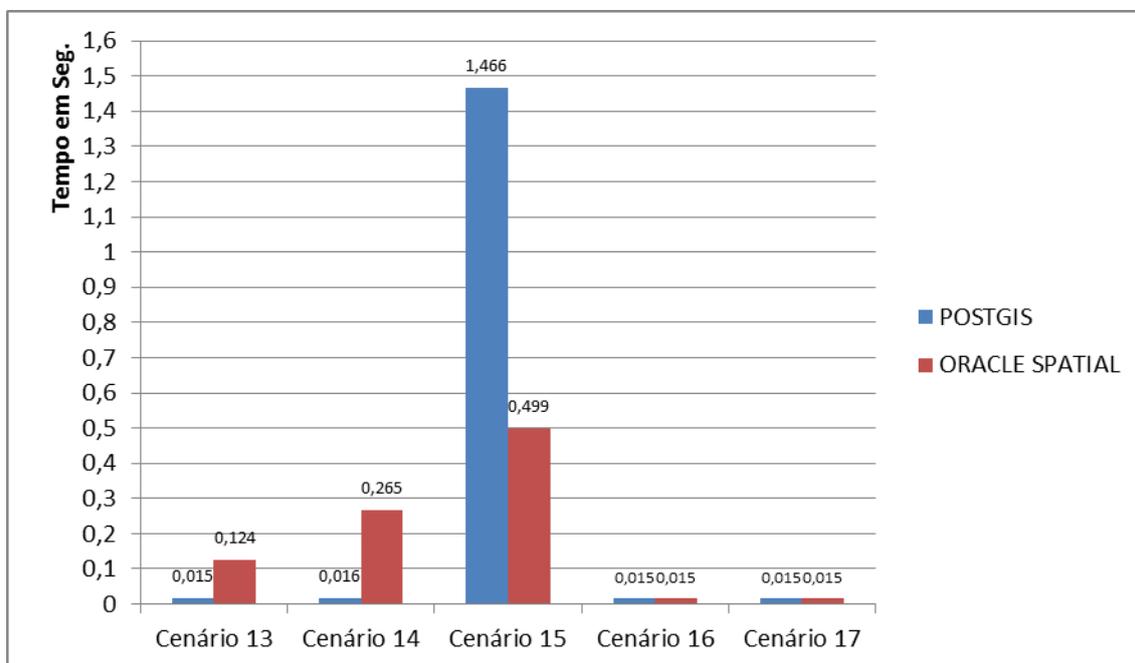


GRÁFICO 4 - CENÁRIOS 13 A 17

4 CONCLUSÃO

O Oracle Spatial é uma extensão espacial disponível em todas as distribuições Oracle. A Enterprise Edition representa uma solução tecnologicamente mais completa incluindo Raster, Geocoding, Mineração de dados, roteamento, dentre outros. O Oracle Spatial está em conformidade com os padrões SFSSQL e SQL/MM 3, embora com total discordância em relação à nomenclatura original das funções e procedimentos, o que traz limitações na interoperabilidade com outros produtos baseados nesses padrões. O Oracle é um SGBD robusto, sendo reconhecidamente um produto líder de mercado em segurança, desempenho, escalabilidade e gerência de dados espaciais armazenados em banco de dados. Possui um conjunto maior de funções de análise espacial em relação aos seus concorrentes.

PostGIS é distribuído gratuitamente sob licença GNU/GPL e oferece ao PostgreSQL o suporte a objetos espaciais, também em conformidade com os padrões SFSSQL e SQL/MM 3, apresentando poucas diferenças. Ao contrário do PostgreSQL/PostGIS, de licença gratuita, a licença do Oracle com sua extensão espacial é paga, tornando-se um grande obstáculo para a maioria dos seus usuários.

Através de estudo e análise das extensões, com a exploração dos recursos espaciais, foi possível avaliar que a utilização do PostGIS torna-se mais vantajosa pelos seguintes aspectos :

- Facilidade de instalação;
- Sintaxe da SQL mais simples, embora as duas extensões estejam em conformidade com os padrões SFSSQL e SQL/MM 3.
- O PostGIS apresentou um melhor desempenho quando comparado com o Oracle Spatial na execução da maioria das QUERIES dos 17 cenários propostos no capítulo 3.

De acordo com o estudo efetuado, o PostgreSQL/PostGIS apresenta-se como uma excelente alternativa técnica e econômica ao Oracle Enterprise Edition com a extensão Oracle Spatial.

A partir dos estudos feitos, pode ser sugerido como trabalho futuro estender este estudo comparativo, incluindo no mesmo mais uma extensão espacial de um SGBD comercial e mais outra extensão espacial de um SGBD de código aberto.

5 REFERÊNCIAS BIBLIOGRÁFICAS

- BAPTISTA, Cláudio. **Bancos de Dados Espaciais**. 2010. Disponível em <<http://www.dsc.ufcg.edu.br/~baptista/cursos/SIG/bdespacial.ppt>>. Acessado em: Fev. 2011.
- CODD, E.F. **The relational model for database management**. Version 2. New York: Addison-Wesley Publishing, 1990.
- DATE, C. J. **Introdução a sistemas de bancos de dados**. 4. ed. Rio de Janeiro: Campus, 1990.674 p.
- DAVIS Jr.(a), Clodoveu A.; QUEIROZ, G. R. Métodos de acesso para dados espaciais. In Casanova, M.A., Camara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (eds.) **Bancos de Dados Geográficos**. Curitiba: MundoGeo, 2005.
- DAVIS Jr.(b), Clodoveu A.; QUEIROZ, G. R. Algoritmos geométricos e relacionamentos topológicos. In Casanova, M.A., Camara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (eds.) **Bancos de Dados Geográficos**. Curitiba: MundoGeo, 2005.
- DAVIS Jr.(c), Clodoveu A., et al. O Open Geospatial Consortium. In Casanova, M.A., Camara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (eds.) **Bancos de Dados Geográficos**. Curitiba: MundoGeo, 2005.
- ELMASRI, E.; NAVATHE, S.– **Sistemas de banco de Dados**, 2005, São Paulo, Editora: PEARSON, 4º Edição.
- EGENHOFER, M.; FRANZOSA, R. **On the Equivalence of Topological Relations**. *International Journal of Geographical Information Systems*, v. 9, n.2, p. 133-152, 1995.
- EGENHOFER, M.; HERRING, J. **Categorizing Binary Topological Relationships Between Regions, Lines, and Points in Geographic Databases**. Orono, ME: Department of Surveying Engineering, University of Maine, 1991.
- FERREIRA, K. R., et.al. Arquiteturas e Linguagens. In Casanova, M.A., Camara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (eds.) **Bancos de Dados Geográficos**. Curitiba: MundoGeo, 2005.
- ISO, 2004, **ISO/IEC 13249-3:1999, Information technology Database languages - SQL Multimedia and Application Packages - Part 3: Spatial**, International Organization For Standardization, 2004.
- LOURENÇO, Paula Moreira Barbosa. **Um Estudo Sobre Recursos de Tratamento de Dados Espaciais em SGBDs Geográficos** – UFMG – 2008.
- MELTON, J.; EISENBERG, A. **SQL Multimedia and Application Packages (SQL/MM)**. SIGMOD Record, 2001.
- MURRAY, C., 2003, Oracle® Spatial User's Guide and Reference 10g Release 1 (10.1), Redwood City, Oracle Corporation, p. 602.
- OGC. **OpenGIS® simple features specification for SQL revision 1.1**, 1999.

ORACLE CORPORATION. **Oracle Spatial Developer's Guide: 11g Release 2 (11.2)**, October 2010. Part Number E11830-07. Disponível em: <http://www.oracle.com/pls/db112/portal.portal_db?selected=7&frame=#oracle_spatial_and_location_information>. Acesso em: Mar. 2011.

CHEN, P. **The entity-relationship model: Toward a unified view of data**, 1976.

PROJETO SMARTVIEW(a): **Relatório Técnico, Modelo de Dados, Etapa 4 – Mês 5**. Campina Grande: UFCG/CEEI/DSC, 2008.

PROJETO SMARTVIEW(b): **Relatório Técnico, Modelo de Dados, Etapa 4 – Mês 8**. Campina Grande: UFCG/CEEI/DSC, 2008.

QUEIROZ, Gilberto Ribeiro; FERREIRA, Karine Reis. **Tutorial sobre Bancos de Dados Geográficos**. Instituto Nacional de Pesquisas Espaciais. GeoBrasil, 2006.

QUEIROZ, G. R., Ferreira, K. R. **SGBD com extensões espaciais**. In Casanova, M.A., Camara, G., Davis Jr., C. A., Vinhas, L., Queiroz, G. R. (eds.) Bancos de Dados Geograficos. Curitiba: MundoGeo, 2005.

RAMSEY, P.; CAVE-AYLAND, M.; NEUFELD, K.; OBE, R. **POSTGIS 1.5.2 Manual**. 2010. Disponível em: <<http://postgis.refractor.net/docs>>. Acesso em: Abr. 2011.

RAVADA, S.; SHARMA, J. Oracle8i Spatial: experiences with extensible databases. In: **International Symposium on Spatial Databases**, 6., jul. 1999, Hong Kong, China. Proceedings... Berlin: Springer-Verlag, 1999, p. 355-359.

RIGAUX, P.; SCHOLL, M.; VOISARD, A. **Spatial databases with application to GIS**. San Francisco: Morgan Kaufman Publishers, 2002.

SILVA, Rosângela. **Banco de Dados Geográfico: Uma Análise das Arquiteturas DUAL(SPRING) e Integrada (RACLE SPATIAL)**. Dissertação (Mestrado em Engenharia) - Universidade de São Paulo – São Paulo, 2002.

SILBERSCHATZ, A.; KORTH, H.F.; SUDARSHAN, S. **Sistema de banco de dados**. 3. ed., São Paulo: Makron Books, 1999.

STOLZE, K. **SQL/MM Spatial: The Standard to Manage Spatial Data in Relational Database Systems**. In: BTW 2003, Datenbanksysteme für Business, Technologie und Web. Leipzig, Alemanha, 2003. p. 247-264.

STONEBRAKER, M. **Object-relational DBMSs: the next great wave** San Francisco Morgan Kaufmann, 1996.

VINHAS, L – **Um sub-sistema extensível para o armazenamento de Geo-Campos em Banco de Dados Geográfico**, 2006 – Tese de Doutorado – INPE

YEUNG, A.K.W.; HALL, G. B. - **Spatial Database Systems Design, Implementation and Project Management**, 2007, Published by Springer, The Netherlands.

ANEXO A - Consultas nas Bases de Dados Espaciais para os Cenários propostos

A seguir estão apresentados os scripts SQL para os cenários propostos, em ambas extensões espaciais comparadas.

As tabelas utilizadas foram carregadas tanto em um banco de dados Oracle Spatial como também em um banco de dados PostGreSQL/PostGis a partir de arquivos shapes. Esses arquivos shapes foram gerados a partir das seguintes tabelas do banco de dados do SmartView (**PROJETO SMARTVIEW(a), 2008**).

1. Ibge_estados_ne (polígono) – contém os estados do Nordeste;
2. ibge_municipios_ne (polígono) – contém os municípios do Nordeste;
3. chesf_subestacoes (ponto) – contém as Subestações da Chesf.
4. linha_transmissao (linha) – contém as Linhas de Transmissão da Chesf.

Para criação das tabelas considera-se as seguintes definições:

```
CREATE TABLE chesf_subestacoes
```

```
(
  gid serial NOT NULL,
  sigla character varying(3),
  nome_da_in character varying(23),
  ger character varying(4),
  dt_energ character varying(8),
  tens character varying(6),
  tipo character varying(6),
  municipio character varying(18),
  uf character varying(2),
  latitude numeric,
  longitude numeric,
  latitude_1 character varying(8),
  longitud_1 character varying(8),
  telefome character varying(15),
  the_geom geometry
);
```

```
CREATE TABLE linha_transmissao
```

```
(
  gid serial NOT NULL,
  codigo character varying(50),
  nome character varying(50),
  tensao character varying(16),
  the_geom geometry
);
```

```
CREATE TABLE ibge_estados_ne
(
  gid serial NOT NULL,
  nomeuf character varying(40),
  count numeric,
  the_geom geometry
);
```

```
CREATE TABLE ibge_municipios_ne
(
  gid serial NOT NULL,
  nomemunicp character varying(40),
  nomemunic character varying(40),
  populacao numeric,
  nomeuf character varying(40),
  regioao character varying(12),
  nomemicro character varying(40),
  area97 numeric,
  longitude numeric,
  latitude numeric,
  sede character varying(1),
  the_geom geometry
);
```

Cenário 1 – Encontrar todos os municípios que estejam a menos de 100 km do Recife.

SQL em Oracle Spatial:

```
SELECT m1.nomemunicp
  FROM ibge_municipios_ne m1, ibge_municipios_ne m2
 WHERE
SDO_GEOM.SDO_DISTANCE(m1.geometria,m2.geometria,0.5,
'UNIT=KM') < 100
  AND m1.nomemunicp<>'RECIFE'
  AND m2.nomemunicp='RECIFE'
 ORDER BY m1.NOMEMUNICP;
```

SQL em PostGIS:

```
SELECT m1.nomemunicp
  FROM ibge_municipios_ne m1, ibge_municipios_ne m2
 WHERE DISTANCE(m1.the_geom,m2.the_geom)< 100/111.32
  AND m1.nomemunicp<>'RECIFE'
  AND m2.nomemunicp='RECIFE'
 ORDER BY m1.nomemunicp;
```

Cenário 2 – Encontre todas as Subestações localizadas no estado de PE.

SQL em Oracle Spatial:

```
SELECT s.nome_da_in, s.uf
      FROM ibge_estados_ne e, chesf_subestacoes s
      WHERE e.nomeuf='PERNAMBUCO'
      AND SDO_RELATE(s.geometria,e.geometria,
                    'MASK=INSIDE+COVEREDBY')='TRUE'
      ORDER BY s.nome_da_in;
```

SQL em PostGIS:

```
SELECT s.nome_da_in, s.uf
      FROM ibge_estados_ne e, chesf_subestacoes s
      WHERE ST_CONTAINS(e.the_geom,s.the_geom)
      AND e.nomeuf = 'PERNAMBUCO'
      ORDER BY s.nome_da_in;
```

Cenário 3 – Mostre todos os municípios que são vizinhos de Recife.

SQL em Oracle Spatial:

```
SELECT m1.nomemunicp
      FROM ibge_municipios_ne m1, ibge_municipios_ne m2
      WHERE m2.nomemunicp='RECIFE'
      AND SDO_RELATE(m1.geometria,m2.geometria,
                    'MASK=TOUCH')='TRUE';
```

SQL em PostGIS:

```
SELECT m1.nomemunicp
      FROM ibge_municipios_ne m1, ibge_municipios_ne m2
      WHERE TOUCHES(m1.the_geom,m2.the_geom)='T'
      AND m2.nomemunicp='RECIFE';
```

Cenário 4 – Quais são as Linha de Transmissão que atravessam o estado de PE ?

SQL em Oracle Spatial:

```
SELECT l.codigo
      FROM ibge_estados_ne e, linha_transmissao l
      WHERE e.nomeuf='PERNAMBUCO'
      AND SDO_RELATE(l.geometria,e.geometria,
                    'MASK=OVERLAPBDYDISJOINT')='TRUE'
      ORDER BY l.codigo;
```

SQL em PostGIS:

```
SELECT l.codigo
      FROM ibge_estados_ne e, linha_transmissao l
      WHERE e.nome='PERNAMBUCO' AND
            CROSSES(l.the_geom,e.the_geom)
      ORDER BY l.codigo;
```

Cenário 5 – Mostrar o comprimento das linhas de transmissão em cada um dos estados pelos quais elas passam.

SQL em Oracle Spatial:

```
SELECT l.codigo, e.nomeuf,
       SDO_GEOM.SDO_LENGTH(SDO_GEOM.SDO_INTERSECTION
                           (l.geometria,e.geometria,0.5),0.5)/1000
      FROM linha_transmissao l, ibge_estados_ne e
      WHERE SDO_RELATE(l.geometria,e.geometria,
                      'MASK=OVERLAPBDYDISJOINT')='TRUE'
      ORDER BY l.codigo;
```

SQL em PostGIS:

```
SELECT l.codigo, e.nomeuf,
       LENGTH(INTERSECTION(l.the_geom,e.the_geom))*111.32
      FROM linha_transmissao l, ibge_estados_ne e
      WHERE CROSSES(l.the_geom,e.the_geom)='T'
      ORDER BY l.codigo;
```

Cenário 6 – Listar as subestações que estão num raio de 10 Km da linha de transmissão 04C7-MRR/GNN.

SQL em Oracle Spatial:

```
SELECT s.nome_da_in
      FROM linha_transmissao l, chesf_subestacoes s
      WHERE SDO_INSIDE(s.geometria,
                      SDO_GEOM.SDO_BUFFER(l.geometria,10,0.05,'UNIT=KM'))='TRUE'
      AND l.codigo='04C7-MRR/GNN';
```

SQL em PostGIS:

```
SELECT s.nome_da_in
      FROM linha_transmissao l, chesf_subestacoes s
      WHERE CONTAINS(BUFFER(l.the_geom,10/111,32),s.the_geom)
      AND l.codigo='04C7-MRR/GNN';
```

Cenário 7 – A LT 04S1-AGL/RIB pode suprir a energia das cidades que estão a um raio de 100km. Encontrar as cidades que consomem energia da LT 04S1-AGL/RIB.

SQL em Oracle Spatial:

```
SELECT m1.nomemunicp, m1.nomeuf
      FROM ibge_municipios_ne m1, linha_transmissao lt
      WHERE SDO_RELATE (m1.geometria,
        SDO_GEOM.SDO_BUFFER(lt.geometria,100,0.5,'UNIT=KM'),
        'mask=ANYINTERACT') = 'TRUE'
      AND lt.codigo ='04S1-AGL/RIB'
      AND m1.sede='S';
```

SQL em PostGIS:

```
SELECT m.nomemunicp, m.nomeuf
      FROM linha_transmissao l, ibge_municipios_ne m
      WHERE
        ST_INTERSECTS (BUFFER(l.the_geom,100/111.0),m.the_geom)
      AND l.codigo ='04S1-AGL/RIB'
      AND m.sede='S';
```

Cenário 8 – Qual a distância da subestação de Camaçari II (SE CMD) para a subestação de Recife II (SE RCD) ?

SQL em Oracle Spatial:

```
SELECT
      SDO_GEOM.SDO_DISTANCE(a.geometria,b.geometria,0.5,'UNIT=KM')
      FROM chesf_subestacoes b, chesf_subestacoes a
      WHERE a.sigla='CMD'
      AND b.sigla='RCD';
```

SQL em PostGIS:

```
SELECT DISTANCE(s1.the_geom,s2.the_geom)*111.32
      FROM chesf_subestacoes s2, chesf_subestacoes s1
      WHERE s1.sigla='CMD'
      AND s2.sigla='RCD';
```

Cenário 9 – Quais são as LT's que passam pelo município de Recife ?

SQL em Oracle Spatial:

```
SELECT lt.codigo
      FROM ibge_municipios_ne m1,linha_transmissao lt
      WHERE m1.nomemunicp = 'RECIFE'
      AND SDO_RELATE(m1.geometria,lt.geometria,
        'MASK=ANYINTERACT')='TRUE';
```

SQL em PostGIS:

```
SELECT lt.codigo
      FROM ibge_municipios_ne m1,linha_transmissao lt
      WHERE m1.nomemunicp = 'RECIFE'
      AND ST_INTERSECTS(m1.the_geom,lt. the_geom);
```

Cenário 10 – Retornar a geometria a partir da união das geometrias dos municípios de ARCOVERDE e RECIFE.

SQL em Oracle Spatial:

```
SELECT U.GEOM.GET_WKT() FROM (SELECT
SDO_GEOM.SDO_UNION(m1.geometria,m.diminfo,m2.geometria,
                    m.diminfo) AS geom
      FROM ibge_municipios_ne m1,ibge_municipios_ne m2,
                    user_sdo_geom_metadata m
      WHERE m.table_name = 'IBGE_MUNICIPIOS_NE'
      AND m.column_name = 'GEOMETRIA'
      AND m1.nomemunicp = 'ARCOVERDE'
      AND m2.nomemunicp = 'RECIFE') u
```

SQL em PostGIS:

```
SELECT ST_AsText(st_union(m1.the_geom))
      FROM ibge_municipios_ne m1
      WHERE m1.nomemunicp = 'ARCOVERDE'
      OR m1.nomemunicp = 'RECIFE');
```

Cenário 11 – Quais são as LT's que passam pelos municípios de ARCOVERDE e RECIFE ?

SQL em Oracle Spatial:

```
SELECT lt.codigo
      FROM linha_transmissao lt,
      (SELECT SDO_GEOM.SDO_UNION(m1.geometria,
                                m.diminfo,m2.geometria,
                                m.diminfo) AS geom
      FROM ibge_municipios_ne m1,
            ibge_municipios_ne m2,
            user_sdo_geom_metadata m
      WHERE m.table_name = 'IBGE_MUNICIPIOS_NE'
      AND m.column_name = 'GEOMETRIA'
      AND m1.nomemunicp = 'ARCOVERDE'
      AND m2.nomemunicp = 'RECIFE') u
      WHERE SDO_RELATE(lt.geometria,u.geom,'MASK=ANYINTERACT')=
      'TRUE';
```

SQL em PostGIS:

```
SELECT lt.codigo
FROM linha_transmissao lt,
     (SELECT ST_UNION(m1.the_geom) as uni
      FROM ibge_municipios_ne m1
      WHERE m1.nomemunicp = 'ARCOVERDE'
            OR m1.nomemunicp = 'RECIFE') m1
WHERE ST_INTERSECTS(lt.the_geom,m1.uni);
```

Cenário 12 – Encontre todas as Subestações localizadas nos municípios de DIAS D’AVILA, TACAIMBO e RECIFE.

SQL em Oracle Spatial:

```
SELECT s1.sigla
FROM ibge_municipios_ne m1, chesf_subestacoes s1
WHERE (m1.nomemunicp = 'DIAS D'AVILA'
      OR m1.nomemunicp = 'TACAIMBO'
      OR m1.nomemunicp = 'RECIFE')
AND SDO_RELATE(s1.geometria,m1.geometria,
'MASK=INSIDE+COVEREDBY')='TRUE';
```

SQL em PostGIS:

```
SELECT s1.sigla
FROM ibge_municipios_ne m1, chesf_subestacoes s1
WHERE (m1.nomemunicp = 'DIAS D'AVILA'
      OR m1.nomemunicp = 'TACAIMBO'
      OR m1.nomemunicp = 'RECIFE')
AND ST_WITHIN(s1.the_geom,m1.the_geom);
```

Cenário 13 – Encontre todas as Subestações localizadas fora da cidade de RECIFE.

SQL em Oracle Spatial:

```
SELECT s.sigla
FROM ibge_municipios_ne m, chesf_subestacoes s
WHERE m.nomemunicp='RECIFE'
AND NOT (SDO_INSIDE(s.geometria,m.geometria)='TRUE');
```

SQL em PostGIS:

```
SELECT s.sigla
FROM ibge_municipios_ne m, chesf_subestacoes s
WHERE m.nomemunicp='RECIFE'
AND NOT (ST_WITHIN(s.the_geom,m.the_geom));
```

Cenário 14 – Mostre todos os municípios de Pernambuco que não são vizinhos de Recife.

SQL em Oracle Spatial:

```
SELECT m1.nomemunicp
      FROM ibge_municipios_ne m1, ibge_municipios_ne m2
      WHERE NOT
            (SDO_ANYINTERACT(m1.geometria,m2.geometria)='TRUE')
      AND m1.nomeuf='PERNAMBUCO'
      AND m2.nomemunicp='RECIFE'
      AND m1.nomemunicp<>'RECIFE';
```

SQL em PostGIS:

```
SELECT m1.nomemunicp
      FROM ibge_municipios_ne m1, ibge_municipios_ne m2
      WHERE ST_DISJOINT(m1.the_geom,m2.the_geom)='T'
      AND m1.nomeuf='PERNAMBUCO'
      AND m2.nomemunicp='RECIFE';
```

Cenário 15 – Encontrar as cidades cujas geometrias sobrepõem a geometria de um buffer de 100KM na LT 04S1-AGL/RIB. Para esta consulta deve-se utilizar o operador OVERLAPS.

SQL em Oracle Spatial:

```
SELECT m1.nomemunicp, m1.nomeuf
      FROM ibge_municipios_ne m1, linha_transmissao lt
      WHERE SDO_OVERLAPS (m1.geometria,
            SDO_GEOM.SDO_BUFFER(lt.geometria,100,0.5,'UNIT=km')) =
            'TRUE'
      AND lt.codigo ='04S1-AGL/RIB'
      AND m1.sede='S';
```

SQL em PostGIS:

```
SELECT m1.nomemunicp, m1.nomeuf
      FROM ibge_municipios_ne m1, linha_transmissao lt
      WHERE
            ST_OVERLAPS (BUFFER(lt.the_geom,100/111.1),m1.the_geom)
      AND lt.codigo ='04S1-AGL/RIB'
      AND m1.sede='S';
```

Cenário 16 – Qual a distância de Recife para Arcoverde?**SQL em Oracle Spatial:**

```
SELECT SDO_GEOM.SDO_DISTANCE(a.geometria,b.geometria,
                             0.5, 'UNIT=KM')
FROM   ibge_municipios_ne b, ibge_municipios_ne a
WHERE  a.NOMEMUNICP='RECIFE' AND
       b.NOMEMUNICP='ARCOVERDE';
```

SQL em PostGIS:

```
SELECT ST_DISTANCE(m1.the_geom,m2.the_geom) * 111.32
FROM   ibge_municipios_ne m2, ibge_municipios_ne m1
WHERE  m1.nomemunicp='RECIFE'
AND    m2.nomemunicp='ARCOVERDE';
```

Cenário 17 – Qual a distância entre os centróides de Recife e Arcoverde?**SQL em Oracle Spatial:**

```
SELECT SDO_GEOM.SDO_DISTANCE(SDO_GEOM.SDO_CENTROID
                             (m1.geometria, 0.5),SDO_GEOM.SDO_CENTROID
                             (m2.geometria, 0.5),0.5, 'UNIT=KM')
FROM   ibge_municipios_ne m2, ibge_municipios_ne m1
WHERE  m1.nomemunicp='RECIFE'
AND    m2.nomemunicp='ARCOVERDE';
```

SQL em PostGIS:

```
SELECT ST_DISTANCE(st_CENTROID
                  (m1.the_geom),ST_CENTROID
                  (m2.the_geom)) * 111.32
FROM   ibge_municipios_ne m2, ibge_municipios_ne m1
WHERE  m1.nomemunicp='RECIFE'
AND    m2.nomemunicp='ARCOVERDE';
```

ANEXO B – Sistema SmartView

1. Descrição do SmartView

Atualmente existe na Chesf um sistema de visualização de informações, em um ambiente geográfico, que fornece suporte à Operação do Sistema Elétrico. Tais informações são compostas por informações espaciais e não espaciais. Esse sistema, denominado SmartView, tem como objetivo integrar as informações de diversos sistemas (SAGE, Sistema de tratamento de alarmes, Oscilografia e Qualimetria) os quais estão envolvidos no processo de monitoramento da rede elétrica da Chesf. O SmartView utiliza-se de uma base de dados georeferenciada (base GIS), fornecendo informações sobre o estado do sistema elétrico que auxiliará melhor às equipes de gerência e operação do COOS (Centro de Operação do Sistema). A área de abrangência do sistema é a região Nordeste onde atua a Chesf.

É possível, através do SmartView, visualizar informações espaciais em mapas temáticos tais como estados, municípios, usinas, subestações, torres, linhas de transmissão, localização de faltas em linhas de transmissão, rios e reservatórios.

2. O Banco de Dados do SmartView

De acordo com o relato em **PROJETO SMARTVIEW(a) (2008)**: Relatório Técnico, Modelo de Dados, Etapa 4 – Mês 5 (2008, p. 17)

A base de dados do Projeto SmartView pode ser dividida em três partes: segurança, hidrografia e sistema elétrico. A primeira parte contém as tabelas usadas para o controle de acesso dos usuários ao sistema. A parte de hidrografia armazena dados sobre os recursos hidrográficos da empresa e do seu sistema de medição de cotas. Já a parte do sistema elétrico contém todas as informações sobre os equipamentos e centros regionais da empresa, além dos dados colhidos a partir do sistema de oscilografia, da rede de qualimetria e do sistema de tratamento de alarmes.

O SGBD do SmartView é o PostGresSQL mais a extensão PostGIS com funcionalidades espaciais. As informações espaciais que alimentam a base de dados do SmartView estão no padrão ESRI(*Environmental System Research Institute*) Shapefile e são em dois grupos, quais sejam : CHESF (hidrelétricas, linhas de transmissão, torres, subestações, usinas hidrelétricas e térmicas) e do IBGE (áreas urbanas, estados do nordeste, ferrovias, rodovias, hidrografia e municípios). Uma ferramenta para carga dos arquivos Shapefile no banco de dados existe para prover suporte a esse tipo de arquivo.

O PostGresSQL possui um programa chamado *shp2pgsql* que através desse programa é possível extrair os dados dos arquivos shapefile e inserir no banco de dados **(PROJETO SMARTVIEW(b) (2008): Relatório Técnico, Modelo de Dados, Etapa 4 – Mês 8, dezembro de 2008, p. 6).**