

IBM[®] DB2[®] Spatial Extender



Referência e Manual do Usuário

Versão 7

IBM[®] DB2[®] Spatial Extender



Referência e Manual do Usuário

Versão 7

Observação!

Antes de utilizar estas informações e o produto a que elas se referem, leia as instruções gerais em “Avisos” na página 361.

Segunda Edição (junho de 2001)

Esta edição aplica-se à Versão 7, Release 2 do IBM DB2 Spatial Extender e a todos os releases e modificações subsequentes até que seja indicado de outra forma em novas edições.

© Copyright International Business Machines Corporation 1998, 2001. Todos os direitos reservados.

Índice

| | | | |
|--|------------|--|-----------|
| Figuras | vii | Considerações pós-instalação | 29 |
| Tabelas | ix | Efetuando o download do ArcExplorer | 29 |
| Sobre este manual | xi | Utilização dos CD-ROMs para Dados e | |
| Quem deveria ler este manual | xi | Mapas e Dados de Referência do Geocoder | |
| Convenções | xi | do DB2 Spatial Extender | 30 |
| Como enviar seus comentários | xi | Invocação do Spatial Extender | 32 |
| <hr/> | | | |
| Parte 1. Utilizando o Spatial | | Capítulo 3. Configurando recursos | 33 |
| Extender | 1 | Inventário dos recursos | 33 |
| Capítulo 1. Sobre o Spatial Extender | 3 | Dados de referência | 33 |
| A finalidade do Spatial Extender | 3 | Recursos que ativam um banco de dados | |
| Dados que representam recursos geográficos | 4 | para operações espaciais | 34 |
| Como os dados representam os recursos | | Ativando um banco de dados para operações | |
| geográficos. | 4 | espaciais | 34 |
| A natureza dos dados espaciais | 6 | Criando um sistema de referência espacial | 35 |
| De onde vêm os dados espaciais | 6 | Sobre sistemas de coordenadas e de | |
| Como criar e usar um GIS do Spatial Extender | 8 | referência espacial | 35 |
| Interfaces para o Spatial Extender e | | Criando um sistema de referência espacial | |
| funcionalidade associada | 9 | a partir do Centro de Controle | 40 |
| Tarefas que você executa para criar e usar | | Capítulo 4. Definindo colunas espaciais, | |
| um GIS do Spatial Extender | 9 | registrando-as como camadas e ativando | |
| Cenário: Uma companhia de seguros | | um geocoder para mantê-las. | 45 |
| atualiza seu GIS. | 12 | Sobre tipos de dados espaciais | 45 |
| Capítulo 2. Instalando o Spatial Extender | 17 | Tipos de dados para recursos de uma | |
| Configuração do DB2 Spatial Extender | 17 | única unidade | 46 |
| Exigências do sistema | 18 | Tipos de dados para recurso de | |
| Sistemas operacionais suportados | 18 | multi-unidades | 47 |
| Software de banco de dados exigido | 18 | Um tipo de dados para todos os recursos | 48 |
| Requisitos de espaço em disco | 19 | Definindo uma coluna espacial para uma | |
| Instalação do DB2 Spatial Extender para | | tabela, registrando esta coluna como uma | |
| Windows NT e Windows 2000 | 20 | camada e ativando um geocoder para | |
| Instalação do DB2 Spatial Extender para AIX | 22 | mantê-la | 48 |
| Montagem do CD-ROM | 22 | Restrições | 50 |
| Utilização do SMIT ou do comando | | Registrando uma coluna de exibição como | |
| installp. | 25 | uma camada | 50 |
| Estabelecimento do ambiente da instância | | Capítulo 5. Preenchendo colunas espaciais | 53 |
| do DB2 Spatial Extender | 26 | Utilizando os geocoders | 53 |
| Verificando a instalação | 26 | Sobre geocodificação | 53 |
| Dicas de detecção de problemas do | | Executando o geocoder no modo batch | 57 |
| programa de amostra | 27 | Importando e exportando dados | 58 |
| | | Sobre importação e exportação | 59 |

| | | | |
|---|------------|--|------------|
| Importação de dados para uma tabela nova ou existente a partir do nível do banco de dados | 59 | Mensagens retornadas pelos procedimentos armazenados | 123 |
| Importação de dados para uma tabela existente a partir do nível da tabela | 62 | Mensagens retornadas pelas funções espaciais | 134 |
| Exportando dados para um arquivo de formato | 64 | | |
| Capítulo 6. Criando índices espaciais. | 67 | Capítulo 11. Exibições do catálogo | 143 |
| Utilizando o Centro de Controle para criar um índice espacial | 67 | DB2GSE.COORD_REF_SYS | 143 |
| Determinando os tamanhos de células da grade | 68 | DB2GSE.GEOMETRY_COLUMNS | 144 |
| | | DB2GSE.SPATIAL_GEOCODER | 144 |
| | | DB2GSE.SPATIAL_REF_SYS | 145 |
| Capítulo 7. Recuperando e analisando informações espaciais | 69 | Capítulo 12. Índices espaciais | 147 |
| Métodos de execução de análise espacial | 69 | Fragmento de um programa de amostra | 147 |
| Construindo uma consulta espacial | 69 | Índices B tree | 148 |
| Funções espaciais e SQL | 69 | Formas de se criar um índice espacial | 148 |
| Predicados espaciais e SQL | 71 | Como é gerado um índice espacial | 149 |
| | | Diretrizes sobre o uso de um índice espacial | 153 |
| | | Selecionando o tamanho da célula de grade | 154 |
| | | Selecionando o número de níveis | 154 |
| Capítulo 8. Gravando aplicações para Spatial Extender | 73 | Capítulo 13. Figuras geométricas e funções espaciais associadas | 157 |
| Utilizando o programa de amostra | 73 | Sobre figuras geométricas | 157 |
| As etapas do programa de amostra | 73 | Propriedades e funções associadas | 159 |
| | | Classe. | 160 |
| | | Coordenadas e medidas. | 160 |
| | | Coordenadas X e Y | 160 |
| | | Coordenadas Z e medidas | 161 |
| | | A função ST_CoordDim. | 162 |
| | | Interior, limite e exterior | 162 |
| | | Simples ou não-simples. | 162 |
| | | Vazia ou não vazia | 163 |
| | | Envelope. | 163 |
| | | Dimensão | 163 |
| | | Identificador do sistema de referência espacial | 164 |
| | | figuras geométricas instanciáveis e funções associadas | 164 |
| | | Pontos | 165 |
| | | Cadeias de linhas | 166 |
| | | Polígonos | 167 |
| | | MultiPontos. | 169 |
| | | Cadeias de linhas múltiplas | 169 |
| | | Multipolígonos. | 171 |
| | | Funções que mostram relações e comparações, geram figuras geométricas, e convertem formatos de valores | 172 |
| | | Funções que mostram relações ou comparações entre recursos geográficos | 172 |
| Parte 2. Material de referência | 81 | | |
| Capítulo 9. Procedimentos armazenados | 83 | | |
| db2gse.gse_disable_autogc | 86 | | |
| db2gse.gse_disable_db | 89 | | |
| db2gse.gse_disable_sref | 90 | | |
| db2gse.gse_enable_autogc | 91 | | |
| db2gse.gse_enable_db | 94 | | |
| db2gse.gse_enable_idx | 95 | | |
| db2gse.gse_enable_sref | 97 | | |
| db2gse.gse_export_shape | 100 | | |
| db2gse.gse_import_sde | 102 | | |
| db2gse.gse_import_shape | 104 | | |
| db2gse.gse_register_gc | 107 | | |
| db2gse.gse_register_layer | 109 | | |
| db2gse.gse_run_gc | 116 | | |
| db2gse.gse_unregist_gc | 118 | | |
| db2gse.gse_unregist_layer | 119 | | |
| Capítulo 10. Mensagens | 123 | | |
| Mensagens retornadas pelo Centro de Controle | 123 | | |

| | |
|--|-----|
| Funções que geram novas figuras geométricas a partir de existentes | 184 |
| Funções que convertem o formato dos valores de uma figura geométrica | 190 |

Capítulo 14. Funções espaciais para consultas SQL 195

| | |
|--|-----|
| Aninhamento | 196 |
| Conversão de ST_Geometry em um subtipo | 196 |
| Conversão de uma coleção em uma figura geométrica básica | 197 |
| AsShape | 198 |
| EnvelopesIntersect | 199 |
| GeometryFromShape. | 201 |
| Is3d | 203 |
| IsMeasured | 204 |
| LineFromShape | 205 |
| LocateAlong | 207 |
| LocateBetween. | 209 |
| M | 211 |
| MLine FromShape | 212 |
| MPointFromShape | 214 |
| MPolyFromShape. | 215 |
| PointFromShape | 216 |
| PolyFromShape | 218 |
| ShapeToSQL | 220 |
| ST_Area | 222 |
| ST_AsBinary | 224 |
| ST_AsText | 225 |
| ST_Boundary | 226 |
| ST_Buffer | 228 |
| ST_Centroid | 230 |
| ST_Contains | 231 |
| ST_ConvexHull | 233 |
| ST_CoordDim | 235 |
| ST_Crosses | 237 |
| ST_Difference | 239 |
| ST_Dimension | 240 |
| ST_Disjoint | 242 |
| ST_Distance. | 244 |
| ST_Endpoint | 245 |
| ST_Envelope | 246 |
| ST_Equals | 249 |
| ST_ExteriorRing | 250 |
| ST_GeometryN | 252 |
| ST_GeometryType | 253 |
| ST_GeomFromText | 255 |
| ST_GeomFromWKB | 257 |
| ST_InteriorRingN. | 259 |

| | |
|-----------------------------|-----|
| ST_Intersection. | 264 |
| ST_Intersects | 266 |
| ST_IsClosed. | 267 |
| ST_IsEmpty. | 269 |
| ST_IsRing | 271 |
| ST_IsSimple. | 273 |
| ST_IsValid | 274 |
| ST_Length | 276 |
| ST_LineFromText | 278 |
| ST_LineFromWKB | 279 |
| ST_MLineFromText | 281 |
| ST_MLineFromWKB | 282 |
| ST_MPointFromText | 284 |
| ST_MPointFromWKB | 285 |
| ST_MPolyFromText | 286 |
| ST_MPolyFromWKB | 287 |
| ST_NumGeometries | 288 |
| ST_NumInteriorRing. | 289 |
| ST_NumPoints. | 290 |
| ST_OrderingEquals | 291 |
| ST_Overlaps | 292 |
| ST_Perimeter | 294 |
| ST_Point | 295 |
| ST_PointFromText. | 296 |
| ST_PointFromWKB | 297 |
| ST_PointN | 299 |
| ST_PointOnSurface | 300 |
| ST_PolyFromText | 301 |
| ST_PolyFromWKB | 302 |
| ST_Polygon | 304 |
| ST_Relate | 305 |
| ST_SRID | 307 |
| ST_StartPoint | 308 |
| ST_SymmetricDiff. | 309 |
| ST_Touches | 311 |
| ST_Transform | 312 |
| ST_Union | 314 |
| ST_Within | 315 |
| ST_WKBToSQL | 316 |
| ST_WKToSQL | 318 |
| ST_X | 319 |
| ST_Y | 320 |
| Z | 321 |

Capítulo 15. Sistemas de coordenadas 323

| | |
|---|-----|
| Visão geral dos sistemas de coordenadas | 323 |
| Unidades lineares suportadas | 325 |
| Unidades angulares suportadas | 326 |
| Esferóides suportados | 326 |
| Dados geodéticos suportados | 328 |

| | |
|--|-----|
| Meridianos principais suportados | 330 |
| Projeções do mapa suportadas | 330 |
| Projeções cônicas suportadas | 331 |
| Projeções azimutais ou do circuito impresso suportadas | 331 |
| Parâmetros de projeção do mapa suportados | 332 |

Capítulo 16. Formatos de arquivos para dados espaciais 335

| | |
|--|-----|
| As representações de texto reconhecidas de OGC | 335 |
| As representações (WKB) do modo binário reconhecidas | 340 |
| Definições do tipo numérico | 341 |
| XDR (Big Endian) codificação de tipos numéricos | 342 |
| NDR (Little Endian) codificação de tipos numéricos | 342 |
| Conversão entre NDR e XDR | 342 |

| | |
|---|-----|
| Descrição dos fluxos de bytes | |
| WKBGeometry. | 342 |
| Assertivas para a representação WKB | 344 |
| As representações de formato ESRI | 344 |
| Tipos de formatos no espaço XY | 346 |
| Tipos de formatos medidos no espaço XY | 350 |
| Tipos de formatos no espaço XYZ | 354 |

Parte 3. Apêndices 359

| | |
|-------------------------|------------|
| Avisos | 361 |
| Marcas | 364 |

Índice Remissivo 367

| | |
|---|------------|
| Entre em Contate com a IBM | 373 |
| Informações Sobre o Produto | 373 |

Figuras

1. Linha da tabela que representa um recurso geográfico; linha da tabela cujos dados do endereço representam um recurso geográfico 5
2. Tabelas com colunas espaciais incluídas 5
3. Tabelas que contêm dados espaciais derivados de dados fonte 7
4. Tabela que contém novos dados espaciais derivados de dados espaciais existentes . 8
5. Configuração do cliente-servidor 18
6. Hierarquia dos tipos de dados espaciais 46
7. Aplicação de um nível de grade 10.0e0 150
8. Efeito do acréscimo dos níveis de grade 30.0e0 e 60.0e0 152
9. Hierarquia das figuras geométricas suportadas pelo Spatial Extender . . . 158
10. Objetos da cadeia de linha 167
11. Polígonos. 168
12. Cadeias de linhas múltiplas 170
13. Multipolígonos 171
14. ST_Equals 174
15. ST_Disjoint 176
16. ST_Touches 178
17. ST_Overlaps. 179
18. ST_Within 182
19. ST_Contains. 183
20. Distância mínima entre duas cidades 184
21. ST_Intersection 185
22. ST_Difference 186
23. ST_Union. 186
24. ST_Buffer. 187
25. LocateAlong. 188
26. LocateBetween 189
27. ST_ConvexHull. 189
28. Utilizando a área para encontrar a base da construção 223
29. Um buffer com um raio de 8 km é aplicado a um ponto 229
30. Utilizando ST_Contains para garantir que todas as construções fiquem dentro de seus lotes. 232
31. Utilizando ST_Crosses para encontrar cursos de água que passam por área de lixo tóxico 238
32. Usando ST_Disjoint para encontrar as construções que não estejam dentro de (interseção) alguma área de lixo tóxico . 243
33. Utilizando ST_ExteriorRing para determinar o comprimento do contorno da ilha 251
34. Usando ST_InteriorRingN para determinar o comprimento das margens dos lagos dentro de cada ilha . 259
35. Utilizando ST_Intersection para determinar o quanto uma área em cada construção poderá ser afetada pelo lixo tóxico 265
36. Utilizando ST_Length para determinar o comprimento total dos cursos de água num município 277
37. Usando ST_Overlaps para determinar as construções que pelo menos parcialmente estão dentro de uma área de lixo tóxico. 293
38. Utilizando o ST_SymmetricDiff para determinar as áreas de lixo tóxico que não contêm áreas sensíveis (construções desabitadas). 310
39. Representação no formato NDR 344
40. Um polígono com um orifício e oito vértices 349
41. Conteúdo do fluxo de bytes do polígono 349

Tabelas

| | | | | | |
|-----|---|----|-----|---|-----|
| 1. | Requisitos de espaço em disco | 19 | 18. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_export_shape. | 100 |
| 2. | Informações sobre o CD-ROM de dados e mapas | 30 | 19. | Parâmetros de saída para o procedimento armazenado db2gse.gse_export_shape. | 101 |
| 3. | Operações e funções espaciais | 69 | 20. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_import_sde. | 103 |
| 4. | Regras para exploração de índice | 72 | 21. | Parâmetros de saída para o procedimento armazenado db2gse.gse_import_sde. | 103 |
| 5. | Programa de amostra do Spatial Extender | 74 | 22. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_import_shape. | 105 |
| 6. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_disable_autogc. | 87 | 23. | Parâmetros de saída para o procedimento armazenado db2gse.gse_import_shape. | 106 |
| 7. | Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_autogc. | 88 | 24. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_gc. | 107 |
| 8. | Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_db. | 89 | 25. | Parâmetros de saída para o procedimento armazenado db2gse.gse_register_gc. | 108 |
| 9. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_disable_sref. | 90 | 26. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_layer. | 110 |
| 10. | Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_sref. | 90 | 27. | Parâmetros de saída para o procedimento armazenado db2gse.gse_register_layer. | 115 |
| 11. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_enable_autogc. | 92 | 28. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_run_gc. | 116 |
| 12. | Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_autogc. | 93 | 29. | Parâmetros de saída para o procedimento armazenado db2gse.gse_run_gc. | 117 |
| 13. | Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_db. | 94 | 30. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_unregist_gc. | 118 |
| 14. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_enable_idx. | 95 | 31. | Parâmetros de saída para o procedimento armazenado db2gse.gse_unregist_gc. | 118 |
| 15. | Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_idx. | 96 | 32. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_unregist_layer. | 120 |
| 16. | Parâmetros de entrada para o procedimento armazenado db2gse.gse_enable_sref. | 97 | | | |
| 17. | Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_sref. | 99 | | | |

| | | | | | |
|-----|---|-----|-----|---|-----|
| 33. | Parâmetros de saída para o procedimento armazenado db2gse.gse_unregist_layer. | 120 | 55. | Matriz para ST_Within | 182 |
| 34. | Valores SQLSTATE e SQLCODE de mensagens retornadas pelas funções espaciais | 140 | 56. | Matriz para ST_Contains | 183 |
| 35. | Colunas na exibição de catálogo DB2GSE.COORD_REF_SYS. | 143 | 57. | Matriz de padrão igual | 305 |
| 36. | Colunas na exibição de catálogo DB2GSE.GEOMETRY_COLUMNS | 144 | 58. | Unidades lineares suportadas | 325 |
| 37. | Colunas na exibição de catálogo DB2GSE.SPATIAL_GEOCODER | 145 | 59. | Unidades angulares suportadas | 326 |
| 38. | Colunas na exibição do catálogo DB2GSE.SPATIAL_REF_SYS | 145 | 60. | Esferóides suportados | 326 |
| 39. | As entradas da célula de grade 10.0e0 para os exemplos de figuras geométricas | 150 | 61. | Dados geodéticos suportados | 328 |
| 40. | As interseções das figuras geométricas no índice de três fileiras | 152 | 62. | Meridianos principais suportados | 330 |
| 41. | Matriz para ST_Within | 173 | 63. | Projeções do mapa suportadas | 330 |
| 42. | Matriz para igualdade | 175 | 64. | Projeções cônicas suportadas | 331 |
| 43. | Matriz para ST_Disjoint | 176 | 65. | Parâmetros de projeção do mapa suportados | 332 |
| 44. | Matriz para ST_Intersects (1) | 177 | 66. | Tipos de figura geométrica e suas representações de texto | 338 |
| 45. | Matriz para ST_Intersects (2) | 177 | 67. | Conteúdo do fluxo de bytes do ponto | 346 |
| 46. | Matriz para ST_Intersects (3) | 177 | 68. | Conteúdo do fluxo de bytes MultiPonto | 346 |
| 47. | Matriz para ST_Intersects (4) | 177 | 69. | Conteúdo do fluxo de bytes Polilinha | 347 |
| 48. | Matriz para ST_Touches (1). | 178 | 70. | Conteúdo do fluxo de dados do polígono | 349 |
| 49. | Matriz para ST_Touches (2). | 178 | 71. | Conteúdo do fluxo de bytes do pontoM | 350 |
| 50. | Matriz para ST_Touches (3). | 179 | 72. | Conteúdo do fluxo de bytes do MultipontoM | 351 |
| 51. | Matriz para ST_Overlaps (1) | 179 | 73. | Conteúdo do fluxo de bytes da PolilinhaM | 352 |
| 52. | Matriz para ST_Overlaps (2) | 180 | 74. | Conteúdo do fluxo de bytes do PolígonoM | 353 |
| 53. | Matriz para ST_Crosses (1) | 181 | 75. | Conteúdo do fluxo de bytes do PontoZ | 354 |
| 54. | Matriz para ST_Crosses (2) | 181 | 76. | Conteúdo do fluxo de bytes do MultipontoZ. | 354 |
| | | | 77. | Conteúdo do fluxo de bytes da PolilinhaZ | 356 |
| | | | 78. | Conteúdo do fluxo de bytes do PolígonoZ | 358 |

Sobre este manual

Este manual está dividido em duas partes. A primeira parte contém informações conceituais sobre o DB2 Spatial Extender e explica como instalar, configurar, administrar e programar o Spatial Extender nos sistemas Windows NT e AIX. A segunda parte consiste em informações de referência sobre procedimentos armazenados, figuras geométricas, funções, mensagens e exibições de catálogos que você utiliza com o Spatial Extender.

As alterações técnicas no texto são indicadas por uma linha vertical à esquerda da alteração.

Quem deveria ler este manual

Este manual é para administradores que configuram o ambiente espacial e para programadores de aplicações que desenvolvem aplicações com os dados espaciais.

Convenções

Este manual utiliza estas convenções para destaque:

Tipo negrito

Indica comandos e controles da interface gráfica com o usuário (GUI) (por exemplo, nomes de campos, nomes de pastas, opções de menu).

Tipo monoespaço

Indica exemplos de codificação ou de texto digitado por você.

Tipo itálico

Indica variáveis que devem ser substituídas por um valor. O tipo itálico também indica títulos de manuais e enfatiza palavras.

TIPO MAIÚSCULA

Indica palavras-chave do SQL e nomes de objetos (como, tabelas, exibições e servidores).

Como enviar seus comentários

Seu retorno auxilia a IBM a fornecer informações de qualidade. Por favor envie quaisquer comentários que tenha sobre este manual ou qualquer outra documentação DB2. Você pode utilizar qualquer dos seguintes métodos para fornecer comentários:

- Envie seus comentários a partir da Web. Você pode acessar o comentário de leitores online do IBM Data Management em <http://www.ibm.com/software/data/rcf>
- Envie seus comentários por e-mail para comments@vnet.ibm.com.
Assegure-se de que tenha incluído o nome do produto, o número da versão do mesmo e o nome e número da peça (part number), do manual (se aplicável). Caso esteja comentando sobre um texto específico, por favor inclua a localização do texto (por exemplo, um capítulo e um título de seção, um número de tabela, um número de página ou um título de tópico de auxílio).

Parte 1. Utilizando o Spatial Extender

Capítulo 1. Sobre o Spatial Extender

Este capítulo apresenta o Spatial Extender explicando sua finalidade, comentando sobre os dados que ele processa e ilustrando como utilizá-lo. O capítulo termina com um manual rápido do resto do manual.

A finalidade do Spatial Extender

O Spatial Extender é utilizado na criação de um *sistema de informações geográficas* (GIS - geographic information system): um complexo de objetos, dados e aplicações que permitem a você gerar e analisar informações espaciais sobre recursos geográficos. Os *recursos geográficos* são os objetos que compõem a superfície da terra e os objetos que a ocupam. Eles compõem o ambiente natural (exemplos são os rios, florestas, montanhas e desertos) e o ambiente cultural (cidades, residências, prédios de escritórios, terrenos e assim por diante).

As *informações espaciais* contêm fatos como:

- A localização de recursos geográficos em relação ao seu meio (por exemplo, pontos dentro de uma cidade onde hospitais e clínicas estão localizados, ou a proximidade das residências da cidade em relação a zonas de terremoto)
- Modos como os recursos geográficos estão relacionados entre si (por exemplo, informações de que um determinado sistema fluvial está contido dentro de um região específica, ou de que determinadas pontes naquela região atravessam os braços do sistema fluvial)
- Medidas que se aplicam a um ou mais recursos geográficos (por exemplo, a distância entre um prédio de escritórios e sua divisão de terreno ou o comprimento de um perímetro de preservação de um pássaro)

As informações espaciais, isoladas ou em conjunto com saídas tradicionais do sistema de gerenciamento de banco de dados relacional (RDBMS), podem ajudá-lo a realizar projetos e fazer negócios e tomar decisões políticas. Suponha, por exemplo, que o gerente de um distrito municipal precise verificar quais requerentes e beneficiários realmente moram dentro da área atendida pelo distrito. O Spatial Extender pode tirar estas informações a partir da localização da área atendida e dos endereços dos requerentes e beneficiários.

Ou suponha que o proprietário de uma cadeia de restaurantes queira fazer negócio em cidades próximas. Para determinar onde abrir novos restaurantes, o proprietário precisa responder a perguntas como: Em que locais destas cidades está concentrada a clientela que geralmente frequenta meus

restaurantes? Onde ficam as principais estradas? Onde é mais baixa a taxa de crimes? Onde se encontram os restaurantes da concorrência? O Spatial Extender pode produzir informações espaciais em exibições visuais para responder estas dúvidas e o RDBMS subjacente pode gerar rótulos e textos para explicar estas exibições.

Outros exemplos do uso do Spatial Extender aparecem neste manual, especialmente no “Capítulo 7. Recuperando e analisando informações espaciais” na página 69, “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73 e “Capítulo 14. Funções espaciais para consultas SQL” na página 195.

Dados que representam recursos geográficos

Esta seção apresenta uma visão geral dos dados que são gerados, armazenados e manipulados para obtenção de informações espaciais. Os tópicos abrangidos são:

- Como os dados representam os recursos geográficos
- A natureza dos dados espaciais
- Formas de produção de dados espaciais

Como os dados representam os recursos geográficos

No Spatial Extender, um recurso geográfico pode ser representado por uma linha em uma tabela ou exibição ou por uma parte desta linha. Considere, por exemplo, dois dos recursos geográficos mencionados no “A finalidade do Spatial Extender” na página 3, prédios de escritórios e residências. Na Figura 1 na página 5, cada linha da tabela BRANCHES representa uma filial de um banco. Como uma variação, cada linha da tabela CUSTOMERS na Figura 1 na página 5, tomada por inteiro, representa um cliente do banco. No entanto, parte de cada linha — especificamente, as células que contêm o endereço de um cliente — podem ser consideradas como representando a residência do cliente.

BRANCHES

| ID | NAME | ADDRESS | CITY | STATE | ZIP |
|-----|-----------------|--------------------|----------|-------|-------|
| 937 | Airzone-Multern | 92467 Airzone Blvd | San Jose | CA | 95141 |

CUSTOMERS

| ID | LAST NAME | FIRST NAME | ADDRESS | CITY | STATE | ZIP | CHECKING | SAVINGS |
|---------|-----------|------------|-------------------|----------|-------|-------|----------|---------|
| 59-6396 | Kriner | Endela | 9 Concourt Circle | San Jose | CA | 95141 | A | A |

Figura 1. Linha da tabela que representa um recurso geográfico; linha da tabela cujos dados do endereço representam um recurso geográfico. A linha de dados na tabela BRANCHES representa uma filial de um banco. As células para dados de endereço na tabela CUSTOMERS representam a residência de um cliente. Os nomes e endereços em ambas as tabelas são fictícios.

As tabelas na Figura 1 contêm dados que identificam e descrevem as filiais e clientes do banco. Tais dados são chamados de *dados do atributo*.

Um subconjunto dos dados do atributo — os valores que denotam os endereços das filiais e dos clientes — podem ser convertidos em valores que resultam em informações espaciais. Por exemplo, como mostrado na Figura 1, o endereço de uma filial é 92467 Airzone Blvd., San Jose CA 95141. O endereço de um cliente é 9 Concourt Circle, San Jose CA 95141. O Spatial Extender pode converter estes endereços em valores que indicam onde estão situadas a filial e a casa do cliente em relação ao seu meio. A Figura 2 mostra as tabelas BRANCHES e CUSTOMERS com novas colunas que são projetadas para conter tais valores.

BRANCHES

| ID | NAME | ADDRESS | CITY | STATE | ZIP | LOCATION |
|-----|-----------------|--------------------|----------|-------|-------|----------|
| 937 | Airzone-Multern | 92467 Airzone Blvd | San Jose | CA | 95141 | |

CUSTOMERS

| ID | LAST NAME | FIRST NAME | ADDRESS | CITY | STATE | ZIP | LOCATION | CHECKING | SAVINGS |
|---------|-----------|------------|-------------------|----------|-------|-------|----------|----------|---------|
| 59-6396 | Kriner | Endela | 9 Concourt Circle | San Jose | CA | 95141 | | A | A |

Figura 2. Tabelas com colunas espaciais incluídas. Em cada tabela, a coluna LOCATION irá conter as coordenadas que correspondem aos endereços.

Quando endereços e identificadores similares são utilizados como ponto de partida para informações espaciais, eles são chamados de *dados fonte*. Como os valores derivados deles resultam em informações espaciais, estes valores

derivados são chamados de *dados espaciais*. A seção a seguir descreve os dados espaciais e introduz seus tipos de dados associados.

A natureza dos dados espaciais

Muitos dados espaciais são compostos por coordenadas. Uma *coordenada* é um número que denota uma posição relativa a um ponto de referência. As latitudes, por exemplo, são coordenadas que denotam posições relativas ao equador. As longitudes são coordenadas que denotam posições relativas ao meridiano de Greenwich. Assim, a posição do Yellowstone National Park está definida por sua latitude (44,45 graus ao norte do equador) e sua longitude (110,40 graus oeste do meridiano de Greenwich).

As latitudes, longitudes, seus pontos de referência e outros parâmetros associados são chamados coletivamente de um *sistema de coordenadas*. Também existem sistemas de coordenadas baseados em valores que não sejam a latitude e a longitude. Estes sistemas de coordenadas possuem suas próprias medidas de posição, pontos de referência e parâmetros de distinção adicionais.

O mais simples item de dados espaciais consiste em duas coordenadas que definem a posição de um único recurso geográfico. (Um *item de dados* é o valor ou valores que ocupam uma célula de uma tabela relacional.) Um item de dados espaciais mais amplo consiste em várias coordenadas que definem um caminho linear como uma rua ou rio pode formar. Um terceiro tipo consiste em coordenadas que definem o perímetro de uma área; por exemplo, a margem de um pedaço de terra ou planície aluvial. Estes e outros tipos de itens de dados espaciais que o Spatial Extender suporta estão descritos mais completamente no “Capítulo 13. Figuras geométricas e funções espaciais associadas” na página 157.

Cada item de dados espaciais é uma instância de um tipo de dados espacial. O tipo de dados para duas coordenadas que marcam a localização é `ST_Point`; o tipo de dados para coordenadas que definem caminhos lineares é `ST_LineString`; e o tipo de dados para coordenadas que definem perímetros é `ST_Polygon`. Estes tipos, juntamente com os outros tipos de dados para dados espaciais, são tipos estruturados que pertencem a uma única hierarquia. Para obter uma visão geral da hierarquia, consulte “Sobre tipos de dados espaciais” na página 45.

De onde vêm os dados espaciais

Os dados espaciais podem ser:

- Derivados de dados do atributo
- Derivados de outros dados espaciais
- Importados

Utilizando dados do atributo como dados fonte

O Spatial Extender pode obter dados espaciais a partir de dados do atributo, como endereços (conforme mencionado em “Como os dados representam os recursos geográficos” na página 4). Esse processo é chamado *geocoding*. Para ver a seqüência envolvida, considere a Figura 2 na página 5 como uma imagem “antes” e a Figura 3 como uma imagem “depois”. A Figura 2 na página 5 mostra que a tabela BRANCHES e a tabela CUSTOMERS possuem uma coluna que contém apenas valores NULOS designados aos dados espaciais. Suponha que o Spatial Extender efetue o geocode nos endereços nestas tabelas para obter coordenadas que correspondem aos endereços e coloca as coordenadas em colunas. A Figura 3 ilustra este resultado.

BRANCHES

| ID | NAME | ADDRESS | CITY | STATE | ZIP | LOCATION |
|-----|-----------------|--------------------|----------|-------|-------|-----------|
| 937 | Airzone-Multern | 92467 Airzone Blvd | San Jose | CA | 95141 | 1653 3094 |

CUSTOMERS

| ID | LAST NAME | FIRST NAME | ADDRESS | CITY | STATE | ZIP | LOCATION | CHECKING | SAVINGS |
|---------|-----------|------------|-------------------|----------|-------|-------|----------|----------|---------|
| 59-6396 | Kriner | Endela | 9 Concourt Circle | San Jose | CA | 95141 | 953 1527 | A | A |

Figura 3. Tabelas que contêm dados espaciais derivados de dados fonte. A coluna LOCATION na tabela CUSTOMERS contém coordenadas que um geocoder obteve a partir dos endereços nas colunas ADDRESS, CITY, STATE e ZIP. De forma semelhante, a coluna LOCATION na tabela BRANCHES contém coordenadas que o geocoder obteve a partir do endereço nas colunas ADDRESS, CITY, STATE e ZIP desta tabela. Este exemplo é fictício; coordenadas simuladas, e não verdadeiras, são apresentadas.

O Spatial Extender utiliza uma função, chamada de *geocoder*, para converter dados do atributo em dados espaciais e para colocar estes dados espaciais em colunas da tabela. Para obter maiores informações sobre os geocoders, consulte “Sobre geocodificação” na página 53.

Utilizando outros dados espaciais como dados fonte

Os dados espaciais podem ser gerados não apenas a partir de dados do atributo, mas também de outros dados espaciais. Suponha, por exemplo, que o banco, cujas filiais estão definidas na tabela BRANCHES, deseja saber quantos clientes estão localizados dentro de cinco milhas de cada filial. Para que o banco possa obter estas informações do banco de dados, ele deve fornecer ao banco de dados a definição da zona que se encontra em um raio de cinco milhas ao redor de cada filial. Uma função do Spatial Extender, *ST_Buffer*, pode criar esta definição. Utilizando as coordenadas de cada filial como entrada, o *ST_Buffer* pode gerar as coordenadas que demarcam os

perímetros das zonas desejadas. A Figura 4 mostra a tabela BRANCHES com informações fornecidas pelo ST_Buffer.

BRANCHES

| ID | NAME | ADDRESS | CITY | STATE | ZIP | LOCATION | SALES_AREA |
|-----|-----------------|--------------------|----------|-------|-------|-----------|---|
| 937 | Airzone-Multern | 92467 Airzone Blvd | San Jose | CA | 95141 | 1653 3094 | 1002 2001, 1192 3564, 2502 3415, 1915 3394, 1002 2001 |

Figura 4. Tabela que contém novos dados espaciais derivados de dados espaciais existentes. As coordenadas na coluna SALES_AREA foram obtidas pela função ST_Buffer a partir das coordenadas na coluna LOCATION. Assim como as coordenadas na coluna LOCATION, as na coluna SALES_AREA são simuladas; elas não são verdadeiras.

Além do ST_Buffer, o Spatial Extender fornece várias outras funções que obtêm novos dados espaciais de dados espaciais existentes. Para obter descrições do ST_Buffer e destas outras funções, consulte “Funções que geram novas figuras geométricas a partir de existentes” na página 184.

Importando dados espaciais

Um terceiro modo de se obter dados espaciais é importando-os dos arquivos que estejam em um dos formatos suportados pelo Spatial Extender. Para obter descrições destes formatos, consulte “Capítulo 16. Formatos de arquivos para dados espaciais” na página 335. Estes arquivos contêm dados que geralmente são empregados em mapas: roteiros do censo, planícies aluviais, deslocamentos de terremotos e outros. Ao utilizar tais dados em conjunto com dados espaciais produzidos por você, você pode aumentar as informações geográficas disponíveis. Se, por exemplo, uma departamento de trabalho público precisa determinar a quais riscos uma comunidade residencial está vulnerável, ele pode usar o ST_Buffer para definir uma zona ao redor da comunidade. Depois, o departamento poderia importar dados sobre planícies aluviais e de deslocamentos de terremotos para saber quais destas áreas problemáticas abrangem a zona.

Como criar e usar um GIS do Spatial Extender

Um GIS do Spatial Extender é criado através da configuração do Spatial Extender e do desenvolvimento de projetos do GIS dentro de ambientes combinados do Spatial Extender e seu DB2 RDBMS subjacente. Você utiliza o GIS através da implementação destes projetos; isto é, gerando e analisando as informações — espaciais e tradicionais — que eles foram projetados para fornecer. O trabalho inteiro envolve a realização de vários conjuntos de tarefas. Esta seção apresenta as interfaces com as quais você pode executar estas tarefas, fornece uma visão geral das tarefas e apresenta um cenário para ilustrá-las.

Interfaces para o Spatial Extender e funcionalidade associada

Esta seção inspeciona as interfaces mediante as quais você pode criar um GIS do Spatial Extender (isto é, configurar recursos para ele, obter dados espaciais e assim por diante) e utilizá-lo (isto é, gerar e analisar informações sobre recursos geográficos).

Você pode criar um GIS do Spatial Extender:

- Utilizando as janelas e opções de menu do Spatial Extender do Centro de Controle do DB2. Para obter instruções, consulte:
 - “Capítulo 3. Configurando recursos” na página 33
 - “Capítulo 4. Definindo colunas espaciais, registrando-as como camadas e ativando um geocoder para mantê-las” na página 45
 - “Capítulo 5. Preenchendo colunas espaciais” na página 53
 - “Capítulo 6. Criando índices espaciais” na página 67
- Executando um programa de aplicação que chame procedimentos armazenados do Spatial Extender. Para obter diretrizes sobre o desenvolvimento de um programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.
- Utilizando o Centro de Controle e um programa de aplicação. Você pode usar, por exemplo, o Centro de Controle para chamar o geocoder padrão. Se, além disso, você quiser usar outro geocoder, você primeiro deve registrá-lo para o Spatial Extender solicitando o procedimento armazenado `db2gse.gse_register_gc` em um programa de aplicação. (Para obter informações sobre geocoders não-padrões, consulte “Sobre geocodificação” na página 53. Para obter informações sobre o procedimento armazenado `db2gse.gse_register_gc`, consulte “`db2gse.gse_register_gc`” na página 107.)
- Utilizando o Centro de Controle, um programa de aplicação, ou ambos, em conjunto com outras interfaces. Para criar, por exemplo, uma tabela para guardar dados que devem ser gerados por uma função espacial, como um geocoder, você pode usar o Processador de Linha de Comandos ou as interfaces do Centro de Controle.

Você pode usar um GIS do Spatial Extender:

- Transmitindo informações graficamente com um geobrowser; por exemplo, ArcExplorer Java Version 3.0, que é oferecido pelo Environmental Systems Research Institute (ESRI)
- Submetendo explicitamente consultas SQL a partir do Centro de Controle do DB2 ou do Processador de Linha de Comandos
- Submetendo consultas SQL a partir de um programa de aplicação

Tarefas que você executa para criar e usar um GIS do Spatial Extender

Esta seção fornece uma visão geral das tarefas através das quais você cria e utiliza um GIS do Spatial Extender. As tarefas através das quais você cria o GIS envolvem configurar o Spatial Extender e desenvolver projetos do GIS. As

tarefas através das quais você utiliza o GIS envolvem implementar os projetos. Esta visão geral começa com a configuração do Spatial Extender e depois passa para o desenvolvimento e implementação de um projeto do GIS. A seção termina indicando como as tarefas descritas na visão geral podem variar na prática.

Configurando o Spatial Extender

Para configurar o Spatial Extender:

1. Faça planos e preparações (decida quais projetos do GIS serão desenvolvidos, decida o banco de dados a ser ativado para o Spatial Extender, selecione pessoal para administrar o Spatial Extender e desenvolva projetos, e outras coisas).
2. Instale o Spatial Extender.
3. Coloque recursos no lugar para suportar projetos do GIS; por exemplo:

Recursos fornecidos pelo Spatial Extender

Incluem um catálogo do sistema, tipos de dados espaciais, funções espaciais (incluindo um geocoder padrão) e outros. A tarefa de configuração destes recursos é chamada de *ativação do banco de dados para operações espaciais*.

Geocoders desenvolvidos por usuários, fornecedores ou ambos.

O geocoder padrão converte endereços dos Estados Unidos em dados espaciais. Sua organização e outras podem fornecer geocoders que convertam endereços estrangeiros e outros tipos de dados do atributo em dados espaciais.

Para obter instruções sobre a instalação do Spatial Extender, consulte “Capítulo 2. Instalando o Spatial Extender” na página 17. Para obter instruções sobre o uso do Centro de Controle para colocar os recursos no lugar, consulte “Capítulo 3. Configurando recursos” na página 33. Para obter diretrizes sobre o uso de um programa da aplicação para este fim, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73. Para obter um cenário que ilustre o trabalho geral de configuração do Spatial Extender, consulte “Um sistema para integrar dados espaciais e tradicionais” na página 13.

Desenvolvendo e implementando um projeto do GIS

Para desenvolver e implementar um projeto do GIS:

1. Faça planos e preparações (defina metas para o projeto, decida as tabelas e dados necessários, determine o sistema ou sistemas de coordenadas a serem utilizados e outras coisas).
2. Decida o sistema ou sistemas de referência espacial a ser utilizado. Valores de coordenada geralmente incluem inteiros positivos, números negativos e números decimais. O Spatial Extender, no entanto, deve armazenar todos os valores de coordenada na forma de inteiros positivos. Um *sistema de*

referência espacial é um conjunto de parâmetros que define como números negativos e decimais em um sistema de coordenadas específico devem ser convertidos em inteiros positivos, para que o Spatial Extender possa armazená-los. Depois de decidir o sistema de coordenadas a ser utilizado para uma coluna espacial, é preciso especificar o sistema de referência espacial através do qual a conversão necessária pode ocorrer para aquela coluna. Quando um sistema de referência espacial existente satisfaz seus requisitos, você pode utilizá-lo; do contrário, você pode criar um.

3. Defina uma ou mais colunas para conter os dados espaciais, registre-as para o Spatial Extender e ative um geocoder para mantê-las automaticamente.

O registro de uma coluna espacial envolve seu registro no catálogo do Spatial Extender. A partir do momento em que ela é registrada, ela é chamada de *camada*, pois as informações geradas a partir dela acrescentam um estrato ou camada, na paisagem geográfica virtual que o GIS cria para você. Depois de registrá-la, você pode executar operações espaciais na mesma; por exemplo, você pode preenchê-la e definir um índice espacial na mesma.

4. Preencha as colunas espaciais:
 - Para um projeto que precisa de um geocoder, defina parâmetros para o geocoder. Depois, execute-o de modo que, em uma única operação, ele efetue a geocodificação em todos os dados fonte disponíveis e carregue as coordenadas resultantes para a camada.
 - Para um projeto que requer que os dados espaciais sejam importados, importe-os.
5. Facilite o acesso às colunas espaciais. Isto envolve especificamente a definição de índices que permitem que o DB2 acesse dados espaciais rapidamente e a definição de exibições que permite que os usuários recuperem dados inter-relacionados de modo eficiente. Depois de definir a exibição, é necessário registrar suas colunas espaciais como camadas.
6. Gere e analise informações espaciais e informações de negócios relacionadas. Isto envolve consultar colunas espaciais e colunas de atributo relacionadas. Em tais consultas, você pode incluir funções do Spatial Extender que retornam uma grande variedade de informações; por exemplo, a distância mínima entre dois recursos geográficos ou coordenadas que definem uma área ao redor de um recurso geográfico. Para obter informações sobre a função, *ST_Buffer*, que retorna tais coordenadas, consulte “Utilizando outros dados espaciais como dados fonte” na página 7 e “*ST_Buffer*” na página 228. Para obter exemplos de consultas que utilizam funções espaciais, consulte “Capítulo 7. Recuperando e analisando informações espaciais” na página 69 e “Capítulo 14. Funções espaciais para consultas SQL” na página 195.

Para obter instruções sobre o uso do Centro de Controle para realizar tarefas envolvidas no desenvolvimento de um projeto do GIS, consulte:

- “Capítulo 3. Configurando recursos” na página 33
- “Capítulo 4. Definindo colunas espaciais, registrando-as como camadas e ativando um geocoder para mantê-las” na página 45
- “Capítulo 5. Preenchendo colunas espaciais” na página 53
- “Capítulo 6. Criando índices espaciais” na página 67

Para obter diretrizes sobre o uso do Centro de Controle na implementação de um projeto do GIS, consulte “Capítulo 7. Recuperando e analisando informações espaciais” na página 69.

Para obter diretrizes sobre o uso de um programa da aplicação para desenvolver e implementar um projeto do GIS, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Para ver um cenário ilustrativo do trabalho como um todo, consulte “Uma projeto para estabelecer escritórios e ajustar prêmios” na página 13.

Como os conjuntos de tarefas pode variar

Os conjuntos de tarefas realizadas para criar e utilizar um GIS do Spatial Extender podem variar em conteúdo e seqüência, dependendo dos seus requisitos e das interfaces utilizadas. Considere, por exemplo, as tarefas de definição de colunas para conter dados espaciais, registro das mesmas como camadas e ativação de um geocoder para mantê-las automaticamente. Com o Centro de Controle, estas tarefas podem ser realizadas juntas, a partir de uma única janela. No entanto, se estiver solicitando procedimentos armazenados a partir de um programa, você pode realizar estas tarefas separadamente, e pode regulá-las a seu critério.

Cenário: Uma companhia de seguros atualiza seu GIS

Esta seção apresenta um cenário ilustrativo dos conjuntos de tarefas descritos na seção anterior.

O ambiente de sistemas de informações da Safe Harbor Real Estate Insurance Company contém um sistema DB2 Universal Database e um sistema separado de gerenciamento de banco de dados do GIS. Até certo ponto, as consultas podem recuperar combinações de dados dos dois sistemas. Por exemplo, uma tabela do DB2 armazena informações sobre rendimentos e uma tabela do GIS armazena as localizações das filiais da empresa. Assim, fica possível descobrir as localizações dos escritórios que apresentam rendimentos de quantias especificadas. Mas os dados dos dois sistemas não podem ser integrados (por exemplo, os usuários não podem juntar colunas do DB2 com colunas do GIS) e serviços do DB2 como otimização de consultas não estão disponíveis ao GIS. Para superar estas desvantagens, a Safe Harbor adquire o Spatial Extender e

estabelece um novo departamento de desenvolvimento do GIS. As seções a seguir descrevem como o departamento configura o Spatial Extender e leva adiante seu primeiro projeto.

Um sistema para integrar dados espaciais e tradicionais

Para configurar o Spatial Extender, o departamento de desenvolvimento do GIS da Safe Harbor procede da seguinte maneira:

1. O departamento prepara a inclusão do Spatial Extender em seu ambiente do DB2. Por exemplo:
 - a. A equipe de gerenciamento do departamento aponta uma equipe de administração espacial para instalar e implementar o Spatial Extender e uma equipe de análise espacial para gerar e analisar informações espaciais.
 - b. Como as decisões empresariais da Safe Harbor são conduzidas principalmente pelas exigências dos clientes, a equipe de gerenciamento decide instalar o Spatial Extender no banco de dados que contém informações sobre seus clientes. Grande parte das informações é armazenada em uma tabela chamada CUSTOMERS.
Como uma forma conveniente de se referir ao banco de dados selecionado, os membros do departamento de desenvolvimento do GIS o chamam de *banco de dados do GIS*. No entanto, eles estão cientes de que ele não está reservado apenas para projetos do GIS; aplicações não-espaciais podem continuar utilizando-o, como antes.
2. A equipe de administração espacial instala o Spatial Extender.
3. Esta mesma equipe configura recursos que os projetos do GIS irão exigir:
 - A equipe utiliza o Centro de Controle para fornecer os recursos que ativam o banco de dados do GIS para operações espaciais. Estes recursos incluem o catálogo do Spatial Extender, os tipos de dados espaciais, as funções espaciais e outros.
 - Como a Safe Harbor está começando a expandir seus negócios para o Canadá, a equipe de administração espacial começa a solicitar aos fornecedores canadenses geocoders que convertem os endereços canadenses em dados espaciais.

Uma projeto para estabelecer escritórios e ajustar prêmios

Para levar adiante seu primeiro projeto do GIS sob o Spatial Extender, o departamento de desenvolvimento do GIS procede da seguinte maneira:

1. O departamento prepara o desenvolvimento do projeto; por exemplo:
 - A equipe de gerenciamento define metas para o projeto:
 - Determinar onde estabelecer novas filiais.

- Ajustar prêmios com base na proximidade dos clientes às áreas de risco (áreas com altas taxas de acidentes de tráfego, áreas com altas taxas de criminalidade e zonas de enchentes, terremotos e assim por diante).
 - O projeto do GIS se preocupa com os clientes e escritórios nos Estados Unidos. Assim, a equipe de administração espacial decide:
 - Utilizar sistemas de coordenadas que definem com precisão as localizações dos setores nos Estados Unidos em que a Safe Harbor faz negócios.
 - Utilizar o geocoder padrão, pois ele está preparado para efetuar o geocode de endereços nos Estados Unidos.
 - A equipe de administração espacial decide quais os dados necessários para satisfazer as metas do projeto e em que tabelas estes dados ficarão contidos.
2. Utilizando o Centro de Controle, a equipe cria dois sistemas de referência espacial. Um determina como as coordenadas que definem as localizações dos escritórios devem ser convertidas em itens de dados que o Spatial Extender pode armazenar. A outra determina como as coordenadas que definem as localizações das residências dos clientes devem ser convertidas em itens de dados que o Spatial Extender pode armazenar.
 3. Utilizando o Centro de Controle, a equipe de administração espacial define colunas para conter os dados espaciais, as registra como camadas e ativa um geocoder para mantê-las automaticamente:
 - A equipe acrescenta uma coluna LOCATION na tabela CUSTOMERS. A tabela já contém endereços dos clientes. O geocoder padrão os converte em dados espaciais e carrega estes dados para a coluna LOCATION.
 - A equipe cria uma tabela OFFICES para conter os dados que agora estão armazenados no GIS separado. Estes dados incluem os endereços das filiais da Safe Harbor, os dados espaciais originados destes endereços através de um geocoder, e dados espaciais que definem uma zona dentro de um raio de cinco milhas ao redor de cada escritório. Os dados gerados pelo geocoder vão para a coluna LOCATION. Os dados que definem as zonas vão para uma coluna SALES_AREA.
 - A equipe registra como camadas as duas colunas LOCATION e as colunas SALES_AREA.
 - A equipe ativa o geocoder padrão para manter automaticamente as duas colunas LOCATION.
 4. A equipe de administração espacial preenche a coluna LOCATION da tabela CUSTOMERS, a tabela OFFICES inteira e uma nova tabela HAZARD_ZONES:
 - A equipe utiliza o Centro de Controle para preencher a coluna LOCATION da tabela CUSTOMER:

- a. A equipe instrui o geocoder para que ele insira dados espaciais para um endereço na coluna LOCATION somente sob a seguinte condição: uma correspondência entre o endereço e sua contrapartida nos registros da Repartição de Censo dos Estados Unidos (United States Census Bureau) devem ser de 100 por cento de precisão. (Um arquivo de endereços que são fornecidos pela Repartição de Censo é enviado com o Spatial Extender. Para que o geocoder possa converter um endereço nos dados fonte em dados espaciais, o geocoder deve tentar corresponder este endereço a uma contraparte no arquivo. Os usuários especificam a porcentagem de precisão da correspondência para que os dados espaciais sejam colocados na tabela. Esta porcentagem é chamada de *precisão*.)
 - b. A equipe executa o geocoder no modo batch, para que a realização do geocode de todos os endereços da tabela sejam feita em uma operação. Para desânimo da equipe, o geocoder rejeita aproximadamente um em cada dez endereços!
 - c. A equipe supõe que os rejeitados devem ser endereços novos que não possuem correspondência exata nos registros da Repartição de Censo. Para solucionar o problema, a equipe reduz a precisão para 85.
 - d. A equipe executa novamente o geocoder no modo batch. A taxa de rejeição dos endereços cai para um nível aceitável.
- Utilizando um utilitário que é fornecido pelo GIS separado, a equipe carrega os dados do escritório em um arquivo. Depois, a equipe utiliza o Centro de Controle para importar estes dados do arquivo para uma nova tabela OFFICES.
 - Utilizando o Centro de Controle, a equipe cria uma tabela HAZARD ZONES, registra suas colunas espaciais como camadas e importa os dados para ela. Os dados se originam de um arquivo adquirido de um fornecedor de mapas.
5. Utilizando o Centro de Controle, a equipe de administração espacial facilita o acesso às novas camadas:
 - A equipe cria índices para elas.
 - A equipe cria uma exibição que junta colunas das tabelas CUSTOMERS e HAZARD ZONES. Depois, a equipe registra como camadas as colunas espaciais da exibição.
 6. A equipe de análise espacial executa consultas para obter informações que irão ajudá-la a satisfazer os objetivos originais: determinar onde estabelecer as novas filiais, e ajustar prêmios com base na proximidade dos clientes às áreas de risco.

Capítulo 2. Instalando o Spatial Extender

Este capítulo fornece instruções sobre a instalação do DB2 Spatial Extender para AIX, Windows NT e Windows 2000. Os tópicos a seguir também são abordados:

- Configuração do DB2 Spatial Extender
- Requisitos do Sistema
- Instalação do DB2 Spatial Extender para Windows NT e Windows 2000
- Instalação do DB2 Spatial Extender para AIX
- Verificação da Instalação
- Considerações pós-instalação
- Invocação do Spatial Extender

Configuração do DB2 Spatial Extender

Um sistema Spatial Extender consiste no DB2 Universal Database, Spatial Extender e em um geobrowser (por exemplo, ArcExplorer Java Versão 3.0). Geralmente, um banco de dados habilitado para operações espaciais encontra-se no servidor. Você pode utilizar os aplicativos do cliente para acessar os dados espaciais por meio dos procedimentos armazenados e das consultas espaciais do Spatial Extender. Pode também configurar o DB2 Spatial Extender em um ambiente independente, que é uma configuração em que o cliente e o servidor residem na mesma máquina. Nas duas configurações, cliente-servidor e independente, é possível exibir dados espaciais com um geobrowser.

No momento, a IBM não está distribuindo um geobrowser que possa produzir resultados visuais de consultas. Para obter mais informações sobre os geobrowsers e como adquirir um, consulte “Efetuando o download do ArcExplorer” na página 29.

A Figura 5 na página 18 ilustra a arquitetura do Spatial Extender.

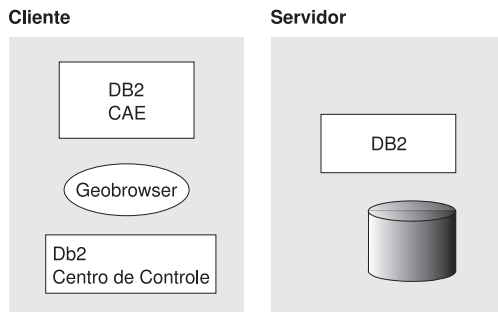


Figura 5. Configuração do cliente-servidor

Exigências do sistema

Esta seção explica os requisitos de software e hardware para o DB2 Spatial Extender.

Sistemas operacionais suportados

O Spatial Extender pode ser instalado no AIX Versão 4.2 (ou posterior), no Windows NT Versão 4.0 (ou posterior) com Service Pack 5 e no Windows 2000.

Nota: Se planeja utilizar o ArcSDE para transmitir e exibir dados espaciais, o AIX Versão 4.3.3 ou posterior será necessário.

Software de banco de dados exigido

Antes de instalar o DB2 Spatial Extender, é necessário ter o seguinte software DB2 instalado e configurado no cliente e no servidor:

Software Cliente

Para produtos do cliente DB2 Spatial Extender:

- DB2 Administration Client, Versão 7.1
Se você não planeja utilizar o Centro de Controle do DB2, um geobrowser para acessar dados espaciais ou o programa de amostra do DB2 Spatial Extender, poderá instalar e utilizar o DB2 Administration Client, Versão 6.0.
- Fixpack 1

O DB2 Administration Client com Fixpack 1 é instalado em seu sistema automaticamente quando você instala o cliente DB2 Spatial Extender a partir do CD-ROM.

Importante: Se o DB2 Universal Database Enterprise Edition Versão 7.1 ou o DB2 Universal Database Enterprise-Extended Edition Versão 7.1

estiver instalado no cliente, será necessário instalar o Fixpack 1 *antes* de instalar o DB2 Spatial Extender.

Software Servidor

Para os produtos do servidor DB2 Spatial Extender, um dos seguintes produtos do servidor deve ser instalado no sistema *antes* da instalação do DB2 Spatial Extender:

- DB2 Universal Database Enterprise Edition Versão 7.1, com Fixpack 1
- DB2 Universal Database Enterprise-Extended Edition Versão 7.1, com Fixpack 1

Se planeja utilizar o Centro de Controle do DB2, crie e configure o DB2 Administration Server (DAS). Para obter mais informações sobre como criar e configurar o DAS, consulte o *IBM DB2 Universal Database Administration Guide: Implementation*.

Nota: Embora você possa utilizar o DB2 Spatial Extender com o DB2 Universal Database Enterprise-Extended Edition, o índice espacial não pode ser particionado em vários nós como no ambiente de processamento paralelo compacto (MPP).

Requisitos de espaço em disco

Tabela 1 lista os requisitos de espaço em disco para o Spatial Extender.

Tabela 1. Requisitos de espaço em disco

| Produtos do Spatial Extender | Espaço em disco |
|---|--|
| Produtos do servidor para DB2 Spatial Extender: <ul style="list-style-type: none">• Código de biblioteca do servidor Spatial Extender, dados de referência do geocoder de amostra e documentação• Opcional e disponível em um CD-ROM separado: dados de referência do geocoder (Estados Unidos) Para obter mais informações sobre como utilizar os dados de referência do geocoder, consulte “Dados de Referência do Geocoder do DB2 Spatial Extender” na página 30. | Espaço total em disco de 594 MB: <ul style="list-style-type: none">• 31 MB (Código de biblioteca do servidor Spatial Extender, dados de referência do geocoder de amostra e documentação)• 563 MB (United States geocoder reference data) |
| Produtos do cliente para o DB2 Spatial Extender (inclui os dados do programa de amostra) | 1 MB |

Tabela 1. Requisitos de espaço em disco (continuação)

| Produtos do Spatial Extender | Espaço em disco |
|------------------------------|--|
| | <p>Nota: O requisito de espaço em disco calculada nesta tabela presume que você instalou o DB2 Universal Database e o DB2 Spatial Extender em uma instalação típica para o Windows NT ou Windows 2000, ou com os componentes pré-selecionados no AIX. Se você estiver instalando o DB2 Spatial Extender ou instalou o DB2 Universal Database com um tipo de instalação diferente, seus cálculos de espaço em disco podem diferenciar.</p> |

Instalação do DB2 Spatial Extender para Windows NT e Windows 2000

Para instalar o Spatial Extender para Windows NT e Windows 2000:

1. Insira o CD-ROM do Spatial Extender na unidade de CD-ROM. O DB2 Launchpad, uma interface da qual você pode instalar o DB2 Spatial Extender, é aberto.
2. Clique em **Instalar**. Após o início do programa de instalação, a janela Selecionar Produtos é aberta.

Observações:

- a. Em qualquer momento durante a instalação, você pode clicar em **Cancelar** para encerrar e sair da instalação.
 - b. Se você receber uma mensagem de aviso informando que o DB2 está atualmente em execução e bloqueado por uma lista de processos, clique em **Sim apenas** se o banco de dados não estiver sendo utilizado e não houver usuários importantes conectados. O DB2 encerrará esses processos e não tentará salvar os dados. Se estiver instalando em um sistema ativo, utilize outros métodos administrativos para encerrar os processos DB2 bloqueados.
 - c. Se estiver instalando o DB2 Spatial Extender em um ambiente independente, selecione **DB2 Administration Client** no programa de configuração da instalação no Windows NT e no Windows 2000. A definição padrão no programa de configuração da instalação *não* tem o **DB2 Administration Client** pré-selecionado.
3. Selecione na lista os produtos que deseja instalar:

- Selecione **DB2 Spatial Extender Server** e **DB2 Administration Client** se estiver instalando o DB2 Spatial Extender em uma configuração independente.
- Selecione **DB2 Spatial Extender Server** se estiver instalando o DB2 Spatial Extender em uma plataforma de servidor.
- Selecione **DB2 Administration Client** se estiver instalando o DB2 Spatial Extender em um ambiente de cliente.

A opção **DB2 Administration Client** instala os sub-componentes do DB2 Universal Database, o programa de amostra e os dados do Spatial Extender e o Centro de Controle do DB2 Universal Database. A opção **DB2 Spatial Extender Server** inclui o código de biblioteca do servidor DB2 Spatial Extender, uma amostra dos dados de referência do geocoder e documentação.

Nota: O produto do servidor DB2 Spatial Extender inclui *apenas* um subconjunto dos dados do geocoder que estão disponíveis. Um conjunto completo dos dados de referência do geocoder dos Estados Unidos é fornecido em um CD-ROM separado, enviado com o DB2 Spatial Extender.

4. Clique em **Avançar**. A janela Selecionar Tipo de Instalação é aberta.
5. Selecione o tipo de instalação. Se você selecionar **Personalizar**, a janela Selecionar Componentes é aberta. Selecione os componentes que deseja instalar. Para isso, é necessário ter conhecimento dos componentes e das definições do DB2.
6. Clique em **Avançar** para abrir a janela Escolher Localização de Destino.
7. Escolha a pasta em que deseja que o DB2 Spatial Extender seja instalado. Para alterar a pasta da definição padrão, clique em **Procurar**.

Nota: Se o DB2 Universal Database já estiver instalado em seu sistema, você não poderá escolher uma nova localização ou criar uma nova pasta a partir desta janela. A janela Escolher Localização de Destino exibe o seguinte:

- A localização da pasta em que o DB2 Spatial Extender será instalado
 - O espaço em disco necessário para instalar o DB2 Spatial Extender
8. Clique em **Avançar** para instalar o DB2 Spatial Extender. A janela Progresso de Instalação é aberta e exibe o progresso do procedimento de instalação.

Instalação do DB2 Spatial Extender para AIX

Esta seção descreve as etapas necessárias para executar a instalação do DB2 Spatial Extender para AIX. Você pode utilizar o instalador do DB2 (interativamente ou não-assistido), o System Management Interface Tool (SMIT) ou o comando **installp**. Os tópicos nesta seção incluem:

- Montagem do CD-ROM
- Utilização do instalador do DB2
- Utilização do SMIT e do comando **installp**
- Estabelecimento do ambiente da instância do DB2 Spatial Extender

Montagem do CD-ROM

Durante as várias etapas na instalação no AIX do DB2 Spatial Extender, é necessário montar o CD-ROM. Sempre que encontrar */cdrom* mencionado neste documento, certifique-se de que você executou o procedimento de montagem localizado nesta seção *antes* de iniciar a próxima etapa. As instruções de montagem a seguir se aplicam às instalações que utilizam o instalador do DB2, o SMIT e o comando **installp**.

Para montar o CD-ROM:

1. Inicie sessão como um usuário com autoridade de root.
2. Insira o CD-ROM na unidade.
3. Crie um diretório para montar o CD-ROM digitando o seguinte comando:

```
mkdir -p /cdrom
```

em que */cdrom* é o diretório de montagem do CD-ROM.

4. Aloque um sistema de arquivos do CD-ROM digitando o seguinte comando:

```
smitty storage
```
5. Selecione **Sistema de Arquivos**.
6. Selecione **Adicionar/Alterar/Mostrar/Excluir Sistema de Arquivos/**.
7. Selecione **Sistema de Arquivos do CD-ROM**.
8. Selecione **Adicionar Sistema de Arquivos do CDROM**.
9. Selecione **Nome do Dispositivo**.

Os nomes dos dispositivos dos sistemas de arquivos do CD-ROM devem ser exclusivos. Se o nome de um sistema já existente duplicar aquele que deseja selecionar, exclua o sistema de arquivos do CD-ROM existente ou utilize outro nome para seu diretório.

10. Na janela popup, digite o seguinte ponto de montagem:

```
/cdrom
```
11. Monte o sistema de arquivos do CD-ROM digitando o seguinte comando:

smit mountfs

12. Digite o nome do sistema de arquivos (por exemplo, /dev/cd0).
13. Digite o nome do diretório, por exemplo cdrom).
14. Digite o tipo do sistema de arquivos, por exemplo, cdrfs).
15. Defina o sistema de montagem READ-ONLY como **Sim**.
16. Encerre a sessão.

Utilização do instalador do DB2

Esta seção descreve os seguintes tópicos:

- Instalação interativa
- Instalação não-assistida
- Geração de um log ou arquivo de rastreamento

Durante a instalação no AIX utilizando o instalador do DB2, você pode optar por instalar interativamente ou em um ambiente não-assistido. Em uma instalação interativa, você estabelece interface com uma série de telas para configurar o DB2 Spatial Extender. Em uma instalação não-assistida, você fornece as informações de configuração em um arquivo de resposta que você cria antes de invocar o instalador do DB2. Se precisar instalar o DB2 Spatial Extender em mais de uma máquina, poderá personalizar o arquivo e utilizá-lo para instalar o produto em várias estações de trabalho.

Para instalar o DB2 Spatial Extender interativamente:

1. Inicie sessão no cliente de destino ou na máquina servidora como usuário root.
2. Insira o CD-ROM na unidade de CD-ROM.
3. Digite `cd /cdrom`.
4. Digite `./db2setup`. O instalador do DB2 é aberto.
5. Selecione os produtos que deseja instalar:
 - Selecione **Spatial Extender Server** e **Spatial Extender Client** se estiver instalando o DB2 Spatial Extender em uma configuração independente.
 - Selecione **DB2 Spatial Extender Server** se estiver instalando o DB2 Spatial Extender em uma plataforma de servidor.
 - Selecione **DB2 Spatial Extender Client** se estiver instalando o DB2 Spatial Extender em um ambiente de cliente.
 - Selecione **DB2 Administration Client** se desejar suporte adicional ao cliente DB2.

Utilize a tecla Tab para navegar para o componente que deseja selecionar e pressione Enter. Uma janela de personalização está disponível para cada produto selecionado.

6. Selecione o idioma nacional dos componentes escolhidos.

7. Utilize a tecla Tab para ir até o botão **OK** e pressione Enter para instalar o DB2 Spatial Extender.

Para instalar o DB2 Spatial Extender não-assistido:

1. Crie um arquivo de resposta para uma instalação não-assistida.
2. Inicie uma instalação não-assistida.

Estas etapas são explicadas com mais detalhes nesta seção.

Etapa 1. Crie um arquivo de resposta para uma instalação não-assistida:

Nota: Você pode pular a Etapa 1 e continuar com a Etapa 2 se aceitar os valores padrão no arquivo de resposta de amostra.

1. Abra o arquivo de resposta de amostra para os produtos que deseja instalar. Os arquivos de resposta de amostra estão localizados em *cdrom/db2/install/samples*, em que */cdrom* é a localização da versão instalável do DB2 Spatial Extender. Há dois arquivos de resposta de amostra, um para o servidor Spatial Extender (*db2gse.rsp*) e outro para o cliente Spatial Extender (*db2gsec.rsp*). Os arquivos de resposta de amostra contêm:
 - Palavras-chave exclusivas para instalação
 - Valor/definições de registro para as variáveis de ambiente
 - Parâmetros de configuração do gerenciador de banco de dados
2. Faça alterações em um valor no arquivo de resposta por meio da ativação de um item. Para ativar um item:
 - a. Remova o asterisco (*) à esquerda da variável de ambiente/teclado.
 - b. Apague a definição atual à direita do valor.
 - c. Digite uma nova definição.
 - d. Se fizer alterações, salve o arquivo com um novo nome para preservar o arquivo de resposta de amostra original. Se estiver instalando diretamente a partir do CD-ROM, será necessário armazenar o arquivo de resposta renomeado em um sistema de arquivos local.

Etapa 2. Iniciar uma instalação não-assistida com um arquivo de resposta.

1. Inicie sessão como um usuário com autoridade de root.
2. Digite o comando **db2setup**:

```
/cdrom/db2setup -r responsefile_directory/responsefile_file
```

em que */cdrom* é a localização da imagem instalável do DB2 Spatial Extender, *responsefile_directory* é o diretório onde o arquivo de resposta está localizado e *responsefile_file* é o nome do arquivo de resposta.

3. Verifique as mensagens no arquivo de log quando a instalação for concluída. A localização padrão do arquivo de log é `/tmp/db2setup.log`.

Geração de um arquivo de log de rastreamento para o instalador do DB2: Se você teve problemas com o instalador do DB2, poderá gerar um arquivo de log de rastreamento (`db2setup.trc`). Você pode enviar o arquivo de log de rastreamento e o arquivo `db2setup.log` para o IBM Software Support para diagnósticos adicionais. Quando gerados, os dois arquivos são colocados no diretório `/tmp`.

Para gerar um arquivo de log de rastreamento, execute o comando **db2setup** utilizando o indicador **-d** como segue: `./db2setup -d`. Esse comando aciona o instalador do DB2 para execução no modo de rastreamento. Continue a operar a interface para reproduzir o problema encontrado. Quando concluir, será criado o arquivo de log de rastreamento `/tmp/db2setup.trc`.

Utilização do SMIT ou do comando **installp**

Para instalar o DB2 Spatial Extender utilizando o System Management Interface Tool (SMIT) ou o comando **installp**:

1. Inicie sessão na máquina servidora ou cliente de destino como usuário `root`.
2. Insira o CD-ROM na unidade de CD-ROM.
3. Instale o DB2 Spatial Extender executando o SMIT ou o comando **installp**.

Para executar o SMIT:

- a. Digite o comando **smit install_latest**. O menu de ferramentas SMIT é aberto.
- b. Digite `/cdrom/db2` no dispositivo/diretório INPUT no campo do software.
- c. Clique em **DO** ou pressione Enter para verificar se o diretório de instalação existe.
- d. No campo **Software a ser instalado**, identifique se devem ser instalados os componentes cliente ou servidor.
Para cliente, digite o seguinte: `db2_07_01.gc1n`.
Para servidor, digite o seguinte: `db2_07_01.gsrv`.
- e. Clique em **DO** ou pressione Enter (é solicitado que você confirme os parâmetros de instalação).
- f. Pressione Enter para confirmar.

Os arquivos do produto são instalados do CD-ROM para seu disco rígido, o que requer alguns minutos.

- g. Encerre a sessão.

Estabelecimento do ambiente da instância do DB2 Spatial Extender

O comando **db2icrt** é utilizado para criar novas instâncias do DB2. Todas as novas instâncias do DB2 criadas após a instalação do DB2 Spatial Extender incluem o DB2 Spatial Extender no ambiente da instância.

As instâncias do DB2 criadas antes da instalação do Spatial Extender não incluem o DB2 Spatial Extender em seus ambientes de instância. Para atualizar as instâncias do DB2 já existentes como Spatial Extender, utilize o comando **db2iupdt**.

Como um usuário com autoridade de root, digite o seguinte comando:

```
db2iupdt instance_name
```

O parâmetro *instance_name* é o nome da instância.

No AIX, este utilitário está localizado no seguinte diretório: `/usr/lpp/db2_07_01/instance`. Se precisar de ajuda, digite `db2iupdt -h` na linha de comandos para abrir o menu de ajuda.

Nota: Atualize o ambiente de instância com o Spatial Extender antes de verificar a instalação.

Verificando a instalação

Depois de instalar o DB2 Spatial Extender, crie um banco de dados e execute o programa de verificação de instalação para verificar se o DB2 Spatial Extender está instalado e configurado corretamente.

Nota: Para instalações AIX, verifique se você estabeleceu o ambiente de instâncias do DB2 Spatial Extender antes de executar o programa de verificação de instalação.

Verifique a instalação utilizando o programa de amostra do DB2 Spatial Extender (`runGseDemo`). Os parâmetros de configuração do banco de dados podem ser alterados na linha de comandos com as ferramentas do DB2 ou com a interface do usuário no Centro de Controle do DB2. As instruções a seguir se aplicam ao AIX, Windows NT e ao Windows 2000.

Para verificar a instalação:

1. Inicie sessão como o proprietário da instância (apenas AIX).
2. Aumente a configuração do gerenciador do banco de dados para o tamanho da utilização de memória do UDF com um valor mínimo de

2048. Por exemplo, digite `db2 update dbm cfg using UDF_MEM_SZ 2048`. Se 2048 for inadequado, aumente o parâmetro `UDF_MEM_SZ` em quantidades de 256.

Nota: Os requisitos de memória para o tamanho de utilização de memória do UDF aumentam conforme o número de UDFs mencionados em um aumento do aplicativo. Isso é válido especialmente quando os UDFs espaciais com os tipos de dados espaciais são utilizados como parâmetros de entrada e/ou de saída.

3. Crie um banco de dados. Por exemplo, digite `db2 create database mydb`, em que *mydb* é o nome do banco de dados.
4. Aumente o tamanho do arquivo de log do DB2 de seu banco de dados.
Para aumentar o tamanho do arquivo de log:
 - a. Conecte-se ao banco de dados criado por você. Por exemplo, digite `db2 connect to mydb`, em que *mydb* é o nome do banco de dados.
 - b. Aumente o tamanho do arquivo de log. Por exemplo, digite `db2 update db logfilesize using LOGFILE 1000`.
 - c. Desconecte-se de seu banco de dados. Por exemplo, digite `db2 connect reset`.

Nota: Você deve aumentar o tamanho do arquivo de log do DB2 toda vez que ativar espacialmente um banco de dados.

5. Localize a verificação da instalação. Por exemplo, digite `runGseDemo`.
Para AIX, digite `cd $HOME/sql1lib/samples/spatial`, em que *\$HOME* é o diretório pessoal do proprietário da instância.
No Windows NT e no Windows 2000, digite `cd c:\sql1lib\samples\spatial`, em que *c:\sql1lib* é o diretório no qual você instalou o DB2 Spatial Extender.
6. Execute o programa de verificação da instalação. Por exemplo digite:
`runGseDemo mydb userID password`

O parâmetro *mydb* é o nome do banco de dados.

Dicas de detecção de problemas do programa de amostra

O programa de amostra do DB2 é projetado para que os problemas com a instalação se tornem conhecidos. Durante a verificação da instalação, você pode receber mensagens de erro que podem ajudar a diagnosticar problemas específicos do sistema. A maioria das mensagens de erro são provocadas por um pequeno número de falhas típicas do usuário. Para evitar esses erros, certifique-se de proceder da seguinte forma cada vez que executar o programa de verificação da instalação:

- Certifique-se de ter instalado os produtos cliente e servidor do DB2 Spatial Extender nos ambientes apropriados. Em uma configuração independente, certifique-se de que os produtos cliente e servidor estão instalados.
- Utilize um novo banco de dados que não possua nenhuma operação espacial a ele associada.
- Aumente a configuração do gerenciador do banco de dados para o tamanho da utilização de memória do UDF.
- Aumente o tamanho do arquivo de log.

Administration Client

Se você não selecionou a opção **DB2 Administration Client** (para instalações NT) ou a opção **Spatial Extender Client** (para instalações AIX) quando instalou o DB2 Spatial Extender, receberá a seguinte mensagem de erro: "O nome especificado não é reconhecido como um comando interno ou externo, programa operável ou arquivo batch."

Esse erro ocorre porque o programa de amostra não está disponível no sistema. O programa de amostra é fornecido com o DB2 Administration Client e o Spatial Extender Client. Se o DB2 Administration Client ou o Spatial Extender Client não foram instalados em seu sistema, o programa de amostra também não estará disponível no sistema.

Para corrigir esse problema.

1. Instale novamente o DB2 Spatial Extender. Selecione **DB2 Administration Client** no programa de instalação do Windows NT, Windows 2000, ou **Spatial Extender Client** no instalador do DB2 no AIX.
2. Execute o programa de amostra novamente repetindo as etapas em "Verificando a instalação" na página 26.

O banco de dados já está espacialmente ativado

Você receberá a seguinte mensagem de erro se o banco de dados que está executando o programa de amostra já estiver ativado espacialmente:

```
Ativando o banco de dados logtst...
Returning from ENABLE_DB:
Return code = -14
Return message text =
GSE0014E  O banco de dados já foi ativado para operações espaciais.
```

Para corrigir esse problema, elimine o banco de dados e repita as etapas em "Verificando a instalação" na página 26.

Nota: Verifique se o banco de dados para o qual você está verificando a instalação é novo e não possui operações espaciais associadas assim; nesse caso, o programa de amostra falhará.

Configuração do gerenciador do banco de dados

Você receberá a seguinte mensagem de erro caso tenha falhado ao aumentar o tamanho da utilização de memória do UDF na configuração do gerenciador do banco de dados:

```
Ocorreu um erro SQL inesperado ("SQL0973N Armazenamento
insuficiente disponível no heap do "UDF_MEM" para processar a
instrução. ").  SQLSTATE=57011
```

Para obter instruções sobre como aumentar o tamanho da configuração do gerenciador do banco de dados, consulte a Etapa 2 na página 26 da “Verificando a instalação” na página 26.

Tamanho do arquivo de log

Você receberá a seguinte mensagem de erro caso não tenha aumentado o tamanho do arquivo de log:

```
Ativando o banco de dados logtst...
Returning from EN
ABLE_DB:
Return code = -8
Return message text =
GSE0008E Ocorreu um erro SQL inesperado ("SQL3306N
Ocorreu um erro SQL "-964"
ao inserir uma linha em ").
```

Para obter instruções sobre como aumentar o tamanho do arquivo de log, consulte a Etapa 4 na página 27 em “Verificando a instalação” na página 26.

Considerações pós-instalação

Após a instalação do Spatial Extender, considere o seguinte:

- Download do ArcExplorer Java Versão 3.0
- Utilização dos CD-ROMs para Dados e Mapas e Dados de Referência do Geocoder do DB2 Spatial Extender

Efetando o download do ArcExplorer

O Environmental Systems Research Institute (ESRI) fornece um navegador que pode produzir resultados visuais de consultas para os dados do DB2 Spatial Extender. Este navegador é o ArcExplorer Java Versão 3.0. Você pode fazer o download de uma cópia do ArcExplorer Java Versão 3.0 a partir do site do ESRI na Web, em <http://www.esri.com>. O ArcExplorer requer o Standard Edition ou o Enterprise Edition Java[®] 2 Runtime Environment (JRE), Versão 1.2.2.

Para obter mais informações sobre a instalação e o uso do ArcExplorer Java Versão 3.0, consulte o manual *Utilização do ArcExplorer*, que está disponível no site do ESRI na Web.

Importante: O DB2 Universal Database Versão 7.1 é fornecido com o IBM Java Development Kit (JDK) Versão 1.1.8. Quando você instala o JRE Versão 1.2.2 para ArcExplorer, coloque-o em um diretório separado do DB2. Lembre-se de definir a variável de ambiente CLASSPATH.

Utilização dos CD-ROMs para Dados e Mapas e Dados de Referência do Geocoder do DB2 Spatial Extender

O DB2 Spatial Extender é fornecido com cinco CD-ROMs de dados e mapas e um CD-ROM de dados do geocoder.

Dados e Mapas do DB2 Spatial Extender

As informações sobre os dados e os mapas, etiquetadas "Dados e Mapas do DB2 Spatial Extender 1 – 5", são fornecidas em cinco CD-ROMs. O Tabela 2 fornece um resumo dos dados localizados em cada CD-ROM.

Tabela 2. Informações sobre o CD-ROM de dados e mapas

| CD-ROM de Dados e Mapas | Tipo de resumo de dados do mapa |
|-------------------------|--|
| CD-ROM 1 | Canadá, Europa, México, Estados Unidos e Mundial |
| CD-ROM 2 | Estados Unidos (detalhado) |
| CD-ROM 3 | Estados Unidos (região oeste) |
| CD-ROM 4 | Estados Unidos (região leste) |
| CD-ROM 5 | Estados Unidos (região sul) e dados de imagem de amostra |

Para obter uma descrição detalhada dos dados fornecidos pelo ESRI, consulte o arquivo de auxílio do ESRI, *esridata.hlp*, localizado no CD-ROM de Dados e Mapas do DB2 Spatial Extender.

- Para Windows NT e Windows 2000, exiba o arquivo de auxílio em *x:esridata.hlp*, em que *x* é a unidade do CD-ROM.
- Para AIX, exiba ou imprima o arquivo de auxílio localizado no CD-ROM em */cdrom/esridata.hlp*, em que */cdrom* é o ponto de montagem.

Dados de Referência do Geocoder do DB2 Spatial Extender

Os dados de referência do geocoder no CD-ROM de Dados de Referência do Geocoder do DB2 Spatial Extender são criados especificamente para funcionarem com o geocoder padrão do DB2 Spatial Extender. Ele é composto de dados de rede de ruas do mapa básico dos Estados Unidos que o geocoder padrão utiliza para determinar a latitude e a longitude de endereços em um banco de dados ativado espacialmente. Esses dados do mapa básico é coletivamente chamado "dados de referência". O geocoder padrão obtém os dados de endereço (não espacial) em seu banco de dados, compara-os com os

dados de referência e os convertem em coordenadas que podem ser armazenadas pelo DB2 Spatial Extender. Esse processo é chamado *geocodificação*.

Para obter mais informações sobre a geocodificação, consulte “Utilizando os geocoders” na página 53.

Acesso aos dados de referência do geocoder: Os dados do geocoder podem ser acessados diretamente do CD-ROM ou podem ser copiados em seu disco rígido. Para copiar os arquivos de dados do geocoder do CD-ROM no ambiente do servidor DB2 Spatial Extender, execute as etapas explicadas nesta seção.

Para AIX:

1. Monte o CD-ROM. Para obter instruções sobre como montar um CD-ROM, consulte “Montagem do CD-ROM” na página 22.
2. Inicie sessão na máquina servidora de destino como um usuário com autoridade de root.
3. Digite o seguinte:

```
cp /cdrom/db2/* /usr/lpp/db2_07_01/gse/refdata/
```
4. Encerre a sessão.

No Windows NT e no Windows 2000, você pode utilizar a janela de Comando ou o Windows Explorer.

Para utilizar a janela de Comando para acessar os dados do geocoder:

1. Clique em **Iniciar** -> **Programa** -> **IBM DB2** -> **Janela de Comando**.
2. Digite o seguinte:

```
copy d:\db2\* %db2path%\gse\refdata
```

Substitua *d:* pela letra que corresponde a sua unidade de CD-ROM.

Para utilizar o Windows Explorer para acessar os dados do geocoder:

Copie todos os arquivos do *d:\db2* em *c:\sqllib\gse\refdata*, em que *d:* é a unidade de CD-ROM e *c:\sqllib* é o diretório em que o DB2 está instalado.

Fornecimento do arquivo EDGELocator.loc para o geocoder padrão: Os dados de referência fornecidos no CD-ROM incluem o arquivo EDGELocator.loc. O arquivo EDGELocator.loc é utilizado pelo geocoder padrão para localizar dados de referência específicos. Por exemplo, se estiver geocodificando endereços na Califórnia, Kentucky e Oregon, o geocoder padrão utilizará o arquivo localizador para determinar as localizações de endereço no CD-ROM.

Ao ativar geocodificação incremental (conhecida também como geocodificação automática) com o geocoder padrão, ou ao executar o geocoder padrão no modo batch, será necessário utilizar o parâmetro de entrada **vendorSpecific**. Quando você especifica o parâmetro de entrada **vendorSpecific**, o nome do arquivo e o caminho do diretório do arquivo localizador devem ser passados para o geocoder.

Por exemplo, o comando de amostra a seguir **gseadm**, é utilizado para invocar a geocodificação do modo batch com o geocoder padrão:

```
gseadm run_gc database_name -layerSchema inst1 -layerTable myTable  
-layerColumn column1 -gcId 1 -vendorSpecific c:\sqlib\gse\refdata\EDGELocator.loc
```

Para obter mais informações sobre como executar o geocoder e como utilizar o parâmetro **vendorSpecific**, consulte “Capítulo 4. Definindo colunas espaciais, registrando-as como camadas e ativando um geocoder para mantê-las” na página 45 e “Capítulo 5. Preenchendo colunas espaciais” na página 53.

Invocação do Spatial Extender

Após o Spatial Extender ser instalado, você pode utilizar o Centro de Controle do DB2 para configurar o ambiente GIS e começar a trabalhar com as informações espaciais.

Para invocar o Spatial Extender a partir do Centro de Controle do DB2:

1. Na janela Centro de Controle, clique no servidor em que deseja que o Spatial Extender seja executado.
2. Clique na pasta **Bancos de Dados**. Os bancos de dados são exibidos na área de janela conteúdo.
3. Clique com o botão direito do mouse sobre o banco de dados com o qual deseja trabalhar e, em seguida, clique na operação espacial no menu popup que deseja executar.

Capítulo 3. Configurando recursos

Depois de instalar o Spatial Extender, você está pronto para fornecer ao seu banco de dados os recursos necessários para criar colunas espaciais e manipular dados espaciais. Este capítulo faz uma síntese destes recursos e descreve duas das tarefas através das quais você os torna disponíveis: ativando seu banco de dados para operações espaciais e criando sistemas de referência espacial.

Inventário dos recursos

Os recursos aos quais você recorre para criar colunas espaciais e manipular dados espaciais são:

- *Dados de referência*: endereços que o Spatial Extender confere para verificar endereços nos quais você deseja efetuar o geocode
- Recursos que ativam um banco de dados para operações espaciais: procedimentos armazenados, funções espaciais e outros
- Geocoders não-padrões fornecidos pelos usuários e fornecedores
- Sistemas de referência espacial

Esta seção discute os dados de referência e os recursos que ativam um banco de dados para operações espaciais. Para obter informações sobre geocoders não-padrões, consulte “Sobre geocodificação” na página 53. Para obter mais informações sobre sistemas de referência espacial, consulte “Sobre sistemas de coordenadas e de referência espacial” na página 35.

Dados de referência

Os *dados de referência* consistem nos endereços mais recentes nos Estados Unidos coletados pela Repartição de Censo do Estados Unidos (United States Census Bureau). Para que o geocoder padrão possa converter em coordenadas um endereço em seu banco de dados, ele deve primeiro igualar parte ou todo o endereço a um endereço nos dados de referência.

Os dados de referência tornam-se disponíveis quando o Spatial Extender é instalado. Para a quantia de espaço em disco que estes dados requerem, consulte “Requisitos de espaço em disco” na página 19. Para verificar no AIX se os dados foram carregados adequadamente, procure-os no diretório `$DB2INSTANCE/sqlib/gse/refdata/`. Para verificar no Windows NT se os dados foram carregados adequadamente, procure-os no diretório `%DB2PATH%\gse\refdata\`.

Recursos que ativam um banco de dados para operações espaciais

A primeira etapa que você executa depois de instalar o Spatial Extender é ativar seu banco de dados para operações espaciais. Isto envolve iniciar uma ação que faz com que o Spatial Extender carregue o banco de dados com os seguintes recursos:

- Procedimentos armazenados. Quando você solicita uma ação do Centro de Controle, o Spatial Extender chama um destes procedimentos armazenados para executar a ação.
- Tipos de dados espaciais. Você deve atribuir um tipo de dados espacial a cada coluna da tabela ou da exibição em que os dados espaciais devem ser armazenados. Para obter mais informações, consulte “Sobre tipos de dados espaciais” na página 45.
- Tabelas e exibições de catálogo do Spatial Extender. Algumas operações dependem do catálogo do Spatial Extender. Por exemplo, para que uma coluna com tipo de dados espaciais possa ser preenchida, ela deve ser registrada no catálogo como uma camada. Para obter informações sobre camadas, consulte “Desenvolvendo e implementando um projeto do GIS” na página 10.
- Um tipo de índice espacial. Ele permite que você defina índices para camadas.
- Funções espaciais. Você as utiliza para trabalhar com dados espaciais de diversas formas; por exemplo, para determinar relacionamentos entre recursos geográficos e para gerar mais dados espaciais. Uma destas funções é um geocoder padrão. Ele converte endereços nos Estados Unidos em coordenadas e depois insere estas coordenadas em colunas espaciais. Para obter mais informações sobre funções espaciais, consulte “Capítulo 13. Figuras geométricas e funções espaciais associadas” na página 157 e “Capítulo 14. Funções espaciais para consultas SQL” na página 195. Para obter mais informações sobre o geocoder padrão, consulte “Sobre geocodificação” na página 53.
- Um esquema, chamado DB2GSE, que contém os objetos que acabaram de ser relacionados.

Para obter instruções sobre como usar o Centro de Controle para iniciar o carregamento destes recursos, consulte “Ativando um banco de dados para operações espaciais”. Para obter diretrizes sobre o uso de uma rotina em um programa de aplicação para executar a mesma tarefa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Ativando um banco de dados para operações espaciais

Para descobrir qual autorização é necessária para ativar um banco de dados para operações espaciais, consulte “Autorização” na página 94.

Para ativar um banco de dados para operações espaciais a partir do Centro de Controle:

1. Da janela do Centro de Controle, expanda a árvore de objetos até encontrar a pasta **Bancos de Dados** sob o servidor no qual deseja que o Spatial Extender seja executado.
2. Clique na pasta **Bancos de Dados**. Os bancos de dados são exibidos no painel de conteúdos no lado direito da janela.
3. Dê um clique com o botão direito no banco de dados desejado e clique em **Spatial Extender** —> **Ativar** no menu instantâneo. O Spatial Extender fornece ao banco de dados os recursos que permite a você criar e trabalhar com colunas e dados espaciais.

Lembrete: Para poder ativar um banco de dados para operações espaciais, o Spatial Extender deve estar instalado no servidor onde o banco de dados reside.

Criando um sistema de referência espacial

Esta seção descreve o relacionamento entre sistemas de referência espacial e sistemas de coordenadas e explica como criar um sistema de referência espacial a partir do Centro de Controle.

Sobre sistemas de coordenadas e de referência espacial

Esta seção continua a discussão sobre sistemas de coordenadas que foi iniciada em “A natureza dos dados espaciais” na página 6. Depois expande-se para definição de sistemas de referência espacial fornecida em “Desenvolvendo e implementando um projeto do GIS” na página 10. Ela também fornece diretrizes para determinação dos valores que devem ser atribuídos aos parâmetros do sistema de referência espacial.

Sistemas de coordenadas, coordenadas e medidas

Você pode imaginar um sistema de coordenadas como sendo uma grade imaginária que encobre uma área geográfica específica. Exemplos seriam uma grade que cobre a terra, uma grade que cobre uma nação ou uma grade que cobre uma região em um estado. Cada recurso geográfico na área está situado na interseção de uma linha da grade leste-oeste e uma norte-sul. Um valor, chamado de *coordenada X*, indica a localização na linha da grade leste-oeste. Outro valor, uma *coordenada Y*, indica a localização na linha da grade norte-sul. Ambos os valores referenciam a localização até o centro da grade ou a *origem*.

As coordenadas X e Y na origem são zero. Da origem rumo leste, as coordenadas X são positivas; da origem rumo oeste, elas são negativas. De modo similar, da origem rumo norte, as coordenadas Y são positivas; da origem rumo sul, elas são negativas. Para uma ilustração desta distribuição, considere o seguinte exemplo generalizado: O sistema de coordenadas A

contém uma grade que abrange uma grande área metropolitana. Uma coordenada X de 7 denotaria uma posição que é de sete unidades de medida rumo leste a partir da origem desta grade. Uma coordenada X de -9.5 denotaria uma posição que é de nove e meio unidades de medida rumo oeste a partir da origem.

Cada item de dados em uma coluna espacial inclui ou (1) uma coordenada X e uma coordenada Y que define a localização de um recurso geográfico ou (2) várias coordenadas X e Y que definem as localizações das partes de um recurso ou que definem a área que um recurso abrange. Dois outros tipos de valores — uma *coordenada Z* e uma *medida* — também podem ser incluídos. Ao contrário das coordenadas X e Y, as coordenadas Z e as medidas não são utilizadas no Spatial Extender para definir localizações ou áreas. Ao invés disso, elas simplesmente transmitem as informações exigidas por uma aplicação GIS. Uma coordenada Z geralmente indica a altura ou profundidade de um recurso geográfico. As coordenadas Z acima da origem são positivas; as coordenadas Z abaixo dela são negativas. Uma medida é numérica; ela pode transmitir qualquer tipo de informação. Suponha, por exemplo, que você está representando poços de petróleo em seu GIS. Se você solicitar que suas aplicações processem valores que denotam IDs de ponto de detonação para dados sísmicos, você pode armazenar estes valores como medidas.

Sistemas de referência espacial, deslocamentos e fatores de escala

Como indicado no “Sistemas de coordenadas, coordenadas e medidas” na página 35, as coordenadas podem ser negativas e expressas em decimais. O mesmo é verdadeiro para medidas. No entanto, para reduzir armazenamentos suplementares, o Spatial Extender armazena cada coordenada e medida como um inteiro não-negativo (isto é, como um inteiro positivo ou como zero). Sendo assim, coordenadas e medidas negativas e decimais devem ser convertidas em inteiros não-negativos, para que o Spatial Extender possa armazená-las. Além disso, é necessário instruir o Spatial Extender como ele deve fazer a conversão. Isto é feito através da definição de alguns parâmetros. Definições de parâmetro que devem ser utilizadas para converter coordenadas e medidas dentro de uma área geográfica específica são coletivamente chamadas de *sistema de referência espacial*.

Você pode criar um sistema de referência espacial:

- Primeiro, determinando as coordenadas e medidas negativas mais baixas para os recursos que você está representado. (Quanto mais distante do zero um valor negativo estiver, mais baixo ele é. Uma coordenada X de -10 é mais baixa que uma coordenada X de -5; uma medida de -100 é mais baixa que uma medida de -50.)
- Em seguida, especificando os *fatores de deslocamento* (ou *deslocamentos*, abreviando): valores que, quando subtraídos de coordenadas e medidas negativas, deixam números não negativos.

- Em seguida, especificando *fatores de escala*: valores que, quando multiplicados por coordenadas e medidas decimais, produzem inteiros cuja precisão é, pelo menos, a mesma que a das coordenadas e a das medidas. Considere, por exemplo, uma coordenada com uma precisão de quatro: 92.77. Você poderia multiplicá-la por um fator de escala de 100 para obter um inteiro com uma precisão de quatro: 9277. Observe que ao criar um sistema de referência espacial, os fatores de deslocamento são aplicados antes dos fatores de escala.

Determinando as mais baixas coordenadas e medidas negativas

O DB2 Spatial Extender pode armazenar coordenadas e medidas se forem inteiros positivos, mas não se forem números negativos ou decimais. Portanto, é necessário converter as coordenadas e medidas negativas em positivas e as coordenadas e medidas decimais em inteiros. Para concretizar essa conversão, você define um conjunto de parâmetros que, quando aplicado às medidas de coordenadas negativas ou decimais, produzem inteiros positivos. Esse conjunto de parâmetros é chamado *sistema de referência espacial*. Os parâmetros utilizados para converter os valores negativos são chamados *fatores de deslocamento*, os utilizados para converter valores decimais são chamados *fatores de escala*.

Quando você invoca uma função espacial que toma como entrada uma coordenada ou medida decimal e o identificador de um sistema de referência espacial, a função multiplica a coordenada ou a medida decimal por um fator de escala dentro do sistema. O resultado é um inteiro armazenado pelo DB2 Spatial Extender. O fator de escala precisa ser grande o suficiente para garantir que a precisão desse inteiro seja a mesma precisão da coordenada decimal.

Por exemplo, suponha que a função `ST_Point` receba uma entrada que consista em uma coordenada X de 10.01, e uma coordenada Y de 20.03 e o identificador de um sistema de referência espacial. Quando `ST_Point` é invocada, ela multiplica o valor de 10.01 e o valor de 20.03 pelo fator de escala do sistema de referência espacial para as coordenadas X e Y. Se esse fator de escala for 10, os inteiros resultantes que o DB2 Spatial Extender armazena serão 100 e 200, respectivamente. Devido a precisão desses inteiros (3) ser menor que a precisão das coordenadas (4), não será possível ao DB2 Spatial Extender converter esses inteiros de volta para as coordenadas originais ou derivar delas valores que sejam consistentes com o sistema de coordenadas ao qual essas coordenadas pertencem. Porém se o fator de escala for 100, os inteiros resultantes que o DB2 Spatial Extender armazena serão 1001 e 2003 — valores que podem ser convertidos de novo para as coordenadas originais ou dos quais as coordenadas compatíveis podem ser derivadas.

Antes de definir parâmetros para um sistema de referência espacial, é preciso determinar a mais baixa coordenada X, coordenada Y, coordenada Z e medida negativas na área geográfica que contém os recursos sobre os quais você deseja obter informações. Para descobrir quais são estes valores, responda as seguintes perguntas:

- Dos recursos que você está representando, algum deles fica a oeste da origem do sistema de coordenadas sendo utilizado? Caso haja algum, qual coordenada X indica a localização ou borda oeste do recurso que está mais a oeste? (A resposta será a mais baixa das coordenadas X negativas com as quais você está lidando.) Se, por exemplo, você estivesse representando poços de óleo, e alguns deles ficam a oeste da origem, qual coordenada X indica a localização do poço de petróleo que está na extremidades oeste?
- Há algum recurso a sul da origem? Se houver, qual coordenada Y indica a localização ou borda sul do recurso que está mais a sul? (A resposta será a mais baixa das coordenadas Y negativas com as quais você está lidando.) Se, por exemplo, você está representando poços de óleo, e alguns deles ficam a sul da origem, qual coordenada Y indica a localização do poço de petróleo que está na extremidade sul?
- Se você for utilizar coordenadas Z para definir profundidades, qual recurso é o mais profundo e qual coordenada Z representa o ponto mais baixo deste recurso? (A resposta será a mais baixa das coordenadas Z negativas com as quais você está lidando.)
- Se você for incluir medidas em seus dados espaciais, alguma delas será negativa? Caso seja, qual a mais baixa das medidas negativas?

Tendo determinado as coordenadas e medidas negativas mais baixas, inclua em cada uma uma quantia entre cinco e dez por cento de seu valor. Se, por exemplo, a coordenada X negativa mais baixa for -100 , você poderia incluir -5 na mesma. Este manual chama o número resultante de *valor aumentado*.

Nota: O identificador do sistema de referência espacial padrão do DB2 Spatial Extender é 0 (zero). O DB2 Spatial Extender fornece um sistema de referência espacial a ser utilizado com o geocoder padrão. Esse identificador de sistema é 1.

Especificando valores de deslocamento

Em seguida, especifique quais fatores de deslocamento o Spatial Extender deve utilizar para converter coordenadas e medidas negativas em não-negativas:

- Depois de decidir qual será o valor X aumentado, especifique um deslocamento que, quando subtraído deste valor, sobre zero. O Spatial Extender irá então subtrair este número de todas as coordenadas X negativas até chegar a um valor positivo. O Spatial Extender também irá subtrair este número de todas as outras coordenadas X.

Se, por exemplo, o valor X aumentado for -105 , é preciso subtrair -105 do mesmo para obter 0. Depois, o Spatial Extender irá subtrair -105 de todas as coordenadas X associadas aos recursos sendo representados. Como nenhuma destas coordenadas é maior que -100 , todos os valores que resultam da subtração serão positivos.

- De modo similar, especifique deslocamentos que tenham sobre 0 quando subtraídos do valor Y aumentado, o valor Z aumentado e a medida aumentada.

O deslocamento subtraído das coordenadas X é chamado de *X falso*. Os deslocamentos subtraídos das coordenadas Y, coordenadas Z e medidas são chamados de *Y falso*, *Z falso* e *M falso*, respectivamente. Para obter instruções sobre a especificação destes parâmetros, a partir do Centro de Controle, consulte “Criando um sistema de referência espacial a partir do Centro de Controle” na página 40.

Especificando fatores de escala

Em seguida, especifique os fatores de escala que o Spatial Extender deve usar para converter coordenadas e medidas decimais em inteiros:

- Especifique um fator de escala que, quando multiplicado por uma coordenada X decimal ou uma coordenada Y decimal, resulte em um inteiro de 32 bits. É aconselhável fazer deste fator de escala um fator de 10: 10 elevado a um (10), 10 elevado a dois (100), 10 elevado a três (1000), ou, se necessário, um fator maior. Para decidir qual fator de 10 o fator de escala deve ser:
 1. Determine quais coordenadas X e Y são, ou provavelmente serão, números decimais. Suponha, por exemplo, que das várias coordenadas X e Y com as quais você estará lidando, você determina que três delas são números decimais: 1.23, 5.1235 e 6.789.
 2. Pegue a coordenada decimal com a precisão decimal mais longa. Depois, determine por qual fator de dez esta coordenada pode ser multiplicada para que resulte em um inteiro de igual precisão. Para ilustrar: das três coordenadas decimais no exemplo atual, 5.1235 possui a precisão decimal mais longa. Multiplicando-a por dez elevado a quatro (10000) o resultado seria o inteiro, 51235.
 3. Determine se o inteiro produzido pela multiplicação que acabou de ser descrita é muito longo para ser armazenado como um item de dados de 32 bits. 51235 não é longo demais. Mas suponha que além de 1.23, 5.11235 e 6.789, a faixa das coordenadas X e Y inclui um quarto valor decimal, 10006.789876. Como esta precisão decimal da coordenada é mais longa do que aquela das outras três, você multiplicaria *esta* coordenada — e não 5.1235 — por um fator de 10. Para convertê-la em um inteiro, você poderia multiplicá-la por 10 elevado a seis (1000000). Mas o valor resultante, 10006789876, é longo demais para ser

armazenado como um item de dados de 32 bits. Se o Spatial Extender tentasse armazená-lo, os resultados seriam imprevisíveis.

Para evitar este problema, selecione um fator de 10 que, quando multiplicado pela coordenada original, resulte em um número decimal que o Spatial Extender pode truncar transformando-o em um inteiro armazenável, com perda mínima de precisão. Neste caso, você selecionaria 10 elevado a quatro (10000). A multiplicação de 10000 por 10006.789876 resulta em 100067898.76. O Spatial Extender truncaria este número transformando-o em 100067898, reduzindo sua precisão por uma quantia virtualmente insignificante.

- Se os recursos que você está representando possuem coordenadas Z decimais, siga o procedimento anterior para determinar um fator de escala para estas coordenadas. Se os recursos estiverem associados a medidas decimais, siga o mesmo procedimento para determinar um fator de escala para estas medidas.

O fator de escala para as coordenadas X e Y é chamado de *unidade XY*. Os fatores de escala para coordenadas Z e medidas são chamados de *unidades Z* e *unidades M*, respectivamente. Para obter instruções sobre a especificação destes parâmetros, a partir do Centro de Controle, consulte “Criando um sistema de referência espacial a partir do Centro de Controle”.

Criando um sistema de referência espacial a partir do Centro de Controle

Esta seção apresenta uma visão geral das etapas para criação de um sistema de referência espacial a partir do Centro de Controle. A visão geral está seguida por detalhes de como completar cada etapa.

Não é necessária nenhuma autorização para realização destas etapas.

Visão geral das etapas para criação de um sistema de referência espacial a partir do Centro de Controle:

1. Abra a janela Criar Sistema de Referências Espaciais.
2. Indique o sistema de coordenadas que você deseja utilizar.
3. Especifique identificadores para o sistema de referência espacial que você deseja criar.
4. Determine que faixas de coordenadas e medidas se aplicam aos recursos geográficos sobre os quais você quer obter informações.
5. Especifique valores que podem ser usados para converter coordenadas e medidas negativas ou decimais em itens de dados que o Spatial Extender pode armazenar.
6. Peça ao Spatial Extender para criar o sistema de referência espacial desejado.

Etapas detalhadas para criação do sistema de referência espacial a partir do Centro de Controle:

1. Abra a janela Criar Sistema de Referências Espaciais.
 - a. Da janela do Centro de Controle, expanda a árvore de objetos até encontrar a pasta **Bancos de Dados** sob o servidor no qual deseja que o Spatial Extender seja executado.
 - b. Clique na pasta **Bancos de Dados**. Os bancos de dados são exibidos no painel de conteúdos no lado direito da janela.
 - c. Dê um clique com o botão direito no banco de dados que foi ativado para dados espaciais e clique em **Spatial Extender** —> **Referências Espaciais** no menu instantâneo. A janela Referências Espaciais aparece.
 - d. A partir da janela Referências Espaciais, clique em **Criar**. A janela Criar Sistema de Referências Espaciais é aberta.
2. Da janela Criar Referências Espaciais, utilize o campo **Sistema de coordenadas** para indicar o sistema de coordenadas que deseja utilizar.
3. Especifique identificadores para o sistema de referência espacial que você deseja criar.
 - No campo **Nome**, digite um nome com 1 a 64 caracteres para o sistema.

Restrição: Não especifique o nome de outro sistema de referência espacial. Não pode haver dois sistemas de referência espacial no banco de dados com o mesmo nome.
 - No campo da ID, digite um identificador numérico. Ele deve ser um inteiro.

Restrição: Não especifique a ID de outro sistema de referência espacial. Não pode haver dois sistemas de referência espacial no banco de dados com a mesma ID.
4. Utilizando um meio externo ao Centro de Controle — por exemplo um papel ou um quadro branco — determine as coordenadas e medidas negativas mais baixas que se aplicam aos recursos geográficos sendo representados. Para obter diretrizes sobre como fazer isto, consulte “Determinando as mais baixas coordenadas e medidas negativas” na página 37.
5. Da janela Criar Referências Espaciais, especifique os valores para converter coordenadas e medidas negativas ou decimais em itens de dados que o Spatial Extender suporta — ou seja, em inteiros não negativos de 32 bits.
 - a. Especifique valores para converter coordenadas X negativas ou decimais em inteiros não-negativos:
 - Na coluna **Deslocamento**, no campo mais próximo de X, especifique um X falso:

- Se algum dos valores dentro do intervalo de coordenadas X identificados na etapa 4 na página 41 for negativo, digite um X falso que, quando subtraído da mais baixa coordenada negativa, tenha um número positivo como sobra. Para obter orientações, consulte “Especificando valores de deslocamento” na página 38.
- Se todas as coordenadas X forem não-negativas, digite um X falso igual a 0.
- Na coluna **Fator de escala**, especifique uma unidade XY no campo na extremidade direita do X. Esta unidade XY deve ser uma que, quando multiplicada por qualquer coordenada X decimal ou coordenada Y decimal, resulta em um número inteiro que pode ser armazenado como um item de dados de 32 bits, com perda mínima de precisão. Para obter orientações, consulte “Especificando fatores de escala” na página 39.
Depois de especificar a unidade XY no campo na extremidade direita do X, ela também aparece no campo na extremidade direita do Y.
- b. Especifique um Y falso que irá permitir que o Spatial Extender converta coordenadas Y negativas em valores positivos. Isto é feito na coluna **Deslocamento**, no campo mais próximo do Y:
 - Se algum dos valores dentro do intervalo de coordenadas Y identificados na etapa 4 na página 41 for negativo, digite um Y falso que, quando subtraído da coordenada negativa mais baixa, tenha um número positivo como sobra. Para obter orientações, consulte “Especificando valores de deslocamento” na página 38.
 - Se todas as coordenadas Y forem positivas, digite um Y falso igual a 0.
- c. Se você for incluir coordenadas Z em seus dados espaciais, especifique valores para converter coordenadas Z negativas ou decimais em inteiros não-negativos:
 - Na coluna **Deslocamento**, no campo mais próximo de Z, digite um Z falso:
 - Se algum dos valores dentro do intervalo de coordenadas Z identificados na etapa 4 na página 41 for negativo, digite um Z falso que, quando subtraído da coordenada negativa mais baixa, tenha um número positivo como sobra. Para obter orientações, consulte “Especificando valores de deslocamento” na página 38.
 - Se todas as coordenadas Z forem não-negativas, digite um Z falso igual a 0.
 - Na coluna **Fator de escala**, especifique uma unidade Z no campo na extremidade direita de Z. Esta unidade Z deve ser uma que, quando multiplicada por qualquer coordenada Z decimal, resulte em um número inteiro que pode ser armazenado como item de dados de 32

- bits, com perda de precisão mínima. Para obter orientações, consulte “Especificando fatores de escala” na página 39.
- d. Se você for incluir medidas em seus dados espaciais, especifique valores para converter medidas negativas ou decimais em inteiros positivos:
- Na coluna **Deslocamento**, no campo mais próximo do rótulo **Linear**, digite um M falso:
 - Se algum dos valores dentro do intervalo de medidas identificados na etapa 4 na página 41 for negativo, digite um M falso que, quando subtraído da medida negativa mais baixa, tenha um número positivo como sobra. Para obter orientações, consulte “Especificando valores de deslocamento” na página 38.
 - Se todas as medidas forem positivas, digite um 0 para M falso.
 - Na coluna **Fator de escala**, especifique uma unidade M no campo na extremidade direita do rótulo **Linear**. Esta unidade M deve ser uma que, quando multiplicada por qualquer medida decimal, resulte em um número inteiro que pode ser armazenado como um item de dados de 32 bits, com perda mínima de precisão. Para obter orientações, consulte “Especificando fatores de escala” na página 39.
6. Clique em **OK** para criar o sistema de referência espacial desejado.

Capítulo 4. Definindo colunas espaciais, registrando-as como camadas e ativando um geocoder para mantê-las

Depois de configurar os recursos para o seu GIS do Spatial Extender, você está pronto para criar objetos que irão conter dados espaciais. Por exemplo, se precisar de novas tabelas para conter dados espaciais, você pode defini-las, atribuindo tipos de dados espaciais às colunas nas quais deseja que os dados sejam colocados. Se precisar incluir colunas espaciais em tabelas existentes, você também pode fazê-lo.

Quando você fornece uma tabela nova ou existente com uma coluna espacial, é preciso registrar esta coluna como uma camada. Além disso, se pretende ter um geocoder preenchendo a coluna, você pode, no momento em que registra a coluna como uma camada, ativar o geocoder para mantê-la automaticamente. Esta ativação ocorre da seguinte maneira: o Spatial Extender define disparadores que são codificados para chamar o geocoder sempre que a coluna (ou colunas) do atributo correspondente da coluna espacial recebe dados novos ou atualizados. Quando solicitado, o geocoder converte os dados novos ou atualizados em dados espaciais e coloca estes dados espaciais na coluna espacial.

Depois de definir uma coluna espacial para uma tabela, você pode, se quiser, criar uma coluna de exibição sobre esta coluna da tabela. Você deve registrar a coluna de exibição como uma camada depois de registrar a coluna da tabela como uma camada.

Esta capítulo discute a natureza e o uso dos tipos de dados que você pode atribuir à coluna espacial. Em seguida, o capítulo explica como utilizar o Centro de Controle para definir uma coluna espacial para uma tabela, para registrar esta coluna como uma camada e para ativar o geocoder para mantê-la. Por final, o capítulo explica como usar o Centro de Controle para registrar uma coluna de exibição como uma camada.

Sobre tipos de dados espaciais

Esta seção introduz os tipos de dados exigidos para colunas espaciais e fornece diretrizes para escolha de qual deve ser o tipo de dados da coluna espacial.

Quando uma banco de dados é ativado para operações espaciais, o Spatial Extender fornece ao banco de dados uma hierarquia de tipos de dados estruturados. A Figura 6 na página 46 apresenta esta hierarquia. Nesta figura,

os tipos que podem ser instanciados tem o plano de fundo em branco; os tipos que não podem ser instanciados possuem um plano de fundo sombreado.

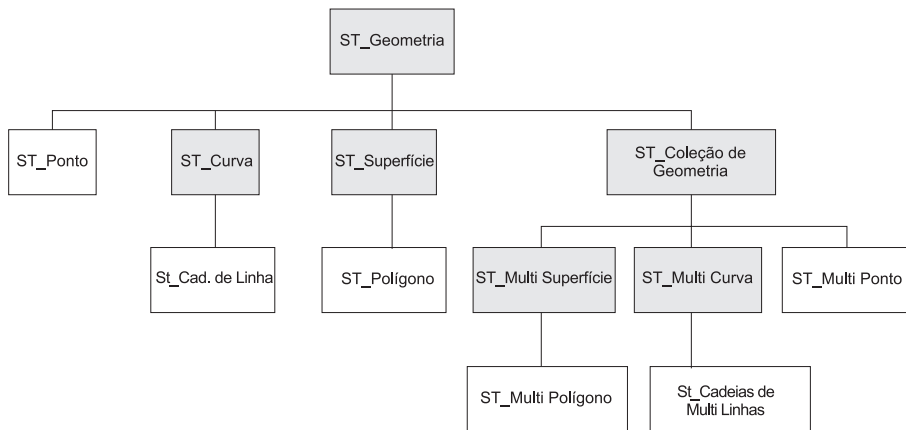


Figura 6. Hierarquia dos tipos de dados espaciais. Tipos de dados em caixas brancas podem ser instanciados. Os tipos de dados nomeados nas caixas sombreadas não são instanciáveis.

A hierarquia na Figura 6 contém:

- Tipos de dados para recursos geográficos que podem ser captados formando uma única unidade; por exemplo, residências individuais e lagos isolados.
- Tipos de dados para recursos geográficos que são compostos por várias unidades ou componentes; por exemplo, sistemas de estradas e cadeias de montanhas.
- Um tipo de dados para recursos geográficos de todos os tipos.

Tipos de dados para recursos de uma única unidade

Utilize ST_Point, ST_LineString, ST_Polygon para armazenar coordenadas que definem o espaço ocupado pelos recursos que podem ser percebidos como formando uma única unidade:

- Utilize ST_Point quando quiser indicar o ponto no espaço que é ocupado por um recurso geográfico separado. O recurso pode ser muito pequeno, como um poço de água; pode ser muito grande, como uma cidade; ou pode ser intermediário, como um conjunto de prédios ou um parque. Em cada caso, o ponto no espaço pode estar localizado na interseção de uma linha de coordenada leste-oeste (por exemplo, uma paralela) e uma norte-sul (por exemplo, um meridiano). Um item de dados ST_Point contém valores — uma coordenada X e uma coordenada Y — que definem tal interseção. A coordenada X indica onde a interseção está situada na linha leste-oeste; a coordenada Y indica onde a interseção está situada na linha norte-sul.

- Utilize `ST_LineString` para coordenadas que definem o espaço que é ocupado por recursos lineares; por exemplo, ruas, canais e canalizações.
- Utilize `ST_Polygon` quando quiser indicar a extensão do espaço abrangido por um recurso multifacetado; por exemplo, um distrito, uma floresta ou um habitat selvagem. Um item de dados `ST_Polygon` consiste em coordenadas que definem o perímetro de tal recurso.

Em alguns casos, `ST_Polygon` e `ST_Point` podem ser usados para o mesmo recurso. Por exemplo, suponha que você precise de informações espaciais sobre diversos conjuntos de apartamentos. Se quiser representar o ponto no espaço onde cada complexo está localizado, você usa `ST_Point` para armazenar as coordenadas X e Y que definem cada ponto. Por outro lado, se quiser representar a área que cada conjunto abrange, você utiliza `ST_Polygon` para armazenar as coordenadas que definem o perímetro de cada área.

Tipos de dados para recurso de multi-unidades

Utilize `ST_MultiPoint`, `ST_MultiLineString` e `ST_MultiPolygon` para armazenar coordenadas que definem espaços ocupados pelos recursos que são compostos por várias unidades:

- Utilize `ST_MultiPoint` quando quiser representar recursos compostos por unidades separadas e quiser indicar o ponto no espaço ocupado por cada componente. Um item de dados `ST_MultiPoint` inclui os pares de coordenadas X e Y que definem a localização de cada componente de tal recurso. Por exemplo, considere uma tabela cujas linhas representam arquipélagos e cujas colunas incluem uma coluna `ST_MultiPoint`. Cada item de dados nesta coluna contém os pares de coordenadas X e Y que definem as localizações das ilhas em cada arquipélago.
- Utilize `ST_MultiLineString` quando quiser representar recursos compostos por unidades lineares e quiser obter informações sobre o espaço ocupado por cada unidade. Um item de dados `ST_MultiLineString` consiste em coordenadas que definem tais espaços. Por exemplo, considere uma tabela cujas linhas representam sistemas de rios e cujas colunas incluem uma coluna `ST_MultiLineString`. Cada item de dados nesta coluna inclui os conjuntos de coordenadas que definem os caminhos dos rios em cada sistema.
- Utilize `ST_MultiPolygon` quando quiser representar recursos compostos por unidades multifacetadas e quiser obter informações sobre o espaço ocupado por cada unidade. Por exemplo, considere uma tabela cujas linhas representam municípios do meio-oeste e cujas colunas incluem uma coluna `ST_MultiPolygon`. Esta coluna contém informações sobre terrenos de fazendas. Especificamente, cada item de dados na coluna contém os conjuntos de coordenadas que definem os perímetros dos terrenos de fazendas em um determinado município.

Um tipo de dados para todos os recursos

Você pode usar ST_Geometry quando não tiver certeza sobre qual dos outros tipos de dados utilizar. Como ST_Geometry é a raiz da hierarquia à qual os outros tipos de dados pertencem, uma coluna ST_Geometry pode armazenar qualquer ou todos os valores que podem ser armazenados em colunas às quais os outros tipos de dados são atribuídos.

Atenção: Se planeja utilizar um geocoder padrão para preencher uma coluna espacial, a coluna deverá ser do tipo ST_Point ou ST_Geometry.

Definindo uma coluna espacial para uma tabela, registrando esta coluna como uma camada e ativando um geocoder para mantê-la

Esta seção apresenta uma visão geral das etapas para definir uma coluna espacial para uma tabela, registrar esta coluna como uma camada e ativar um geocoder para mantê-la. A visão geral está seguida por detalhes de como completar cada etapa.

Para descobrir de qual autorização você precisa para registrar uma coluna da tabela como uma camada, consulte “Autorização” na página 109. Para descobrir de qual autorização você precisa para ativar um geocoder para manter esta coluna, consulte “Autorização” na página 91.

Visão geral das etapas para definir uma coluna espacial para uma tabela, registrar esta coluna como uma camada e ativar um geocoder para mantê-la:

1. Se a coluna espacial deve fazer parte de uma nova tabela, crie esta tabela.
2. Abra a janela Criar Camada Espacial.
3. Inclua uma coluna espacial em uma tabela e indique que deseja registrar esta coluna como uma camada; ou indique que deseja registrar uma coluna existente como uma camada.
4. Indique qual sistema de referência espacial deve ser usado para a camada.
5. Se a camada deverá conter dados importados ou dados que são gerados de outra coluna espacial, peça ao Spatial Extender para criar a camada.
6. Se a camada deverá conter dados derivados de dados do atributo:
 - a. Especifique qual coluna ou colunas contém estes dados do atributo.
 - b. Indique que deseja ativar um geocoder para manter a camada.
 - c. Peça ao Spatial Extender para criar a camada.

Etapas detalhadas para definir uma coluna espacial para uma tabela, registrar esta coluna como uma camada e ativar um geocoder para mantê-la:

1. Se a coluna espacial deve fazer parte de uma nova tabela, crie esta tabela:
 - Utilize uma interface de sua escolha (por exemplo, o Centro de Controle ou o Processador de Linha de Comandos) para criar a tabela.

- Se pretende utilizar um geocoder, inclua de uma a dez colunas nas quais o geocoder irá operar. Um geocoder não pode suportar mais do que dez colunas de dados como entrada.
- Inclua a coluna espacial que estará registrando como uma camada ou defina esta coluna na etapa 3.

Se quiser usar uma tabela existente, vá para a etapa seguinte.

- Abra a janela Criar Camada Espacial.
 - Da janela Centro de Controle, expanda a árvore de objetos até encontrar a pasta **Tabelas** para as tabelas no banco de dados que você utiliza para operações espaciais.
 - Clique na pasta **Tabelas**. As tabelas são exibidas no painel de conteúdos no lado direito da janela.
 - Dê um clique com o botão direito na tabela desejada e clique em **Spatial Extender** —> **Camadas Espaciais** no menu instantâneo. A janela Camadas Espaciais é aberta.
 - Da janela Camadas Espaciais, clique em **Criar**. A janela Criar Camada Espacial aparece.
- Da janela Criar Camada Espacial, inclua uma coluna espacial em uma tabela e indique que deseja registrar esta coluna como uma camada; ou indique que deseja registrar uma coluna existente como uma camada.
 - Se quiser incluir uma coluna espacial em uma tabela e definir esta coluna como uma camada:
 - No campo **Coluna da camada**, digite um nome para a coluna.
 - No campo **Tipo da coluna**, selecione ou digite o tipo de dados que deseja que a coluna tenha. Para ver uma discussão sobre os tipos de dados permitidos, consulte “Sobre tipos de dados espaciais” na página 45.
 - Se quiser definir uma coluna existente como uma camada, selecione-a no campo **Coluna da camada**.

Restrição: Não selecione uma coluna que já foi definida como uma camada.

- No campo **Nome da referência espacial**, especifique o nome do sistema de referência espacial a ser usado para a camada.
- Caso deseje que a camada contenha dados importados, ou dados que são gerados a partir de outra coluna espacial, clique em **OK** para registrá-la.
- Caso deseje que a camada contenha dados derivados de dados do atributo:
 - Especifique qual coluna ou colunas contém estes dados do atributo:
 - 1) Selecione a coluna ou colunas na caixa **Colunas disponíveis**. Você pode selecionar até dez colunas.

- 2) Clique no botão de comando >, o botão de comando >> ou em ambos para listar a colunas ou colunas selecionadas na caixa **Colunas selecionadas**.
- b. Se quiser ativar um geocoder para manter a camada:
 - 1) Selecione a caixa de seleção **Ativar geocoder automático**.
 - 2) No campo **Nome**, selecione o nome do geocoder a ser utilizado.
 - 3) No campo **Nível de precisão**, especifique, em termos de porcentagem, até que grau os registros de entrada devem ser compatíveis aos registros correspondentes nos dados de referência, para serem processados. Esta porcentagem é chamada de *precisão*. Por exemplo, suponha que o geocoder lê um registro de entrada que contém o endereço, 557 Bailey, San Jose 94120. Se a precisão for 100 e se a correspondência entre este endereço e sua contraparte nos dados de referência não for 100 por cento precisa, o geocoder irá rejeitá-la. Se a precisão for 75 e a correspondência entre o registro e sua contraparte de dados de referência for pelo menos 75 por cento precisa, o geocoder irá processá-la.
 - 4) Se o geocoder foi concedido por um fornecedor, utilize a caixa **Propriedades** para especificar todos os parâmetros de geocodificação concedidos pelo fornecedor que você deseja utilizar.
 - c. Clique em **OK** para registrar a coluna selecionada como uma camada e, se tiver solicitado, para ativar o geocoder para manter a coluna.

Restrições

Você não pode alterar as colunas de entrada no geocoder. Entretanto, é possível alterar as propriedades do geocoder. Por exemplo, você pode ativar ou desativar o geocode incremental.

Nota: A janela Excluir Camada não é utilizada para excluir camadas, mas para remover seus registros; ou seja, para remover do catálogo do sistema DB2 Spatial Extender todas as informações sobre elas. Ao remover o registro de uma camada, a tabela ou a coluna de exibição utilizada para a camada continua a existir.

Registrando uma coluna de exibição como uma camada

Para descobrir qual autorização você necessita para registrar uma coluna de exibição como uma camada, consulte “Autorização” na página 109.

Para registrar uma coluna de exibição como uma camada:

1. Abra a janela Criar Camada Espacial.
 - a. Da janela Centro de Controle, expanda a árvore de objetos até encontrar a pasta **Views** para as exibições no banco de dados que você utiliza para operações espaciais.

- b. Clique na pasta **Views**. As exibições são apresentadas no painel de conteúdo à direita da janela.
 - c. Dê um clique com o botão direito na exibição desejada e clique em **Spatial Extender** —> **Camadas Espaciais** no menu instantâneo. A janela Camadas Espaciais é aberta.
 - d. Da janela Camadas Espaciais, clique em **Criar** . A janela Criar Camada Espacial aparece.
2. Utilize a caixa **Coluna da camada** para especificar a coluna que você deseja registrar como uma camada.
 3. No campo **Camada espacial subjacente**, especifique o nome da coluna da tabela na qual está baseada a coluna da exibição selecionada. Esta coluna da tabela já deve estar registrada como uma camada.
 4. Clique em **OK** para registrar a coluna da exibição especificada como uma camada.

Nota: Antes de eliminar uma tabela que possui uma coluna registrada como uma camada, será necessário executar uma ou mais ações:

- Remover o registro da camada.
- Se a camada possuir um índice espacial, exclua esse índice.
- Se uma coluna de exibição foi definida sobre a coluna da tabela e se essa coluna de exibição foi registrada como uma camada, remova também essa camada.

Capítulo 5. Preenchendo colunas espaciais

Após registrar colunas espaciais como camadas, você está pronto para supri-las com dados espaciais. Como observado em “De onde vêm os dados espaciais” na página 6, há três maneiras de fornecer estes dados: use uma função, chamada *geocoder*, para derivá-lo dos dados de atributo; use outras funções para derivá-lo de outros dados espaciais; ou importe-o de arquivos. Este capítulo:

- Fala sobre geocodificação e explica como usar o Centro de Controle para geocodificar dados de atributo no modo batch.
- Discute a importação e exportação de dados e explica como usar o Centro de Controle para importar dados para o seu GIS e exportá-los a partir do GIS

Para informações sobre as funções que podem derivar novos dados espaciais a partir de dados espaciais existentes, consulte “Funções que geram novas figuras geométricas a partir de existentes” na página 184.

Utilizando os geocoders

Esta seção descreve o processo de geocodificação e explica como executar um geocoder no modo batch a partir do Centro de Controle.

Sobre geocodificação

Conforme observado anteriormente, o DB2 Spatial Extender utiliza uma função, chamada *geocoder*, para executar duas atividades: conversão de dados de endereço em dados espaciais e colocação desses dados espaciais em colunas da tabela. Esta seção distingue as diferenças básicas entre os geocoders e suas fontes. Ela também descreve os dois modos nos quais um geocoder pode operar e introduz fatores a considerar quando você planeja usar um geocoder.

Com o Spatial Extender, você pode:

- Usar o geocoder padrão que é fornecido com o Spatial Extender.
- Ativar geocoders que são desenvolvidos por outros fornecedores.
- Ativar seus próprios geocoders.

O geocoder padrão geocodifica endereços nos Estados Unidos e pode convertê-los em dados *ST_Point*. Se você precisa armazenar dados de outros tipos de dados espaciais, você pode se conectar a um geocoder para gerar tais dados. Se você precisa de dados espaciais que representam locais fora dos

Estados Unidos, ou locais que não possuam endereços — por exemplo, área de cultivo que variam por conteúdo de solo — você pode se conectar a um geocoder para encontrar o que precisa.

Antes que um geocoder plug-in possa ser usado, ele deve ser registrado. Usuários e fornecedores podem registrá-lo com o procedimento armazenado `db2gse.gse_register_gc`. Ele não pode ser registrado a partir do Centro de Controle. Para obter mais informações sobre `db2gse.gse_register_gc`, consulte “`db2gse.gse_register_gc`” na página 107. Para obter informações gerais sobre o uso dos procedimentos armazenados Spatial Extender, consulte “Capítulo 9. Procedimentos armazenados” na página 83.

Um geocoder opera em dois modos:

- No *modo batch*, ele tenta, em uma operação única, converter todos os dados fontes existentes para uma coluna espacial em dados espaciais e preencher a coluna com aqueles dados. Você pode iniciar esta operação a partir da janela Executar Geocoder. Alternativamente, você pode iniciá-lo em um programa de aplicação, codificando o programa para chamar o procedimento armazenado `db2gse.gse_run_gc`.
- No *modo incremental*, um geocoder converte dados quando ele é incluído ou atualizado em uma tabela, colocando os valores espaciais resultantes em uma coluna na ordem para manter a atual coluna. Ele é ativado por disparadores de inclusão e atualização que você pode solicitar a partir da janela Criar Camada Espacial. Alternativamente, você pode solicitar o mesmo em um programa de aplicação, codificando o programa para chamar o procedimento armazenado `db2gse.gse_enable_autogc`.

A geocodificação incremental é referenciada também como *geocodificação automática*.

Ao planejar o uso de um geocoder, você deve considerar os seguintes fatores:

1. Quando você usa o Centro de Controle, você geralmente usa a janela Criar Camadas Espaciais antes de utilizar a janela Executar o Geocoder. Isto significa que você pode ter definido disparadores Spatial Extender para a geocodificação incremental antes de iniciar a geocodificação em batch. Portanto, é possível para a geocodificação incremental preceder a geocodificação em batch. Processando todos os dados fontes no modo batch, o geocoder irá geocodificar os mesmos dados que ele operou no modo incremental. Esta redundância não causará duplicações (quando os dados espaciais são produzidos duas vezes, a segunda produção dos dados irá sobrepor a primeira). De qualquer modo, ele pode degradar o desempenho. Uma maneira de evitar isto é postergar a definição dos disparadores até que a geocodificação em batch esteja feita.
2. Se os disparadores estão no lugar quando você estiver pronto para geocodificar no modo batch é aconselhável desativá-los até que a geocodificação em batch tenha encerrado. Você pode desativá-los a partir

da janela Executar Geocoder ou em um programa de aplicação, codificando o programa para chamar o procedimento armazenado `db2gse.gse_disable_autogc`. Se você usa a janela Executar Geocoder, o Spatial Extender reativa-os automaticamente quando a geocodificação estiver encerrada. Se você usa o procedimento armazenado `db2gse.gse_disable_autogc`, você pode reativá-los chamando o procedimento armazenado `db2gse.gse_enable_autogc`.

3. Se deseja executar um geocoder no modo batch para preencher uma coluna espacial que possui um índice, desative ou elimine o índice primeiro. Caso contrário, se o índice permanece operável enquanto o geocoder é executado, o desempenho será severamente prejudicado. Se estiver usando o Centro de Controle, você pode desativar o índice a partir da janela Executar Geocoder. O Spatial Extender reativa o índice automaticamente quando a geocodificação estiver encerrada. Se estiver usando um programa de aplicação, você pode eliminar o índice com a instrução SQL DROP. Caso faça isso, assegure-se de tomar nota dos parâmetros do índice, assim você pode recriá-lo após o encerramento da geocodificação.
4. Quando o geocoder lê um registro dos dados fonte, ele tenta fazer coincidir o registro com uma contra-parte nos dados de referência. A coincidência deve ser exata para um certo grau (chamado uma *precisão*) para o geocoder processar o registro. Por exemplo, uma precisão de 85 significa que a coincidência entre um registro fonte e a sua contrapartida nos dados de referência deve ter pelo menos 85 por cento de precisão a fim de que o registro fonte seja processado.

Você determina qual precisão deve ser. Esteja ciente de que talvez necessite ajustá-lo. Por exemplo, supõem-se que a precisão seja 100. Se muitos registros fonte contêm endereços que são mais recentes do que os dados de referência, a coincidência de 100 por cento entre estes registros e os dados de referência será impossível. Como resultado, o geocoder rejeitará estes registros. No total, se um geocoder produz dados espaciais que parecem insuficientes ou largamente imprecisos, você pode estar apto a resolver este problema alterando a precisão e executando o geocoder novamente.

Há diversas maneiras de controlar o número ou a série de registros que um geocoder processa antes de uma consolidação ser emitida:

Método 1

Um geocoder pode geocodificar dados de endereço em um número específico de registros antes de cada consolidação. Esse método permite gerenciar, de forma precisa, o tamanho de unidades de trabalho. Entretanto, este método consome consideravelmente mais sobrecarga que outros métodos abordados aqui.

Para iniciar o Método 1, especifique o número de registros a serem processados antes de cada consolidação. Se estiver utilizando o Centro de Controle, defina esse número com o botão de rotação **Consolidar escopo** na janela Executar Geocoder. Se estiver gravando um programa aplicativo, atribua esse número ao parâmetro `commitScope` do procedimento armazenado `db2gse.gse_run_gc`.

Método 2

Um geocoder pode geocodificar dados de endereço em todos os registros de uma tabela antes que uma consolidação seja emitida. Com esse método, o geocoder processa os registros de maneira semelhante à operação em batch, consumindo muito menos sobrecarga por registro que o Método 1. Entretanto, você tem um controle menor sobre o tamanho da unidade de trabalho que no Método 1. Conseqüentemente, não é possível controlar quantos bloqueios serão recebidos ou quantas entradas de registros serão feitas conforme o geocoder opera. Além disso, se o geocoder encontrar um erro que precise de uma reversão, será necessário definir o geocoder para que execute todos os registros novamente. O custo resultante com recursos pode ser caro se a tabela for extremamente grande e o erro e a reversão ocorrerem após a maioria dos registros ter sido processada.

Para iniciar o Método 2 do Centro de Controle, defina o botão de rotação **Consolidar escopo** na janela Executar Geocoder para zero. Se estiver gravando um programa aplicativo, defina o parâmetro `commitScope` do procedimento armazenado `db2gse.gse_run_gc` para zero.

Método 3

Um geocoder pode geocodificar dados de endereço em um subconjunto de registros de uma tabela antes de cada consolidação. Ele pode, então, geocodificar dados de endereço em um segundo subconjunto e, se for necessário, um terceiro, um quarto e assim por diante. O geocoder processa cada subconjunto de maneira semelhante à operação em batch, consumindo menos sobrecarga por registro que no Método 1. Porém, como no Método 2, o Método 3 não permite o controle direto sobre o tamanho da unidade de trabalho. Além disso, ele requer a sobrecarga de definição e execução do geocoder várias vezes para cada subconjunto de registros.

Para definir um subconjunto de registros a partir do Centro de Controle:

- Na janela Executar Geocoder, utilize a caixa **Cláusula Where** para codificar uma cláusula `SELECT WHERE` que especifique uma série de registros no assunto.
- Defina o botão de rotação **Consolidar escopo** na janela Executar Geocoder para zero.

Para definir um subconjunto de registros em um programa aplicativo, codifique o procedimento armazenado `db2gse.gse_run_gc` da seguinte forma:

- Utilize o parâmetro `whereClause` para especificar a série de registros no subconjunto.
- Defina o parâmetro `commitScope` como zero.

Executando o geocoder no modo batch

Esta seção oferece uma visão geral das etapas para executar um geocoder no modo batch a partir do Centro de Controle. A visão geral está seguida por detalhes de como completar cada etapa.

Para encontrar a autorização que você precisa para executar um geocoder no modo batch, consulte “Autorização” na página 116.

Visão geral das etapas para executar um geocoder no modo batch:

1. Abra a janela Executar Geocoder.
2. Indique qual geocoder você deseja utilizar.
3. Desative os objetos que possam impedir o desempenho do geocoder.
4. Especifique quantos registros geocodificar antes que o DB2 emita um commit.
5. Indique como deseja que o geocoder opere.
6. instrua o Spatial Extender a executar o geocoder.

Etapas detalhadas para executar um geocoder no modo batch:

1. Abra a janela Executar Geocoder.
 - a. A partir da janela Centro de Controle, expanda a árvore de objetos até encontrar a pasta **Tabelas** em seu banco de dados ativado espacialmente.
 - b. Clique na pasta **Tabelas**. As tabelas são exibidas no painel de conteúdos no lado direito da janela.
 - c. Clique com o botão direito na tabela que deseja no painel de conteúdo e clique em **Camadas Espaciais** no menu instantâneo. A janela Camadas Espaciais é aberta.
 - d. A partir da janela Camadas Espaciais:
 - 1) Selecione a camada que está definida na coluna que você deseja preencher.
 - 2) Clique no botão **Executar Geocoder**. A janela Executar Geocoder é aberta.
2. Se você deseja usar o geocoder padrão, deixe a caixa **Nome**, que exibe o nome deste padrão, como está. Caso contrário use a caixa para selecionar o geocoder que deseja.
3. Desative os objetos que possam impedir o desempenho do geocoder:

- Se a coluna que deseja preencher possui um índice, selecione a caixa de seleção **Desativar temporariamente os índices espaciais durante o processo de geocodificação**.
 - Se os disparadores foram definidos para ativar a geocodificação incremental para esta coluna, selecione a caixa de seleção **Desativar temporariamente os disparadores espaciais durante o processo de geocodificação**.
O índice e disparadores serão automaticamente reativados ao clicar em **OK** na janela Executar Geocoder.
4. Use o botão de rolagem **Escopo do Commit** para especificar quantos registros geocodificar antes da emissão de um commit do DB2. Por exemplo, se você deseja que o DB2 faça o commit a cada 100 registros, especifique o número 100.
- Dica:** Caso você queira o DB2 para emitir somente um commit após todos os registros serem processados, especifique zero.
5. Use os campos na caixa de grupo **Parâmetros do Geocoder** para indicar como deseja que o geocoder opere:
- Use o botão de rolagem **Nível de Precisão**, para especificar, em termos de porcentagem, quão precisa a coincidência entre registros fonte e suas contra-partes dos dados de referência devem ser. Para obter maiores informações sobre precisão, consulte “Sobre geocodificação” na página 53 .
 - Se estiver usando um geocoder fornecido por fornecedor, e deseja usar as propriedades que ela suporta, use a caixa **Propriedades** para definir estas propriedades.
 - Caso queira geocodificar apenas um subconjunto de linhas na tabela selecionada, use a caixa **cláusula WHERE** para codificar uma cláusula SELECT WHERE que irá especificar o critério para as linhas que deseja. Esta cláusula pode referenciar quaisquer colunas na tabela.
Digite somente os critérios. Omita a palavra-chave WHERE. Por exemplo, se a tabela possui uma coluna STATE, e você deseja fazer o geocode apenas para as linhas que contenham o valor MA nesta coluna, digite:
STATE='MA'
6. Clique em **OK** para executar o geocoder.

Importando e exportando dados

Esta seção descreve os processos de importação e exportação de dados, e explica como usar o Centro de Controle para:

- Importar dados de um arquivo de permuta de dados para uma tabela nova ou existente

- Importar dados de um arquivo de permuta de dados para uma tabela existente
- Exportar dados de uma tabela para um arquivo de permuta de dados

Sobre importação e exportação

Esta seção lista as razões para a importação e exportação de dados espaciais. Ele também discute os arquivos de permuta de dados que servem como interfaces entre fontes de exportação e destinos da importação.

Você pode usar o Spatial Extender para importar dados espaciais de, e exportá-los para, arquivos de permuta de dados. Considere estes cenários:

- Seu GIS contém dados espaciais que representam seus escritórios, clientes e outros assuntos comerciais. Você deseja suplementar estes dados com dados espaciais que representam seu ambiente cultural da organização — cidades, ruas, pontos de interesse, e assim por diante. Os dados que você deseja estão disponíveis a partir de um mapa do fornecedor. Você pode usar o Spatial Extender para importá-lo de um arquivo de permuta de dados que o fornecedor fornece.
- Você deseja migrar dados espaciais de um sistema Oracle para o seu GIS Spatial Extender. Você utiliza um utilitário Oracle para carregar os dados em um arquivo de permuta de dados. Você então usa o Spatial Extender para importar os dados deste arquivo para o banco de dados que você ativou nas operações espaciais.
- Você deseja usar um navegador GIS para mostrar as apresentações da informação espacial para os clientes. O navegador precisa apenas de arquivos para trabalhar; ele não precisa estar conectado a um banco de dados. Você pode usar o Spatial Extender para exportar os dados para um arquivo de permuta de dados, e então usar um utilitário de navegação para carregar os dados no navegador.

O Centro de Controle suporta dois tipos de arquivos de permuta de dados para o Spatial Extender: arquivos de formato e arquivos de transferência ESRI_SDE. Arquivos de formato são freqüentemente utilizados para a importação de dados que se originam nos sistemas de arquivo e para a exportação de dados para arquivos que serão carregados nos sistemas de arquivo. Arquivos de transferência ESRI_SDE são usados freqüentemente para a importação de dados que se originam nos bancos de dados ESRI.

Importação de dados para uma tabela nova ou existente a partir do nível do banco de dados

Esta seção oferece uma visão geral das etapas de importação de dados de um arquivo de formato ou de transferência ESRI_SDE para uma tabela nova ou já existente utilizando a pasta **Bancos de Dados** do Centro de Controle. A visão geral está seguida por detalhes de como completar cada etapa.

Ao importar um conjunto de representações de formato do ESRI, você recebe, pelo menos, dois arquivos. Todos os arquivos possuem o mesmo nome, mas extensões diferentes. Por exemplo, as extensões dos dois arquivos que você sempre recebe são .shp e .shx.

Para receber os arquivos de um conjunto de representações de formato, digite o nome que os arquivos têm em comum no campo **Nome do arquivo** da janela Importar Dados Espaciais. Não digite uma extensão. Dessa forma, você pode certificar-se de que todos os arquivos de que precisa - o arquivo .shp, o arquivo .shx e quaisquer outros que possam estar incluídos - serão importados.

Por exemplo, suponha que um conjunto de representações de formato do ESRI esteja armazenado em arquivos chamados Lakes.shp e Lakes.shx. Ao importar estas representações, você deve digitar apenas Lakes no campo Nome do arquivo.

Os arquivos de transferência SDE possuem nomes, mas não extensões. Portanto, ao importar um arquivo de transferência SDE, você digita seu nome sem a extensão, no campo Nome do arquivo. De forma semelhante, no campo **Arquivo de exceção** da janela Importar Dados Espaciais, não digite a extensão do arquivo que deseja especificar. Digite apenas o nome do arquivo.

Para localizar qual autorização é requerida para importar dados de formato, consulte "Autorização" na página 104. Para localizar qual autorização é requerida para importar dados ESRI_SDE, consulte "Autorização" na página 102.

Visão Geral das etapas para importar os dados para uma tabela nova ou já existente:

1. Abra a janela Importar Dados Espaciais
2. Especifique o caminho, o nome e o formato do arquivo que contém os dados a serem importados.
3. Especifique quantos registros importar antes de cada commit.
4. Se deseja importar dados espaciais para uma tabela a ser criada, forneça um nome para esta tabela e um nome para a coluna para a qual os dados são planejados. Se a importação dos dados espaciais é para uma tabela existente, indique para qual coluna os dados estão planejados.
5. Especifique qual sistema de referência espacial deve ser associado aos dados.
6. Designe um arquivo para coletar os registros que falhar na importação.
7. Instrua o Spatial Extender para importar os dados e, se você definiu uma tabela a partir desta janela, para criar a tabela e registrar a coluna para a qual os dados são planejados como uma camada.

Detalhes das etapas para importar os dados para uma tabela nova ou já existente:

1. Abra a janela Importar Dados Espaciais
 - a. A partir da janela Centro de Controle, expanda a árvore objeto até localizar a pasta **Bancos de Dados** sob o servidor que você está executando o Spatial Extender.
 - b. Clique na pasta **Bancos de Dados**. Os bancos de dados são exibidos no painel de conteúdos no lado direito da janela.
 - c. Clique com o botão direito no banco de dados para o qual deseja importar dados e clique em **Spatial Extender** → **Importar Dados Espaciais** no menu instantâneo. A janela Importar Dados Espaciais é aberta.
2. Especifique o caminho, o nome e o formato do arquivo que contém os dados a serem importados:
 - a. Use o campo **Nome de arquivo** para especificar o caminho e nome.
 - b. Use a caixa **Formato de arquivo** para especificar o formato. O formato pode ser:

Formato

Este é o padrão.

ESRI_SDE

Se você especificou este formato, o campo **Nome de referência Espacial** usa o padrão para o nome do sistema de referência espacial que está associado a este formato.

3. Use o campo **Escopo do Commit** para especificar o número de registros que você deseja importar antes de cada commit. Por exemplo, para solicitar o commit do DB2 a cada 100 registros, especifique o número 100.

Dica: Caso você queira o DB2 para emitir somente um commit após todos os registros serem processados, especifique zero.

4. Especifique a tabela e coluna para a qual os dados são planejados.
 - a. Use a caixa **Esquema de Camada** para especificar o esquema para a tabela na qual os dados serão importados.
 - b. Especifique a tabela e a coluna:
 - Caso a tabela ainda não exista:
 - 1) No campo **Tabela de Camadas**, digite um nome para a tabela.
 - 2) No campo **Coluna de Camadas**, digite um nome para a coluna que deverá conter os dados importados. O Spatial Extender registrará automaticamente esta coluna como uma camada.
 - Caso a tabela já exista:

- 1) No campo **Tabela de Camadas**, especifique a tabela. Ela já deve conter a coluna para a qual você deseja importar os dados. Em complemento, esta coluna deve já estar registrada como uma camada.
- 2) No campo **Coluna de Camadas**, especifique o nome da coluna para a qual os dados importados está planejado.
5. No campo **Nome de referência espacial**, digite ou selecione o sistema de referência espacial que deve ser associado a estes dados. (Se os dados devem vir do arquivo de transferência ESRI_SDE, o nome do sistema de referência espacial é exibido no campo automaticamente.)
6. No campo **Arquivo de Exceção**, especifique o caminho e nome para um novo arquivo no qual os registros que falharem durante a importação sejam coletados. Mais tarde, você pode alterar estes registros e importá-los deste arquivo.
O Spatial Extender criará este arquivo; não especifique um que já exista.
7. Clique em **OK** para importar os dados. Também, se você forneceu um nome para uma tabela que não existe ainda, esta tabela será criada e a coluna para a qual os dados são planejados será registrada como uma camada. Complementarmente, o arquivo de exceção que você especificou será criado.

Importação de dados para uma tabela existente a partir do nível da tabela

Esta seção oferece uma visão geral das etapas de importação de dados de um arquivo de formato ou de um arquivo de transferência ESRI_SDE para uma tabela existente utilizando a pasta **Tabelas** do Centro de Controle. A visão geral está seguida por detalhes de como completar cada etapa.

Para localizar qual autorização é requerida para importar dados de formato, consulte “Autorização” na página 104. Para localizar qual autorização é requerida para importar dados ESRI_SDE, consulte “Autorização” na página 102.

Visão Geral das etapas para importar os dados para uma tabela existente:

1. Abra a janela Importar Dados Espaciais
2. Especifique o caminho e o nome do arquivo que contém os dados a serem importados.
3. Especifique quantos registros importar antes de cada commit.
4. Especifique a coluna que conterà os dados espaciais que você está importando.
5. Especifique qual sistema de referência espacial deve ser associado a estes dados.
6. Designe um arquivo para coletar registros que falharem na importação.

7. Instrua ao Spatial Extender para importar os dados e, se você especificou uma coluna que não tenha sido criada ainda, para criar esta coluna e registrá-la como uma camada.

Detalhes das etapas para importar os dados para uma tabela existente:

1. Abra a janela Importar Dados Espaciais
 - a. A partir da janela Centro de Controle, expanda a árvore objeto até encontrar a pasta **Tabelas** para o banco de dados para o qual você deseja importar os dados.
 - b. Clique na pasta **Tabelas**. As tabelas são exibidas no painel de conteúdos no lado direito da janela.
 - c. Clique com o botão direito na tabela que você está importando os dados e clique em **Spatial Extender** —> **Importar Dados Espaciais** no menu instantâneo. A janela Importar Dados Espaciais é aberta.
2. Na caixa **Nome de Arquivo**, especifique o caminho e o nome do arquivo que contém os dados a serem importados.
3. Use a caixa **Escopo do Commit** para especificar o número de registros que você deseja importar antes de cada commit. Por exemplo, para solicitar o commit do DB2 a cada 100 registros, especifique o número 100.

Dica: Caso você queira o DB2 para emitir somente um commit após todos os registros serem processados, especifique zero.

4. Especifique a coluna que conterá os dados espaciais que você está importando.
 - Se a coluna não existe ainda na tabela, use a caixa **Coluna de Camada** para digitar um nome para a coluna.
 - Se a coluna já existe, use a caixa **Coluna de Camada** para selecionar ou digitar o nome da coluna.
5. Use a caixa **Nome de referência espacial** para especificar qual sistema de referência espacial deve ser associado aos dados importados.
 - Se estiver incluindo uma coluna em uma tabela, digite ou selecione o nome do sistema de referência espacial.
 - Se os dados importados estiverem planejados para uma coluna existente, deixe a caixa **Nome de referência espacial** como está. Ela exibe o nome do sistema de referência espacial padrão.
6. No campo **Arquivo de Exceção**, especifique o caminho e nome para um novo arquivo no qual os registros que falharem durante a importação sejam coletados. Mais tarde, você pode alterar estes registros e importá-los deste arquivo.

O Spatial Extender criará este arquivo; não especifique um que já exista.

7. Clique em **OK** para importar os dados. Também, se você especificou uma coluna que não existe ainda, esta coluna será criada e registrada como uma camada. Complementarmente, o arquivo de exceção que você especificou será criado.

Exportando dados para um arquivo de formato

Esta seção oferece uma visão geral das etapas para exportar dados espaciais para um arquivo de formato. A visão geral está seguida por detalhes de como completar cada etapa.

Para encontrar que autorização é requerida para executar estas etapas, consulte “Autorização” na página 100.

Visão geral das etapas para exportar dados a partir de um arquivo de formato:

1. Abra a janela Exportar Dados Espaciais
2. Especifique a coluna que contém os dados espaciais a serem exportados.
3. Caso deseje exportar um subconjunto de linhas de dados, identifique este subconjunto para o Spatial Extender.
4. Especifique o caminho e o nome do arquivo para o qual está exportando os dados.
5. Instrua o Spatial Extender para exportar os dados.

Detalhe das etapas para exportar dados para um arquivo de formato:

1. Abra a janela Exportar Dados Espaciais
 - a. A partir da janela Centro de Controle, expanda a árvore objeto até encontrar a pasta **Tabelas** ou **Exibições** no banco de dados que contém os dados espaciais:
 - b. Clique na pasta **Tabelas** ou **Exibições**. Tabelas ou exibições são exibidas no painel de conteúdos no lado direito da janela.
 - c. Clique com o botão direito na tabela ou exibição que contém os dados a serem exportados e clique em **Spatial Extender** —> **Exportar Dados Espaciais** no menu instantâneo. A janela Exportar Dados Espaciais é aberta.
2. No campo **Coluna de Camadas**, especifique o nome da coluna que contém os dados para serem exportados.
3. Caso queira exportar um subconjunto de linhas da tabela, use a caixa **cláusula WHERE** para digitar uma cláusula WHERE que especifica o critério para as linhas que deseja. Nesta cláusula, você pode referenciar apenas colunas na tabela ou exibição da qual esteja exportando dados.

Digite somente os critérios. Omita a palavra-chave `WHERE`. Por exemplo, se a tabela ou exibição possui uma coluna `STATE`, e você deseja fazer o geocode apenas para as linhas que contenham o valor `MA` nesta coluna, digite:

```
STATE='MA'
```

4. No campo **Nome de arquivo**, especifique o caminho e o nome do arquivo para o qual esteja exportando os dados.
5. Clique em **OK** para exportar os dados.

Capítulo 6. Criando índices espaciais

Este capítulo explica como usar o Centro de Controle para criar um índice para seus dados espaciais.

Após preencher as colunas espaciais com seus dados, você está pronto para criar um índice espacial. Estruturas de indexação comuns, como B tree, realizam ordenações lineares monodimensionais em dados da tabela. Os dados da tabela que foram habilitados para operações espaciais não são armazenados como uma única entrada, mas são bidimensionais. Por exemplo, as figuras geométricas espaciais como polígonos consistem em vários valores de coordenadas em uma coluna ou camada espacial. Como um índice B tree não pode manipular tipos de dados espaciais, o Spatial Extender criou uma tecnologia de indexação de propriedade conhecida como *índice de grade*. O índice de grade baseia-se no índice B tree, que foi aprimorado para manipular dados bidimensionais e realizar indexações em colunas espaciais. O índice de grade suporta três camadas e foi projetado para oferecer um bom desempenho em uma faixa grande de objetos, tamanhos e distribuições de dados. Para obter mais informações sobre índices espaciais, consulte “Capítulo 12. Índices espaciais” na página 147.

Para descobrir qual é a autorização necessária para criação de um índice espacial, consulte “Autorização” na página 95.

Utilizando o Centro de Controle para criar um índice espacial

Para criar um índice espacial utilizando o Centro de Controle:

1. Na árvore de objetos, selecione a pasta **Tabelas**. Todas as tabelas existentes são exibidas no painel de conteúdo.
2. A partir do painel de conteúdo, clique com o botão direito sobre a tabela para a qual deseja criar um índice, e clique em **Spatial Extender** → **Índices Espaciais** no menu instantâneo. A janela Índices Espaciais aparece.
3. A partir da janela Índices Espaciais, clique em **Criar**. A janela Criar Índice Espacial aparece.
4. No campo **Nome**, digite o nome do novo índice espacial a ser criado.

Nota: O Spatial Extender não permite escolher um nome de esquema para um índice. O Spatial Extender inclui automaticamente o nome do esquema e cria um nome completo.

5. No campo **Coluna da camada**, selecione a camada para a qual está criando um índice. Uma camada pode ter apenas um índice espacial.

Uma camada é uma coluna espacial definida ou registrada para o Spatial Extender.

6. Nos campos **Tamanho da grade**, digite o valor do tamanho da grade que deseja atribuir a cada campo.

Os níveis da grade, **Mais fino**, **Médio** e **Mais grosso**, são fornecidos aumentando-se o tamanho da célula. Conseqüentemente, o segundo nível deve ter um tamanho de célula maior do que o primeiro e o terceiro maior que o segundo.

Para definir o tamanho de uma grade no campo **Mais fino**, no campo **Médio** ou no campo **Mais grosso**, pressione as teclas de seta para cima e para baixo no teclado da estação de trabalho. Pressione a seta para cima para aumentar o tamanho por um décimo de um grau; pressione a seta para baixo para reduzir o tamanho por um décimo de um grau.

Não é necessário especificar os três tamanhos da grade.

Determinando os tamanhos de células da grade

A determinação do tamanho correto da grade é feito através de um processo de tentativa e erro. Recomenda-se que o tamanho da grade seja definido em relação ao tamanho aproximado do objeto sendo indexado. Os tamanhos da grade definidos muito pequenos ou muito grandes podem prejudicar o desempenho. Tamanhos pequenos demais afetam a proporção chave/objeto durante uma pesquisa do índice. Neste caso, um número excessivo de chaves é criado e um número grande de candidatos é retornado. Para tamanhos de grades definidos muito grandes, a pesquisa de índice inicial retorna um número pequeno de candidatos, porém o desempenho pode ficar mais lento durante a varredura final da tabela.

Para saber mais sobre a escolha de tamanhos de células de grade e o número de níveis de grade, consulte “Selecionando o tamanho da célula de grade” na página 154.

Capítulo 7. Recuperando e analisando informações espaciais

Após construir os índices espaciais as tabelas espaciais estão prontas para uso. Este capítulo discute temas relacionados para recuperar e analisar dados espaciais. Ele contém uma visão geral de vários métodos de recuperação e fornece exemplo de consultas de tabela que usam funções espaciais.

Métodos de execução de análise espacial

Você pode executar análises espaciais utilizando SQL e funções espaciais com qualquer um dos seguintes ambientes de programação:

- Um geobrowser (por exemplo, ESRI's ArcExplorer).
Para mais informações sobre o uso do ArcExplorer, consulte o *Using ArcExplorer*, que está disponível no site Web ESRI em <http://www.esri.com>.
- Instruções SQL interativas.
Você pode digitar instruções SQL interativas a partir do Centro de Comandos do DB2, da Janela de Comandos do DB2 ou do processador de linha de comandos.
- Aplicações desenvolvidas pelo usuário (por exemplo, ODBC, JDBC, e SQL incorporadas).

Construindo uma consulta espacial

Esta seção discute a construção de consultas espaciais que usam funções e predicados espaciais.

Funções espaciais e SQL

O Spatial Extender inclui funções que executam várias operações nos dados espaciais. Os exemplos nesta seção mostram como usar as funções espaciais para construir suas próprias consultas espaciais.

A Tabela 3 fornece uma lista das funções espaciais e os tipos de operações que elas podem executar.

Tabela 3. Operações e funções espaciais

| Tipo de função | Exemplo de operação |
|----------------|--|
| Cálculo | Cálculo da distância entre 2 pontos |
| Comparação | Localiza todos os clientes localizados dentro de uma zona de inundação |

Tabela 3. Operações e funções espaciais (continuação)

| Tipo de função | Exemplo de operação |
|------------------|--|
| Permuta de dados | Converte dados nos formatos suportados |
| Transformação | Inclui um raio de cinco milhas em um ponto |

Para obter mais informações sobre as funções espaciais, consulte “Capítulo 13. Figuras geométricas e funções espaciais associadas” na página 157 e “Capítulo 14. Funções espaciais para consultas SQL” na página 195.

Exemplo 1: Comparação

A seguinte consulta localiza a distância média do cliente a cada estabelecimento. As funções espaciais usadas neste exemplo são ST_Distance e ST_Within.

```
SELECT s.id, AVG(db2gse.ST_Distance(c.location,s.location))
FROM customers c, stores s
WHERE db2gse.ST_Within(c.location,s.zone)=1
GROUP BY s.id
```

Exemplo 2: Permuta de dados

A seguinte consulta procura as localizações de clientes que vivem na Área da Baía de São Francisco (San Francisco Bay Area). As funções espaciais utilizadas neste exemplo são ST_AsText (permuta de dados) e ST_Within. ST_AsText converte os dados espaciais na coluna c.location no formato OGC TEXT.

```
SELECT db2gse.ST_AsText(c.location, cordref(1))
FROM customers c
WHERE db2gse.ST_Within(c.location, :BayArea)=1
```

Exemplo 3: Cálculo

A seguinte consulta procura todas as ruas maiores que 10.5 milhas. A função espacial utilizada neste exemplo é a ST_Length.

```
SELECT s.name, s.id
FROM street s
WHERE db2gse.ST_Length(s.path) > 10.5
```

Exemplo 4: Transformação

Esta consulta localiza os clientes que vivem dentro de uma área de inundação ou dentro de 2 milhas do limite da zona de inundação. As funções espaciais utilizadas neste exemplo são ST_Buffer (transformação) e ST_Within. A variável :floodzone é uma variável de host em um programa SQL implementado em C/C++.

```
SELECT c.name, c.phoneNo, c.address
FROM customers c
WHERE db2gse.ST_Within(c.location, ST_Buffer(:floodzone, 2))=1
```

Predicados espaciais e SQL

Um grupo especializado de funções espaciais que são chamados predicados espaciais podem aprimorar o desempenho da consulta. Predicados espaciais, tais como `ST_Overlaps`, que compara dois polígonos para ver se eles se sobrepõem, pode ser uma execução custosa por problemas de requisitos de memória e tempo. Portanto, as técnicas de otimizações para minimizar o custo de execução são muito importantes. O otimizador de consultas do DB2 utiliza o índice espacial para aprimorar o desempenho da consulta quando você utiliza os predicados espaciais de acordo com as regras descritas mais adiante nesta seção. Para mais informações sobre os predicados espaciais, consulte “Funções do predicado” na página 172. Os predicados espaciais usados para explorar o índice espacial são:

- `ST_Contains`
- `ST_Crosses`
- `ST_Disjoint`
- `ST_Distance`
- `ST_Envelope`
- `ST_Equals`
- `ST_Intersects`
- `ST_Overlaps`
- `ST_Touches`
- `ST_Within`

Para uma lista completa de todas as funções e predicados espaciais, consulte “Capítulo 14. Funções espaciais para consultas SQL” na página 195.

Regras para exploração de índice

As seguintes regras se aplicam caso deseje otimizar consultas espaciais utilizando predicados espaciais:

- O predicado deve ser utilizado na cláusula `WHERE`.
- O predicado deve estar do lado esquerdo da comparação. Por exemplo:
`WHERE db2gse.ST_Within(c.location,:BayArea)=1`
- Comparações de igualdade devem usar o inteiro constante 1.
`WHERE db2gse.ST_Within(c.location,:BayArea)=1`
- Deve ser uma coluna espacial usada no predicado como a pesquisa de destino, e deve ser um índice espacial criado naquela coluna.

Exemplos da exploração do índice

A Tabela 4 na página 72 mostra as maneiras corretas e incorretas de criar consultas espaciais para explorar o índice espacial.

Tabela 4. Regras para exploração de índice

| Consulta espaciais | Regras violadas |
|--|---|
| <pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=1</pre> | Nenhuma condição é violada neste exemplo. |
| <pre>SELECT * FROM customers c WHERE db2gse.ST_Distance(c.location,:SanJose)<10</pre> | Nenhuma condição é violada neste exemplo. |
| <pre>SELECT * FROM customers c WHERE db2gse.ST_Length(c.location)>10</pre> | ST_Length não é um predicado espacial. |
| <pre>SELECT * FROM customers c WHERE 1=db2gse.ST_Within(c.location,:BayArea)</pre> | O predicado deve estar do lado esquerdo da comparação. |
| <pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=2</pre> | Comparações de igualdade devem usar o inteiro constante 1. |
| <pre>SELECT * FROM customers c WHERE db2gse.ST_Within(:SanJose,:BayArea)=1</pre> | Deve ser uma coluna espacial usada no predicado como a pesquisa de destino, naquela coluna. (SanJose e BayArea não são colunas espaciais e portanto, não podem ter um índice espacial associado às mesmas.) |

Capítulo 8. Gravando aplicações para Spatial Extender

Este capítulo explica como usar o programa de amostra do Spatial Extender para escrever aplicações para trabalhar e personalizar informações espaciais. Estão incluídos os seguintes tópicos:

- Utilizando o programa de amostra
- As etapas do programa de amostra

Utilizando o programa de amostra

O programa de amostra Spatial Extender torna a programação de aplicação mais fácil. Com o programa de amostra, você pode:

- Automatizar as rotinas de procedimentos espaciais
- Cortar e colar o código de amostra em suas aplicações
- Entender as etapas tipicamente requeridas para criar e manter um banco de dados ativado espacialmente

Use o programa de amostra para codificar tarefas complexas para o Spatial Extender, por exemplo para escrever aplicações que usam interface de banco de dados para chamar os procedimentos armazenados do Spatial Extender. A partir do programa de amostra, você pode copiar e personalizar suas aplicações. Caso não esteja familiarizado com as etapas de programação do Spatial Extender, você pode executar o programa de amostra, que exhibe a você cada etapa em detalhe. Primeiro, porém, você deve criar o programa de amostra. Faça isso com o makefile de amostra. Para obter instruções sobre a criação e a execução do programa de amostra, consulte “Verificando a instalação” na página 26.

As etapas do programa de amostra

A Tabela 5 na página 74 exhibe as etapas do programa de amostra, os procedimentos armazenados associados e uma descrição de cada etapa. As funções C para chamar os procedimentos armazenados são exibidas na coluna Ação da Tabela 5 na página 74 e estão entre parêntesis. Para obter mais informações sobre os procedimentos armazenados, consulte “Capítulo 9. Procedimentos armazenados” na página 83. O programa de amostra está baseado em cenários que estão introduzidos em “Cenário: Uma companhia de seguros atualiza seu GIS” na página 12.

Tabela 5. Programa de amostra do Spatial Extender

| Etapas do programa de amostra | Ação | Descrição |
|--|---|---|
| Ativar/desativar o banco de dados espacial | <ol style="list-style-type: none"> 1. Ativar o banco de dados espacial (gseEnableDB) 2. Desativar o banco de dados espacial (gseDisableDB) 3. Ativar o banco de dados espacial (gseEnableDB) | <ol style="list-style-type: none"> 1. Esta é a primeira etapa necessária para utilizar o Spatial Extender. Um banco de dados que foi ativado para operações espaciais possui um conjunto de tipos espaciais, um conjunto de funções espaciais, um conjunto de predicados espaciais, um novo tipo de índice e um conjunto de administração de tabelas e de exibições. 2. Esta etapa é normalmente executada quando você tem capacidades espaciais ativadas para o banco de dados errado. Ao desativar um banco de dados, você remove um conjunto de tipos espaciais, um conjunto de funções espaciais, um conjunto de predicados espaciais, um novo tipo de índice e um conjunto de administração de tabelas e de exibições. Nota: O desativar banco de dados falhará se houver objetos criados que dependem dos objetos criados pelo ativar procedimento de banco de dados. Por exemplo, ao criar uma tabela com uma coluna espacial do tipo ST_Point causará a falha do desativar banco de dados. Isto ocorre porque a tabela depende do ST_Point que é destinada a ser eliminada pelo desativar procedimento de banco de dados. 3. O mesmo que 1. |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|--|---|---|
| Registrar sistemas de referência espaciais | <ol style="list-style-type: none"> 1. Registrar o sistema de referência espacial para a coluna LOCATION da tabela CUSTOMERS (gseEnableSref) 2. Registrar o sistema de referência espacial para a coluna LOCATION da tabela OFFICES (gseEnableSref) 3. Desregistrar o sistema de referência espacial para a coluna LOCATION da tabela OFFICES (gseDisableSref) 4. Re-registrar o sistema de referência espacial para as colunas ZONE da tabela OFFICES (gseEnableSref) | <ol style="list-style-type: none"> 1. Esta etapa define um novo sistema de referência espacial (SRS) a ser utilizado para interpretar os dados espaciais da tabela CUSTOMERS. Um sistema de referencia espacial inclui dados de figura geométrica de uma maneira que podem ser armazenados em uma coluna de um banco de dados espacial. Após o SRS ser registrado para uma camada específica, as coordenadas aplicáveis para a camada podem ser armazenadas na coluna da tabela CUSTOMERS associada. 2. Esta etapa define um novo sistema de referência espacial (SRS) a ser utilizado para interpretar os dados espaciais da camada OFFICES. Cada camada de tabela deve ter um SRS definido para a mesma. As camadas da tabela OFFICES podem requerer um SRS associado diferente da camada de tabela CUSTOMERS. 3. Esta etapa é executada se você especifica um parâmetro SRS errado para a camada ou coluna espacial. Ao desregistrar um SRS para a camada da tabela OFFICES, você remove a definição com os seus parâmetros associados. 4. Esta etapa define um novo sistema de referência espacial (SRS) a ser utilizado para interpretar os dados espaciais da camada OFFICES. |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|--------------------------------|---|---|
| Criar as tabelas espaciais | <ol style="list-style-type: none"> 1. Alterar a tabela CUSTOMERS incluindo a coluna LOCATION (gseSetupTables) 2. Criar a tabela OFFICES (gseSetupTables) | <ol style="list-style-type: none"> 1. A tabela CUSTOMERS representa dados de negócios que foram armazenados no banco de dados por vários anos. A instrução ALTER TABLE inclui uma nova coluna (LOCATION) do tipo ST_Point. Esta coluna será preenchida pela geocodificação das colunas de endereço em uma etapa subsequente. 2. A tabela OFFICES representa, entre outros dados, uma zona de vendas para cada escritório de uma companhia de seguros. A tabela inteira será preenchida com os dados de atributo de um banco de dados não-DB2 em uma etapa subsequente. Esta etapa envolve a importação de dados de atributo para a tabela OFFICES a partir de um arquivo SHAPE. |
| Registrar as camadas espaciais | <ol style="list-style-type: none"> 1. Registrar a coluna LOCATION na tabela CUSTOMERS como uma camada (gseRegisterLayer) 2. Remover registro da coluna LOCATION na tabela CUSTOMERS (gseUnregisterLayer) 3. Registrar a coluna ZONE da tabela OFFICES como uma camada (gseRegisterLayer) | <p>Estas etapas registram as colunas LOCATION e ZONE como camadas para o Spatial Extender. Antes que uma coluna espacial seja preenchida ou acessada pelos utilitários do Spatial Extender (por exemplo, o geocoder), você precisa registrá-lo como uma camada. Uma camada também pode ter seu registro removido após tornar-se acessível aos utilitários do Spatial Extender. A coluna associada ainda existe após você remover seu registro como uma camada.</p> |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|--------------------------------|--|---|
| Preencher as camadas espaciais | <ol style="list-style-type: none"> 1. Fazer o geocode dos dados de endereços para a coluna LOCATION da tabela CUSTOMERS (gseRunGC) 2. Carregar a tabela OFFICES usando o modo anexar (gseImportShape) 3. Carregar a tabela HAZARD_ZONE usando o modo criar (gseImportShape) | <ol style="list-style-type: none"> 1. Esta etapa executa a geocodificação através da chamada do utilitário geocoder. A geocodificação em batch é normalmente executada quando uma porção significativa da tabela precisa ser geocodificada ou re-geocodificada. 2. Esta etapa carrega a tabela OFFICES com os dados espaciais existentes no formulário de um arquivo SHAPE. Por causa da existência da tabela OFFICES e da camada OFFICES/ZONE estar registrada, o utilitário de carga irá anexar os novos registros a uma tabela existente. 3. Esta etapa carrega a camada HAZARD_ZONE com os dados espaciais existentes no formulário de um arquivo SHAPE. Pelo fato da tabela e da camada não existirem, o utilitário de carga criará a tabela e registrará a camada antes que os dados sejam carregados. |
| Registrar o geocoder | <ul style="list-style-type: none"> • Registrar o geocoder se ele não for o padrão (gseRegisterGc) • Remover o registro do geocoder que você pode ter registrado (gseUnregisterGc) • Registrar o geocoder se ele não for o padrão (gseRegisterGc) | |
| Ativar índices espaciais | <ol style="list-style-type: none"> 1. Ativar o índice espacial para a coluna LOCATION da tabela CUSTOMERS (gseEnableIdx) 2. Ativar o índice espacial para a coluna ZONE da tabela OFFICES (gseEnableIdx) 3. Ativar o índice espacial para a coluna LOCATION da tabela OFFICES (gseEnableIdx) 4. Ativar o índice espacial para a coluna BOUNDARY da tabela HAZARD_ZONE (gseEnableIdx) | <p>Estas etapas ativam o índice espacial para as tabelas CUSTOMERS, OFFICES e HAZARD_ZONE.</p> |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|--|---|--|
| Ativar a geocodificação automática | <ol style="list-style-type: none"> 1. Ativar a geocodificação automática para as colunas LOCATION e ADDRESS da tabela CUSTOMERS (gseEnableAutoGC) | <p>Esta etapa torna a chamada do geocoder automática. A utilização da geocodificação automática causa às colunas LOCATION e ADDRESS da tabela CUSTOMERS a sincronização com cada outra para operações de inclusão e atualização subsequentes.</p> |
| Inserir/atualizar/excluir a tabela CUSTOMERS | <ol style="list-style-type: none"> 1. Inserir alguns registros com uma rua diferente (gseInsDelUpd) 2. Atualizar alguns registros com um novo endereço (gseInsDelUpd) 3. Excluir todos os registros da tabela (gseInsDelUpd) | <p>Estas etapas demonstram uma inserção, atualização e exclusão na coluna LOCATION da tabela CUSTOMERS. Uma vez ativada a geocodificação automática, as informações a partir da coluna ADDRESS são automaticamente geocodificadas quando ela é incluída ou atualizada na coluna LOCATION. Este processo foi ativado na etapa anterior.</p> |
| Desativar a geocodificação automática | <ol style="list-style-type: none"> 1. Desativar a geocodificação automática para a camada CUSTOMERS (gseDisableAutoGC) 2. Desativar o índice espacial para a camada CUSTOMERS (gseDisableIdxCustomersLayer) | <p>Estas etapas desativam a chamada automática do geocoder e do índice espacial em preparação para a próxima etapa (a próxima etapa envolve a re-geocodificação da tabela CUSTOMERS completa). Caso esteja carregando uma grande tabela de geodados, é recomendado que você desative o índice espacial antes de carregar os dados, ativando-o após a conclusão da carga.</p> |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|---|--|--|
| Re-geocodificar a tabela CUSTOMERS | <ol style="list-style-type: none"> 1. Fazer o geocode da camada CUSTOMERS novamente com um nível de precisão baixo – 90% ao invés de 100% (gseRunGC) 2. Reativar o índice espacial para a camada CUSTOMERS (gseEnableIdx) 3. Reativar a geocodificação automática com um nível de precisão baixo – 90% ao invés de 100% (gseEnableAutoGC) | <p>Estas etapas executam o geocoder no modo batch novamente, re-ativa a geocodificação automática com um novo nível de precisão e re-ativa o índice espacial e a geocodificação automática. Esta ação é recomendada quando o administrador espacial informa uma taxa de falhas alta no processo de geocodificação. Se o nível de precisão está definido em 100%, ele pode falhar para o geocode de um endereço porque pode não encontrar um endereço correspondente nos dados de referência. Pela redução do nível de precisão, o geocoder tem uma chance maior de encontrar dados coincidentes. Após a tabela ser re-geocodificada no modo batch, ambos, a geocodificação automática e o índice espacial, são ativados novamente para facilitar a manutenção incremental do índice e da coluna espacial para inclusões e alterações subsequentes.</p> |
| Criar uma exibição e registre suas colunas espaciais como camadas de exibição | <ol style="list-style-type: none"> 1. Criar uma exibição, HIGHRISK_CUSTOMERS, baseada na junção das tabelas CUSTOMERS e HAZARD_ZONE (gseCreateView) 2. Registrar as colunas espaciais da exibição como camadas de exibição (gseRegisterLayer) | <p>Estas etapas criam uma exibição e registram suas colunas espaciais como camadas de exibição.</p> |

Tabela 5. Programa de amostra do Spatial Extender (continuação)

| Etapas do programa de amostra | Ação | Descrição |
|--|---|---|
| Executar análise espacial | <ol style="list-style-type: none"> 1. Localizar a distância média do cliente para cada escritório (ST_Within, ST_Distance) 2. Localiza a renda média do cliente e o premio para cada escritório (ST_Within) 3. Localizar clientes que não estão cobertos por um escritório existente (ST_Within) 4. Localizar o número de zonas de risco que cada escritório sobrepõe (ST_Overlaps) 5. Localizar o escritório mais próximo a partir de uma localização de cliente particular assumindo que o escritório esteja localizado no centro da zona do escritório (ST_Distance, ST_Centroid) 6. Localizar os clientes cuja localização esteja próxima ao limite de uma zona de risco particular (ST_Buffer, ST_Overlaps) 7. Localizar os clientes de risco alto que estão cobertos por um escritório em particular | <p>Estas etapas executam análises espaciais utilizando os predicados espaciais e funções na linguagem SQL do DB2. O otimizador de consultas do DB2 explora o índice espacial nas colunas espaciais para aprimorar o desempenho das consultas sempre que possível.</p> |
| Exportar as camadas espaciais para dentro dos arquivos | <p>Exportar as camadas highRiskCustomers (gseExportShape)</p> | <p>A etapa mostra um exemplo de exportação dos resultados de sua consulta para um arquivo SHAPE. A exportação dos resultados de consulta para outro formato de arquivo permite que as informações sejam utilizadas por uma ferramenta de terceiros (por exemplo, ESRI ArcExplorer Java Versão 3.0).</p> |

Parte 2. Material de referência

Capítulo 9. Procedimentos armazenados

Este capítulo documenta os procedimentos armazenados que permitem a construção de um sistema de informações geográficas com o Spatial Extender. Quando o Spatial Extender é ativado e usado a partir do Centro de Controle, estes procedimentos armazenados são chamados implicitamente. Por exemplo, quando você clica em **OK** a partir de uma janela do Spatial Extender, o DB2 chama os procedimentos armazenados associados a essa janela. Como alternativa, você pode chamar os procedimentos armazenados num programa da aplicação. É aconselhável incluir o arquivo de cabeçalho, `db2gse.h`, em tal programa. Este arquivo contém as definições de macro para as constantes que você atribuir aos parâmetros dos procedimentos armazenados. No AIX, ele está armazenado no diretório `$DB2INSTANCE/sqlib/include/`. No Windows NT, está armazenado no diretório `%DB2PATH%\include\`.

Atenção:

Todas as constantes da cadeia de caracteres dos parâmetros de entrada dos procedimentos armazenados são sensíveis a maiúsculas e minúsculas. Para descobrir quais parâmetros requerem essas constantes, consulte as tabelas deste capítulo.

Você deve especificar em letras maiúsculas qualquer nome de esquema, nome de tabela, nome de exibição, nome de coluna ou nome de camada atribuídos a um parâmetro.

Antes de chamar um procedimento armazenado, seja implícita ou explicitamente, você deve estar conectado ao banco de dados no qual o Spatial Extender está instalado. O primeiro procedimento armazenado que você utilizar será `db2gse.gse_enable_db`. Ele ativa o banco de dados para operações espaciais. Você poderá usar outros procedimentos armazenados somente após a ativação do banco de dados.

As implementações dos procedimentos armazenados estão arquivadas na biblioteca `db2gse` do servidor Spatial Extender.

As seguintes listas podem ser usadas para consulta dos procedimentos armazenados, seja pelos nomes ou pelas tarefas que podem realizar. A primeira lista apresenta os nomes:

- “`db2gse.gse_disable_autogc`” na página 86
- “`db2gse.gse_disable_db`” na página 89

- “db2gse.gse_disable_sref” na página 90
- “db2gse.gse_enable_autogc” na página 91
- “db2gse.gse_enable_db” na página 94
- “db2gse.gse_enable_idx” na página 95
- “db2gse.gse_enable_sref” na página 97
- “db2gse.gse_export_shape” na página 100
- “db2gse.gse_import_sde” na página 102
- “db2gse.gse_import_shape” na página 104
- “db2gse.gse_register_gc” na página 107
- “db2gse.gse_register_layer” na página 109
- “db2gse.gse_run_gc” na página 116
- “db2gse.gse_unregist_gc” na página 118
- “db2gse.gse_unregist_layer” na página 119

A lista seguinte apresenta as tarefas que os procedimentos armazenados executam.

- Criação de um índice para uma coluna espacial (consulte “db2gse.gse_enable_idx” na página 95).
- Criação de um sistema de referência espacial (consulte “db2gse.gse_enable_sref” na página 97).
- Desativação de um geocoder para que não seja possível manter automaticamente colunas espaciais sincronizadas com suas colunas de atributos correspondentes (consulte “db2gse.gse_disable_autogc” na página 86).
- Desativação do suporte para operações espaciais num banco de dados (consulte “db2gse.gse_disable_db” na página 89).
- Eliminação de um sistema de referência espacial (consulte “db2gse.gse_disable_sref” na página 90).
- Ativação de um banco de dados para suportar operações espaciais (consulte “db2gse.gse_enable_db” na página 94).
- Ativação de um geocoder para manter automaticamente colunas espaciais sincronizadas com suas colunas de atributos correspondentes (consulte “db2gse.gse_enable_autogc” na página 91).
- Exportação de uma camada e sua tabela associada a um arquivo de formato (consulte “db2gse.gse_export_shape” na página 100).
- Importação de uma camada e sua tabela associada de um arquivo de transferência do ESRI_SDE (consulte “db2gse.gse_import_sde” na página 102).
- Importação de uma camada e sua tabela associada de um arquivo de formato (consulte “db2gse.gse_import_shape” na página 104).

- Registro de um geocoder diferente do padrão (consulte “db2gse.gse_register_gc” na página 107).
- Registro de uma coluna espacial como uma camada (consulte “db2gse.gse_register_layer” na página 109).
- Execução de um geocoder no modo batch (consulte “db2gse.gse_unregist_gc” na página 118).
- Cancelamento do registro de um geocoder diferente do padrão (consulte “db2gse.gse_unregist_layer” na página 119).
- Cancelamento do registro de uma camada (consulte “db2gse.gse_unregist_layer” na página 119).

Para obter informações sobre as seqüências nas quais é possível executar estas tarefas, consulte “Capítulo 1. Sobre o Spatial Extender” na página 3 e “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

db2gse.gse_disable_autogc

Use este procedimento armazenado para eliminar ou desativar temporariamente disparadores que mantêm uma coluna espacial sincronizada com suas colunas de atributos. Por exemplo, é aconselhável desativar os disparadores enquanto você geocodifica os valores na coluna ou colunas atributos no modo batch. Para mais informações, consulte “Sobre geocodificação” na página 53.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C `gseDisableAutoGc` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual esse procedimento armazenado foi chamado deve ter autorização na forma de uma autoridade, privilégio ou conjunto de privilégios, especificamente:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela na qual estão definidos os disparadores que estão sendo eliminados ou desativados temporariamente.
- O privilégio CONTROL nesta tabela.
- Os privilégios ALTER, SELECT e UPDATE nesta tabela.

Parâmetros de entrada

Tabela 6. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_disable_autogc`.

| Nome | Tipo de dados | Descrição |
|--------------------------|---------------|--|
| <code>operMode</code> | SMALLINT | <p>Indica se os disparadores deverão ser eliminados ou desativados temporariamente.</p> <p>Os disparos eliminados não possuem efeito sobre as instruções SQL.</p> <p>Os disparos desativados temporariamente podem ser recriados sem precisar especificar novamente os parâmetros definidos anteriormente.</p> <p>Este parâmetro não pode ser nulo.</p> <p>Comentário: Para eliminar disparadores, use a macro <code>GSE_AUTOGC_DROP</code>. Para desativá-los temporariamente, use a macro <code>GSE_AUTOGC_INVALIDATE</code>. Para descobrir quais valores estão associados a estas macros, consulte o arquivo <code>db2gse.h</code>. No AIX, ele está armazenado no diretório <code>\$DB2INSTANCE/sqllib/include/</code>. No Windows NT, está armazenado no diretório <code>%DB2PATH%\include\</code>.</p> |
| <code>layerSchema</code> | VARCHAR(30) | <p>Nome do esquema ao qual pertence a tabela ou view especificados no parâmetro <code>layerTable</code>.</p> <p>Este parâmetro pode ser nulo.</p> <p>Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code>, ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_disable_autogc</code> é chamado.</p> |
| <code>layerTable</code> | VARCHAR(128) | <p>Nome da tabela na qual estão definidos os disparadores que você deseja eliminar ou desativar temporariamente.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>layerColumn</code> | VARCHAR(128) | <p>Nome da coluna ativada espacialmente que é mantida pelos disparadores que você deseja eliminar ou desativar temporariamente.</p> <p>Este parâmetro não pode ser nulo.</p> |

Parâmetros de saída

Tabela 7. Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_autogc.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_disable_db

Use este procedimento armazenado para remover recursos que permitem que o Spatial Extender armazene dados espaciais e suporte operações executadas nestes dados.

O objetivo deste procedimento armazenado é ajudá-lo a resolver problemas ou questões que surgirem após você ativar o banco de dados para operações espaciais, porém *antes* inclua qualquer coluna ou dados da tabela nele. Por exemplo, ser depois de ativar um banco de dados para operações espaciais, for decidido que Spatial Extender será usado para outro banco de dados. Contanto que você não tenha definido colunas espaciais ou importado dados espaciais, poderá chamar este procedimento armazenado para remover todos os recursos espaciais do primeiro banco de dados.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C `gseDisableDB` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve ter autoridade SYSADM ou DBADM no banco de dados a partir do qual os recursos do Spatial Extender deverão ser removidos.

Parâmetros de saída

Tabela 8. Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_db.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_disable_sref

Use este procedimento armazenado para eliminar um sistema de referência espacial. Quando este procedimento armazenado estiver processado, as informações sobre o sistema de referência espacial serão removidas da view do catálogo DB2GSE.SPATIAL_REF_SYS. Para obter informações sobre esta view, consulte “DB2GSE.SPATIAL_REF_SYS” na página 145.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C gseDisableSref no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

Nenhuma é necessária.

Parâmetro de entrada

Tabela 9. Parâmetros de entrada para o procedimento armazenado db2gse.gse_disable_sref.

| Nome | Tipo de dados | Descrição |
|------|---------------|--|
| srId | INTEGER | Identificador numérico do sistema de referência espacial que será eliminado. |
| | | Este parâmetro não pode ser nulo. |

Parâmetros de saída

Tabela 10. Parâmetros de saída para o procedimento armazenado db2gse.gse_disable_sref.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

Restrição

Antes de eliminar um sistema de referência espacial, você deverá cancelar o registro de qualquer camada que o utilize. Se tais camadas permanecerem sem registro, a solicitação de eliminação do sistema de referência espacial será rejeitada.

db2gse.gse_enable_autogc

Use este procedimento armazenado para:

- Criar disparadores que irão manter uma coluna espacial sincronizada com as colunas de atributos associadas. Sempre que os valores forem inseridos ou atualizados na coluna atributo, um disparador irá chamar um geocoder registrado para geocodificar os valores inseridos ou atualizados e posicionar os dados resultantes na coluna espacial.
- Reativar os disparadores após sua desativação temporária.
- Estabelecer qual função será usada para geocodificar os valores inseridos e atualizados.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C `gseEnableAutoGC` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual esse procedimento armazenado foi chamado deve ter autorização na forma de uma autoridade, privilégio ou conjunto de privilégios, especificamente:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela na qual estão definidos os disparadores criados pelo procedimento armazenado.
- O privilégio CONTROL nesta tabela.
- Os privilégios ALTER, SELECT e UPDATE nesta tabela.

Parâmetros de entrada

Tabela 11. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_enable_autogc`.

| Nome | Tipo de dados | Descrição |
|--------------------------|---------------|--|
| <code>operMode</code> | SMALLINT | <p>Valor que indica se os disparadores que iniciam a geocodificação serão criados pela primeira vez ou serão reativados após a desativação temporária.</p> <p>Este parâmetro não pode ser nulo.</p> <p>Comentário: Para criar os disparadores, use a macro <code>GSE_AUTOGC_CREATE</code>. Para reativá-los, use a macro <code>GSE_AUTOGC_RECREATE</code>. Para descobrir quais valores estão associados a estas macros, consulte o arquivo <code>db2gse.h</code>. No AIX, ele está armazenado no diretório <code>\$DB2INSTANCE/sqlib/include/</code>. No Windows NT, está armazenado no diretório <code>%DB2PATH%\include\</code>.</p> |
| <code>layerSchema</code> | VARCHAR(30) | <p>Nome do esquema ao qual pertence a tabela especificada no parâmetro <code>layerTable</code>.</p> <p>Este parâmetro pode ser nulo.</p> <p>Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code>, ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_enable_autogc</code> é chamado.</p> |
| <code>layerTable</code> | VARCHAR(128) | <p>Nome da tabela em que serão operados os disparadores criados ou reativados por este procedimento armazenado.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>layerColumn</code> | VARCHAR(128) | <p>Nome da coluna espacial que deverá ser mantida pelos disparadores que este procedimento armazenado cria ou reativa.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>gcId</code> | INTEGER | <p>Identificador do geocoder que será chamado pelos disparadores de inserção e atualização que este procedimento armazenado cria ou reativa.</p> <p>Este parâmetro não pode ser nulo se o parâmetro <code>operMode</code> for definido como <code>GSE_AUTOGC_CREATE</code>. Ele poderá ser nulo se <code>operMode</code> estiver definido como <code>GSE_AUTOGC_RECREATE</code>.</p> |

Tabela 11. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_enable_autogc*. (continuação)

| Nome | Tipo de dados | Descrição |
|----------------|---------------|---|
| precisionLevel | INTEGER | <p>O grau em que os dados fonte devem coincidir dados de referência correspondentes para que o geocoder processe os dados fonte com êxito.</p> <p>Este parâmetro não pode ser nulo se o parâmetro <i>operMode</i> for definido como <i>GSE_AUTOGC_CREATE</i>. Ele poderá ser nulo se <i>operMode</i> estiver definido como <i>GSE_AUTOGC_RECREATE</i>.</p> <p>Comentário: O nível de precisão pode variar de 1 a 100%.</p> |
| vendorSpecific | VARCHAR(256) | <p>Informações técnicas fornecidas pelo fornecedor; por exemplo, o caminho e nome de um arquivo que o fornecedor usa para definir parâmetros.</p> <p>Este parâmetro não pode ser nulo se o parâmetro <i>operMode</i> for definido como <i>GSE_AUTOGC_CREATE</i>. Ele poderá ser nulo se <i>operMode</i> estiver definido como <i>GSE_AUTOGC_RECREATE</i>.</p> |

Parâmetros de saída

Tabela 12. Parâmetros de saída para o procedimento armazenado *db2gse.gse_enable_autogc*.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

Restrições

- O parâmetro *layerColumn* deve fazer referência a uma coluna que tenha sido registrada como camada da tabela.
- Se o parâmetro *operMode* estiver definido em *GSE_AUTOGC_CREATE*, você deverá atribuir um identificador de um geocoder registrado ao parâmetro *gcId*.

db2gse.gse_enable_db

Use este procedimento armazenado para fornecer a um banco de dados os recursos de que ele precisa para armazenar dados espaciais e suportar operações. Estes recursos incluem tipos de dados espaciais, um tipo de índice espacial, tabelas e exibições do catálogo, funções fornecidas e outros procedimentos armazenados. A biblioteca externa e o nome da função para este procedimento armazenado é db2gse.gse_enable_db.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C gseEnableDB no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual o procedimento armazenado é chamado deve ter autoridade SYSADM ou DBADM no banco de dados que está sendo ativado.

Parâmetros de saída

Tabela 13. Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_db.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_enable_idx

Use este procedimento armazenado para criar um índice para uma coluna espacial.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C `gseEnableIdx` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela na qual o índice ativado deverá ser usado.
- O privilégio CONTROL ou INDEX nesta tabela.

Parâmetros de entrada

Tabela 14. Parâmetros de entrada para o procedimento armazenado db2gse.gse_enable_idx.

| Nome | Tipo de dados | Descrição |
|-------------|---------------|--|
| layerSchema | VARCHAR(30) | Nome do esquema ao qual pertence a tabela especificada no parâmetro layerTable. Este parâmetro pode ser nulo. Comentário: Forneça um valor para este parâmetro. O parâmetro pode ser um valor NULL. |
| layerTable | VARCHAR(128) | Nome da tabela na qual deverá ser definido o índice que você está criando. Este parâmetro não pode ser nulo. |
| layerColumn | VARCHAR(128) | Nome da coluna ativada espacialmente que deverá ser pesquisada com a ajuda do índice que está sendo criado. Este parâmetro não pode ser nulo. |
| indexName | VARCHAR(128) | Nome do índice que deverá ser criado. Este parâmetro não pode ser nulo. Comentário: Não especifique um nome de esquema. O Spatial Extender automaticamente atribui o índice ao esquema referido pelo parâmetro layerSchema. |

Tabela 14. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_enable_idx*. (continuação)

| Nome | Tipo de dados | Descrição |
|-----------|---------------|---|
| gridSize1 | DOUBLE | Número que indica qual deve ser a granulosidade da melhor grade de índice. Este parâmetro não pode ser nulo. |
| gridSize2 | DOUBLE | Número que indica (1) que não deve haver segunda grade para este índice ou (2) qual deve ser a granulosidade da segunda grade. Este parâmetro pode ser nulo. Comentário: Se a segunda grade não deve existir, especifique 0. Se deseja uma segunda grade, ela deverá ser menos granulosa que a grade indicada por gridSize1. |
| gridSize3 | DOUBLE | Número que indica (1) que não deve haver terceira grade para este índice ou (2) qual deve ser a granulosidade da terceira grade. Este parâmetro pode ser nulo. Comentário: Se a terceira grade não deve existir, especifique 0. Se deseja uma terceira grade, ela deverá ser menos granulosa que a grade indicada por gridSize2. |

Parâmetros de saída

Tabela 15. Parâmetros de saída para o procedimento armazenado *db2gse.gse_enable_idx*.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_enable_sref

Use este procedimento armazenado para especificar como os números negativos e decimais em um sistema de coordenadas específico deverão ser convertidos em inteiros positivos para que o Spatial Extender possa armazená-los. Suas especificações são denominadas coletivamente *sistema de referência espacial*. Quando este procedimento armazenado estiver processado, as informações sobre o sistema de referência espacial serão incluídas na view do catálogo DB2GSE.SPATIAL_REF_SYS. Para obter informações sobre esta view, consulte “DB2GSE.SPATIAL_REF_SYS” na página 145.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C gseEnableSref no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

Nenhuma é necessária.

Parâmetros de entrada

Tabela 16. Parâmetros de entrada para o procedimento armazenado db2gse.gse_enable_sref.

| Nome | Tipo de dados | Descrição |
|--------|---------------|---|
| srId | INTEGER | Um identificador numérico para o sistema de referência espacial. Este parâmetro não pode ser nulo. Comentário: Este identificador deve ser exclusivo dentro do banco de dados ativado espacialmente. |
| srName | VARCHAR(64) | Descrição breve do sistema de referência espacial. Este parâmetro não pode ser nulo. Comentário: Esta descrição deve ser exclusiva dentro do banco de dados ativado espacialmente. |
| falsex | DOUBLE | Um número que, quando subtraído de um valor negativo da coordenada X, deixa um número não-negativo (ou seja, um número positivo ou zero). Este parâmetro não pode ser nulo. |

Tabela 16. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_enable_sref*. (continuação)

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| falsey | DOUBLE | Um número que, quando subtraído de um valor negativo da coordenada Y, permite um número não-negativo (ou seja, um número positivo ou zero). Este parâmetro não pode ser nulo. |
| xyunits | DOUBLE | Um número que, quando multiplicado por uma coordenada X decimal ou uma coordenada Y decimal, produz um inteiro que pode ser armazenado como um item de dados de 32 bits. Este parâmetro não pode ser nulo. |
| falsez | DOUBLE | Um número que, quando subtraído de um valor negativo da coordenada Z, permite um número não-negativo (ou seja, um número positivo ou zero). Este parâmetro não pode ser nulo. |
| zunits | DOUBLE | Um número que, quando multiplicado por uma coordenada Z decimal, produz um inteiro que pode ser armazenado como um item de dados de 32 bits. Este parâmetro não pode ser nulo. |
| falsem | DOUBLE | Um número que, quando subtraído de uma medida negativa, permite um número não-negativo (ou seja, um número positivo ou zero). Este parâmetro não pode ser nulo. |
| munits | DOUBLE | Um número que, quando multiplicado por uma medida decimal, produz um inteiro que pode ser armazenado como um item de dados de 32 bits. Este parâmetro não pode ser nulo. |
| scId | INTEGER | Identificador numérico do sistema de coordenadas do qual é derivado o sistema de referência espacial. Para descobrir o que é um identificador numérico do sistema de coordenadas, consulte a view do catálogo DB2GSE.COORD_REF_SYS “DB2GSE.COORD_REF_SYS” na página 143. Este parâmetro não pode ser nulo. |

Parâmetros de saída

Tabela 17. Parâmetros de saída para o procedimento armazenado db2gse.gse_enable_sref.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_export_shape

Use este procedimento armazenado para exportar uma camada e sua tabela associada a um arquivo de formato ou criar um novo arquivo de formato e exportar uma camada e sua tabela associada para este novo arquivo.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C `gseExportShape` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve ter o privilégio `SELECT` na tabela que deverá ser exportada.

Parâmetros de entrada

Tabela 18. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_export_shape`.

| Nome | Tipo de dados | Descrição |
|--------------------------|---------------------------|---|
| <code>layerSchema</code> | <code>VARCHAR(30)</code> | Nome do esquema ao qual pertence a tabela especificada no parâmetro <code>layerTable</code> . Este parâmetro pode ser nulo. Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code> , ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_export_shape</code> é chamado. |
| <code>layerTable</code> | <code>VARCHAR(128)</code> | Nome da tabela que você está exportando. Este parâmetro não pode ser nulo. |
| <code>layerColumn</code> | <code>VARCHAR(30)</code> | Nome da coluna que foi registrada como sendo a camada que você está exportando. Este parâmetro não pode ser nulo. |
| <code>fileName</code> | <code>VARCHAR(128)</code> | Nome do arquivo de formato para o qual a camada especificada deverá ser exportada. Este parâmetro não pode ser nulo. |

Tabela 18. Parâmetros de entrada para o procedimento armazenado db2gse.gse_export_shape. (continuação)

| Nome | Tipo de dados | Descrição |
|-------------|---------------|---|
| whereClause | VARCHAR(1024) | O corpo da cláusula where. Define uma restrição em um conjunto de linhas a serem exportadas. A cláusula pode referenciar qualquer coluna de atributo na tabela que você está exportando. A palavra-chave WHERE não é necessária nesta cláusula. |

Este parâmetro pode ser nulo.

Parâmetros de saída

Tabela 19. Parâmetros de saída para o procedimento armazenado db2gse.gse_export_shape.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

Restrição

É possível exportar somente uma camada por vez.

db2gse.gse_import_sde

Use este procedimento armazenado para importar um arquivo de transferência SDE para um banco de dados que tenha sido ativado para operações espaciais. O procedimento armazenado pode operar de uma destas duas formas:

- Se o arquivo de transferência SDE destinar-se a uma tabela existente que tenha uma coluna de camada registrada, o Spatial Extender carregará a tabela com os dados do arquivo.
- Caso contrário, Spatial Extender criará uma tabela que tem uma coluna espacial, registrará esta coluna como camada e carregará a camada e as outras colunas da tabela com os dados do arquivo.

O sistema de referência espacial especificado no arquivo de transferência SDE será comparado aos sistemas de referência espacial registrados no Spatial Extender. Se o sistema especificado corresponder a um sistema registrado, os valores negativos e decimais nos dados de transferência serão modificados, quando carregados, do modo estabelecido pelo sistema registrado. Se o sistema especificado não corresponder aos sistemas registrados, o Spatial Extender criará um novo sistema de referência espacial para estabelecer as modificações.

Autorização

Ao importar dados para uma tabela existente, a ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela para a qual os dados serão importados.
- O privilégio CONTROL nesta tabela.

Quando a tabela para qual você deseja importar dados deve ser criada, a ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela que deverá ser criada.

Parâmetros de entrada

Tabela 20. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_import_sde*.

| Nome | Tipo de dados | Descrição |
|-------------|---------------|--|
| layerSchema | VARCHAR(30) | Nome do esquema ao qual pertence a tabela ou view especificados no parâmetro layerTable. Este parâmetro pode ser nulo. Deve ter 30 caracteres ou menos. Comentário: Se você não fornecer um valor para o parâmetro layerSchema, ele assumirá a ID de usuário com que o procedimento armazenado <i>db2gse.gse_import_sde</i> é chamado. |
| layerTable | VARCHAR(128) | Nome da tabela na qual devem ser carregados os dados de transferência SDE. Este parâmetro não pode ser nulo. |
| layerColumn | VARCHAR(30) | Nome da coluna que foi registrada como camada na qual serão carregados os dados espaciais do arquivo de transferência SDE. Este parâmetro não pode ser nulo. Deve ter 30 caracteres ou menos. |
| fileName | VARCHAR(128) | Nome do arquivo de transferência SDE que deverá ser importado. Este parâmetro não pode ser nulo. |
| commitScope | INTEGER | Número de registros por ponto de verificação. Este parâmetro pode ser nulo. |

Parâmetros de saída

Tabela 21. Parâmetros de saída para o procedimento armazenado *db2gse.gse_import_sde*.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_import_shape

Utilize este procedimento armazenado para importar um arquivo de formato do ESRI para um banco de dados que tenha sido ativado para operações espaciais. O procedimento armazenado pode operar de uma destas duas formas:

- Se o arquivo de formato destina-se a uma tabela existente que tenha uma coluna de camada registrada, o Spatial Extender carregará a tabela com os dados do arquivo.
- Caso contrário, Spatial Extender criará uma tabela que tem uma coluna espacial, registrará esta coluna como camada e carregará a camada e as outras colunas da tabela com os dados do arquivo.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função `C gseImportShape` no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Ao importar um conjunto de representações de formato do ESRI, você recebe, pelo menos, dois arquivos. Todos os arquivos possuem o mesmo prefixo para o nome, mas extensões diferentes. Por exemplo, as extensões dos dois arquivos que você sempre recebe são `.shp` e `.shx`.

Para receber os arquivos de um conjunto de representações de formato, atribua o nome que os arquivos possuem em comum ao parâmetro `fileName`. Não especifique uma extensão. Dessa forma, você pode certificar-se de que todos os arquivos de que precisa - o arquivo `.shp`, o arquivo `.shx` e quaisquer outros que possam estar incluídos - serão importados.

Por exemplo, suponha que um conjunto de representações de formatos do ESRI esteja armazenado nos arquivos chamados `Lakes.shp` e `Lakes.shx`. Ao importar estas representações, você deve atribuir somente o nome `Lakes` no parâmetro `fileName`.

Os arquivos de transferência SDE possuem nomes, mas não extensões. Portanto, ao importar um arquivo de transferência SDE, você atribui seu nome, sem a extensão, ao parâmetro `fileName`.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Autoridade `SYSADM` ou `DBADM` no banco de dados que contém a tabela na qual serão carregados os dados de formato importados.
- O privilégio `CONTROL` nesta tabela.

Parâmetros de entrada

Tabela 22. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_import_shape`.

| Nome | Tipo de dados | Descrição |
|----------------------------|---------------------------|---|
| <code>layerSchema</code> | <code>VARCHAR(30)</code> | <p>Nome do esquema ao qual pertence a tabela ou view especificados no parâmetro <code>layerTable</code>.</p> <p>Este parâmetro pode ser nulo.</p> <p>Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code>, ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_import_shape</code> é chamado.</p> |
| <code>layerTable</code> | <code>VARCHAR(128)</code> | <p>Nome da tabela na qual deve ser carregado o arquivo de formato é importado.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>layerColumn</code> | <code>VARCHAR(30)</code> | <p>Nome da coluna que foi registrada como camada na qual serão carregados os dados de formato.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>fileName</code> | <code>VARCHAR(128)</code> | <p>Nome do arquivo de formato que deverá ser importado.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>exceptionFile</code> | <code>VARCHAR(128)</code> | <p>Caminho e nome do arquivo no qual serão armazenados os formatos que não puderam ser importados. Este é um novo arquivo que será criado quando o procedimento armazenado <code>db2gse.gse_import_shape</code> for executado.</p> <p>Atribua um nome do arquivo, sem a extensão, ao parâmetro <code>exceptionFile</code>.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>srId</code> | <code>INTEGER</code> | <p>Identificador do sistema de referência espacial a ser usado pela camada na qual os dados de formato serão carregados.</p> <p>Este parâmetro pode ser nulo.</p> <p>Comentário: Se este identificador não estiver especificado, a transformação interna será definida com a resolução máxima possível para o arquivo de formato.</p> |

Tabela 22. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_import_shape*. (continuação)

| Nome | Tipo de dados | Descrição |
|-------------|---------------|---|
| commitScope | INTEGER | Número de registros por ponto de verificação. Este parâmetro que é I foi nulo. |

Parâmetros de saída

Tabela 23. Parâmetros de saída para o procedimento armazenado *db2gse.gse_import_shape*.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_register_gc

Use este procedimento armazenado para registrar um geocoder diferente do padrão. Para saber se um geocoder já foi registrado, consulte a view do catálogo DB2GSE.SPATIAL_GEOCODER (descrita em “DB2GSE.SPATIAL_GEOCODER” na página 144).

Autorização

A ID de usuário com a qual o procedimento armazenado é chamado deve ter autoridade SYSADM ou DBADM no banco de dados que contém o geocoder que este procedimento armazenado registra.

Parâmetros de entrada

Tabela 24. Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_gc.

| Nome | Tipo de dados | Descrição |
|----------------|---------------|---|
| gcId | INTEGER | Identificador numérico do geocoder que você está registrando. Este parâmetro não pode ser nulo. Comentário: Este identificador deve ser exclusivo dentro do banco de dados. |
| gcName | VARCHAR(64) | Descrição breve do geocoder que você está registrando. Este parâmetro não pode ser nulo. Comentário: Esta descrição deve ser uma cadeia de caracteres exclusiva dentro do banco de dados. |
| vendorName | VARCHAR(64) | Nome do fornecedor que ofereceu o geocoder que você está registrando. Este parâmetro não pode ser nulo. |
| primaryUDF | VARCHAR(256) | Nome completo do geocoder que você está registrando. Este parâmetro não pode ser nulo. |
| precisionLevel | INTEGER | O grau em que os dados fonte devem coincidir dados de referência correspondentes para que o geocoder processe os dados fonte com êxito. Este parâmetro não pode ser nulo. Comentário: O nível de precisão pode variar de 1 a 100%. |

Tabela 24. Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_gc. (continuação)

| Nome | Tipo de dados | Descrição |
|----------------|----------------------|---|
| vendorSpecific | VARCHAR(256) | Informações técnicas fornecidas pelo fornecedor; por exemplo, o caminho e nome de um arquivo que o fornecedor usa para definir parâmetros. Este parâmetro pode ser nulo. |
| geoArea | VARCHAR(256) | Área geográfica a ser geocodificada. Este parâmetro pode ser nulo. |
| descrição | VARCHAR(256) | Observações apresentadas pelo fornecedor Este parâmetro pode ser nulo. |

Parâmetros de saída

Tabela 25. Parâmetros de saída para o procedimento armazenado db2gse.gse_register_gc.

| Nome | Tipo de dados | Descrição |
|-------------|----------------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_register_layer

Use este procedimento armazenado para registrar uma coluna espacial como camada. Quando este procedimento armazenado estiver processado, as informações sobre a camada que está sendo registrada serão incluídas na view do catálogo DB2GSE.GEOMETRY_COLUMNS. Para obter informações sobre esta view, consulte “DB2GSE.GEOMETRY_COLUMNS” na página 144.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C gseRegisterLayer no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Este procedimento armazenado não funcionará nos seguintes tipos de tabelas:

- A = Alias
- H = Tabela de Hierarquia
- N = Pseudônimo
- S = Tabela de Resumo
- U = Tabela Representada
- W = Exibição Representada

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Para uma camada da tabela:
 - Autoridade SYSADM ou DBADM no banco de dados que contém a tabela à qual pertence esta camada.
 - O privilégio CONTROL ou ALTER nesta tabela.
- Para uma camada da view:
 - O privilégio SELECT na tabela base que contém (1) os dados de endereço que serão geocodificados para esta camada e (2) os dados espaciais que resultarem da geocodificação.

Parâmetros de entrada

Tabela 26. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_register_layer`.

| Nome | Tipo de dados | Descrição |
|----------------------------|---------------|---|
| <code>layerSchema</code> | INTEGER(30) | <p>Nome do esquema ao qual pertence a tabela ou view especificados no parâmetro <code>layerTable</code>.</p> <p>Este parâmetro pode ser nulo.</p> <p>Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code>, ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_register_layer</code> é chamado.</p> |
| <code>layerTable</code> | VARCHAR(128) | <p>Nome da tabela ou view que contém a coluna que está sendo registrada como camada.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>layerColumn</code> | VARCHAR(128) | <p>Nome da coluna que está sendo registrada como camada. Para uma tabela, se a coluna não existir, o Spatial Extender irá incluí-la utilizando a instrução ALTER. Para uma exibição, a coluna já deve existir.</p> <p>Pode ser especificada apenas uma coluna para o parâmetro <code>layerColumn</code>. Portanto, quando você registra várias colunas de uma tabela ou exibição como camadas, é necessário executar este procedimento armazenado separadamente para cada coluna.</p> <p>Este parâmetro não pode ser nulo.</p> |
| <code>layerTypeName</code> | VARCHAR(64) | <p>Tipo de dados da coluna que está sendo registrada como camada. Apenas os tipos de dados fornecidos pelo Spatial Extender são aceitos. Você deve especificar o tipo de dados em maiúsculas, por exemplo:</p> <p>ST_POINT</p> <p>Não é necessário especificar o nome do esquema, uma vez que ele é incluído automaticamente.</p> <p>Este parâmetro não poderá ser nulo se a coluna for uma coluna de tabela que deve ser criada quando esse procedimento armazenado for processado. Caso contrário, se a coluna for uma coluna existente dentro de uma tabela ou view, esse parâmetro poderá ser nulo.</p> |

Tabela 26. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_register_layer*. (continuação)

| Nome | Tipo de dados | Descrição |
|-----------|---------------|---|
| srId | INTEGER | <p>Identificador do sistema de referência espacial usado para esta camada.</p> <p>Este parâmetro não pode ser nulo para a camada de uma tabela. O Spatial Extender ignora este parâmetro quando uma camada da view é registrada.</p> |
| geoSchema | VARCHAR(30) | <p>Aplica-se quando uma coluna da view é registrada como camada. O parâmetro geoSchema é o esquema da tabela que sustenta a view à qual pertence a coluna.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma view como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da tabela é registrada como camada.</p> <p>Exibições baseadas em mais de uma tabela base ou outras exibições não são suportadas por este parâmetro.</p> <p>Comentário: Se você não fornecer um valor para o parâmetro geoSchema, ele assumirá o valor do parâmetro layerSchema.</p> |
| geoTable | VARCHAR(128) | <p>Aplica-se quando uma coluna da view é registrada como camada. O parâmetro geoTable é o nome da tabela que sustenta a view à qual pertence a coluna.</p> <p>Exibições baseadas em mais de uma tabela base ou outras exibições não são suportadas por este parâmetro.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma view como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da tabela é registrada como camada.</p> |

Tabela 26. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_register_layer*. (continuação)

| Nome | Tipo de dados | Descrição |
|-------------|---------------|---|
| geoColumn | VARCHAR(128) | <p>Aplica-se quando uma coluna da view é registrada como camada. O parâmetro geoColumn é o nome da coluna da tabela que sustenta esta coluna da view.</p> <p>Exibições baseadas em mais de uma tabela base ou outras exibições não são suportadas por este parâmetro.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma view como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da tabela é registrada como camada.</p> |
| nAttributes | SMALLINT | <p>Número de colunas que contêm os dados fonte que deverão ser geocodificados para esta camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> |
| attr1Name | VARCHAR(128) | <p>Nome da primeira coluna que contém os dados fonte que deverão ser geocodificados para esta camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>Se você pretende usar o geocoder padrão, precisará armazenar os endereços das ruas na coluna attr1Name.</p> |

Tabela 26. Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_layer. (continuação)

| Nome | Tipo de dados | Descrição |
|-------------|----------------------|--|
| attr2Name | VARCHAR(128) | <p>Nome da segunda coluna que contém os dados fonte que deverão ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>Se você pretende usar o geocoder padrão, precisará armazenar os nomes das cidades na coluna attr2Name.</p> |
| attr3Name | VARCHAR(128) | <p>Nome da terceira coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>Se você pretende usar o geocoder padrão, precisará armazenar os nomes ou abreviações dos estados na coluna attr3Name.</p> |
| attr4Name | VARCHAR(128) | <p>Nome da quarta coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>Se você pretende usar o geocoder padrão, precisará armazenar os CEPs na coluna attr4Name.</p> |

Tabela 26. Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_layer. (continuação)

| Nome | Tipo de dados | Descrição |
|-------------|----------------------|---|
| attr5Name | VARCHAR(128) | <p>Nome da quinta coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr5Name.</p> |
| attr6Name | VARCHAR(128) | <p>Nome da sexta coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr6Name.</p> |
| attr7Name | VARCHAR(128) | <p>Nome da sétima coluna que contém os dados fonte que devem ser geocodificados para essa coluna.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr7Name.</p> |
| attr8Name | VARCHAR(128) | <p>Nome da oitava coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr8Name.</p> |

Tabela 26. Parâmetros de entrada para o procedimento armazenado db2gse.gse_register_layer. (continuação)

| Nome | Tipo de dados | Descrição |
|------------|---------------|--|
| attr9Name | VARCHAR(128) | <p>Nome da nona coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr9Name.</p> |
| attr10Name | VARCHAR(128) | <p>Nome da décima coluna que contém os dados fonte que devem ser geocodificados para essa camada.</p> <p>Este parâmetro pode ser nulo quando você registra a coluna de uma tabela como uma camada. O Spatial Extender ignora este parâmetro quando uma coluna da view é registrada como camada.</p> <p>O geocoder padrão ignora a coluna Attr10Name.</p> |

Parâmetros de saída

Tabela 27. Parâmetros de saída para o procedimento armazenado db2gse.gse_register_layer.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

Restrições

- Se você estiver registrando uma coluna da view como camada, este registro deverá estar baseado numa coluna de tabela que já esteja registrada como camada.
- Não mais que dez colunas de atributos podem conter os dados que deverão ser geocodificados para a camada que você está registrando.

db2gse.gse_run_gc

Use este procedimento armazenado para executar um geocoder no modo batch. Para obter informações sobre esta tarefa, consulte “Executando o geocoder no modo batch” na página 57.

Para obter um exemplo do código para chamada deste procedimento armazenado, consulte a função C gseRunGC no programa de amostra. Para obter informações sobre este programa, consulte “Capítulo 8. Gravando aplicações para Spatial Extender” na página 73.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Autoridade SYSADM ou DBADM no banco de dados que contém a tabela na qual o geocoder especificado deve agir.
- O privilégio CONTROL ou UPDATE nesta tabela.

Parâmetros de entrada

Tabela 28. Parâmetros de entrada para o procedimento armazenado db2gse.gse_run_gc.

| Nome | Tipo de dados | Descrição |
|-------------|---------------|--|
| layerSchema | VARCHAR(30) | Nome do esquema ao qual pertence a tabela ou view especificados no parâmetro layerTable. Este parâmetro pode ser nulo. Comentário: Se você não fornecer um valor para o parâmetro layerSchema, ele assumirá a ID de usuário com que db2gse.gse_run_gc será chamado. |
| layerTable | VARCHAR(128) | Nome da tabela que contém a coluna na qual os dados geocodificados serão inseridos. Este parâmetro não pode ser nulo. |
| layerColumn | VARCHAR(128) | Nome da coluna na qual os dados geocodificados serão inseridos. Este parâmetro não pode ser nulo. |

Tabela 28. Parâmetros de entrada para o procedimento armazenado *db2gse.gse_run_gc*. (continuação)

| Nome | Tipo de dados | Descrição |
|----------------|---------------|---|
| gcId | INTEGER | Identificador do geocoder que você deseja executar. Este parâmetro pode ser nulo. Para descobrir os identificadores dos geocoders registrados, consulte a view do catálogo DB2GSE.SPATIAL_GEOCODER. |
| precisionLevel | INTEGER | O grau em que os dados fonte devem coincidir dados de referência correspondentes para que o geocoder processe os dados fonte com êxito. Este parâmetro pode ser nulo. Comentário: O nível de precisão pode variar de 1 a 100%. |
| vendorSpecific | VARCHAR(256) | Informações técnicas fornecidas pelo fornecedor; por exemplo, o caminho e nome de um arquivo que o fornecedor usa para definir parâmetros. Este parâmetro pode ser nulo. |
| whereClause | VARCHAR(256) | O corpo da cláusula WHERE. Define uma restrição ao conjunto de registros que serão geocodificados. A cláusula pode referenciar qualquer coluna de atributo na tabela em que o geocoder será operado. Este parâmetro pode ser nulo. |
| commitScope | INTEGER | Número de registros por ponto de verificação. Este parâmetro pode ser nulo. |

Parâmetros de saída

Tabela 29. Parâmetros de saída para o procedimento armazenado *db2gse.gse_run_gc*.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_unregist_gc

Use este procedimento armazenado para cancelar um registro diferente do geocoder padrão.

Para procurar informações sobre o geocoder ao qual você deseja cancelar o registro, consulte a view do catálogo DB2GSE.SPATIAL_GEOCODER, consulte “DB2GSE.SPATIAL_GEOCODER” na página 144.

Autorização

A ID de usuário com a qual o procedimento armazenado é chamado deve ter autoridade SYSADM ou DBADM no banco de dados que contém o geocoder que deverá ter o registro cancelado.

Parâmetro de entrada

Tabela 30. Parâmetros de entrada para o procedimento armazenado db2gse.gse_unregist_gc.

| Nome | Tipo de dados | Descrição |
|-----------------------------------|---------------|--|
| gcId | INTEGER | O identificador do geocoder que deverá ter o registro cancelado. |
| Este parâmetro não pode ser nulo. | | |

Parâmetros de saída

Tabela 31. Parâmetros de saída para o procedimento armazenado db2gse.gse_unregist_gc.

| Nome | Tipo de dados | Descrição |
|---------|---------------|---|
| msgCode | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| msgText | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

db2gse.gse_unregist_layer

Use este procedimento armazenado para cancelar o registro de uma camada. O procedimento armazenado faz isto:

- Removendo a definição da camada das tabelas de catálogo do Spatial Extender.
- Excluindo a restrição de verificação que o Spatial Extender colocou na tabela base desta camada para assegurar que os dados espaciais da camada atendam aos requisitos do sistema de referência espacial da camada.
- Eliminando os disparadores que são usados para atualizar a coluna espacial sempre que os dados do endereço forem incluídos, alterados ou removidos.

Quando os dados de endereço em uma linha da tabela são geocodificados, os dados espaciais resultantes serão colocados na mesma linha. Portanto, se a linha for excluída, os dados do endereço e os dados espaciais serão excluídos ao mesmo tempo. Os disparos não excluem os dados espaciais.

Quando o procedimento armazenado é processado, as informações sobre a camada são removidas da exibição do catálogo DB2GSE.GEOMETRY_COLUMNS. Para obter informações sobre esta view, consulte "DB2GSE.GEOMETRY_COLUMNS" na página 144.

Autorização

A ID de usuário com a qual este procedimento armazenado é chamado deve conter um dos seguintes privilégios ou autoridades:

- Para uma camada da tabela:
 - Autoridade SYSADM ou DBADM no banco de dados que contém a tabela base desta camada.
 - O privilégio CONTROL ou ALTER nesta tabela.
- Para uma camada da view:
 - O privilégio SELECT na tabela base que contém (1) os dados de endereço que serão geocodificados para esta camada e (2) os dados espaciais que resultarem da geocodificação.

Parâmetros de entrada

Tabela 32. Parâmetros de entrada para o procedimento armazenado `db2gse.gse_unregister_layer`.

| Nome | Tipo de dados | Descrição |
|--------------------------|---------------|---|
| <code>layerSchema</code> | VARCHAR(30) | Nome do esquema ao qual pertence a tabela especificada no parâmetro <code>layerTable</code> . Este parâmetro pode ser nulo. Comentário: Se você não fornecer um valor para o parâmetro <code>layerSchema</code> , ele assumirá a ID de usuário com que o procedimento armazenado <code>db2gse.gse_unregister_layer</code> é chamado. Você deve especificar em letras maiúsculas qualquer nome de esquema, nome de tabela, nome de exibição, nome de coluna ou nome de camada atribuídos a um parâmetro. |
| <code>layerTable</code> | VARCHAR(128) | Nome da tabela que contém a coluna especificada no parâmetro <code>layerColumn</code> . Este parâmetro não pode ser nulo. |
| <code>layerColumn</code> | VARCHAR(128) | Nome da coluna espacial que foi definida como a camada que você deseja cancelar o registro. Este parâmetro não pode ser nulo. Comentário: Pode ser especificada apenas uma camada para o parâmetro <code>layerColumn</code> . Portanto, ao remover o registro de várias camadas em uma tabela ou exibição, será necessário executar este procedimento armazenado separadamente para cada camada. |

Parâmetros de saída

Tabela 33. Parâmetros de saída para o procedimento armazenado `db2gse.gse_unregister_layer`.

| Nome | Tipo de dados | Descrição |
|----------------------|---------------|---|
| <code>msgCode</code> | INTEGER | Código associado às mensagens que o responsável pela chamada deste procedimento armazenado poderá retornar. |
| <code>msgText</code> | VARCHAR(1024) | Mensagem de erro completa, conforme construída no servidor Spatial Extender. |

Restrição

Se uma coluna da view definida como camada da view estiver baseada numa coluna da tabela, que foi definida como uma camada da tabela, não será possível cancelar o registro desta camada da tabela até que o registro da camada da view seja cancelado.

Capítulo 10. Mensagens

O DB2 Spatial Extender gera mensagens que são retornadas por:

- Centro de Controle
- Procedimentos armazenados
- Funções espaciais

Cada mensagem possui um identificador que consiste em um prefixo e um número de mensagem. O número da mensagem também é conhecido como *SQLCODE*.

Há três tipos de mensagens: de erro, de aviso e informativas. Os identificadores de mensagens que terminam em *E* indicam mensagens de erros. Os que terminam em *W* indicam mensagens de aviso. Os identificadores de mensagens que terminam em *I* indicam mensagens informativas.

Mensagens retornadas pelo Centro de Controle

As mensagens a seguir são retornadas pelo Centro de Controle. Seus códigos *SQLCODE* iniciam com as letras "DBA".

DBA7200E Mais de 10 colunas estão selecionadas como entrada para um geocoder

Explicação: Até 10 colunas podem ser selecionadas como entrada para um geocoder

Resposta do Usuário: Mova os nomes das colunas da caixa colunas Selecionadas para a caixa colunas Disponíveis até que a caixa Colunas Selecionadas liste dez nomes ou menos.

DBA7201E O banco de dados não está ativado para operações do Spatial Extender.

Explicação: O banco de dados deve estar ativo para o Spatial Extender antes que você possa utilizá-lo.

Resposta do Usuário: Dê um clique com o botão direito no banco de dados e selecione Spatial Extender —> Ativar a partir dos menus.

Mensagens retornadas pelos procedimentos armazenados

As mensagens a seguir são retornadas pelos procedimentos armazenados. Seus códigos *SQLCODES* iniciam com as letras "GSE", seguidas de números no intervalo de 0000 a 2.035.

Atenção: Duas mensagens de erro — GSE2022E e GSE2035E — não se encontram no catálogo de mensagens do DB2 Spatial Extender. Portanto, quando os erros aos quais essas mensagens estão associadas ocorrem, você recebe uma notificação online de que elas não podem ser recuperadas, com a seguinte mensagem: Não foi possível recuperar a mensagem SQL10007N "<SQLCODE – 2022 ou – 2035>". Código de razão: "4".

GSE0000I A operação foi concluída com sucesso.

GSE0001E O Spatial Extender não pôde executar a operação solicitada ("<nome da operação>") sob a ID de usuário "<id do usuário>".

Explicação: Você solicitou esta operação sob uma ID de usuário que não mantém o privilégio ou autoridade para executar a operação.

Resposta do Usuário: Consulte a documentação para localizar a autorização apropriada ou obtê-la de um administrador do Spatial Extender.

GSE0002E O "<valor>" não é um valor válido para o argumento "<nome do argumento>".

Explicação: O valor que você digitou estava incorreto.

Resposta do Usuário: Consulte a documentação ou um administrador do Spatial Extender para localizar qual valor ou faixa de valores você precisa especificar.

GSE0003E O Spatial Extender não pôde executar a operação solicitada porque o argumento "<nome do argumento>" não foi especificado.

Explicação: Você não especificou um argumento que é requerido para esta operação.

Resposta do Usuário: Especifique o argumento "<nome do argumento>" com o valor que deseja; em seguida solicite a operação novamente.

GSE0004W O argumento "<nome do argumento>" não foi avaliado.

Explicação: A operação que você solicitou não usa o argumento "<nome do argumento>".

Resposta do Usuário: Nenhuma é necessária.

GSE0005E O Spatial Extender não pôde processar sua solicitação para criar um objeto chamado "<nome do objeto>".

Explicação: Ou o objeto "<nome do objeto>" já existe, ou você não possui a permissão apropriada para criá-lo. Ele pode ser uma tabela, coluna, disparador, índice, arquivo ou outro tipo de objeto.

Resposta do Usuário: Se o "<nome do objeto>" é o objeto que você deseja, não faça nada. Caso contrário, especifique o nome corretamente e verifique se você tem a permissão correta para criar o objeto.

GSE0006E O Spatial Extender não pôde executar a operação solicitada em um objeto ativo ou registrado chamado "<nome do objeto>".

Explicação: O objeto "<nome do objeto>" já ativo ou registrado, ou ele já existe. Ele pode ser uma camada, índice, sistema de referência espacial, sistema de coordenadas, geocoder, ou outro tipo de objeto.

Resposta do Usuário: Tenha a certeza de que o objeto "<nome do objeto>" existe e submeta novamente a sua solicitação.

GSE0007E O Spatial Extender não pôde executar a operação solicitada no “<nome do objeto>”, um objeto que ainda não foi ativado ou registrado.

Explicação: O objeto “<nome do objeto>” não foi ativado ou registrado. Ele pode ser uma camada, índice, sistema de referência espacial, sistema de coordenada espacial, geocoder, ou outro tipo de objeto.

Resposta do Usuário: Ative ou registre o objeto “<nome do objeto>”. Em seguida, submeta novamente a sua solicitação.

GSE0008E Ocorreu um erro SQL inesperado (“<mensagem de erro sql>”).

Resposta do Usuário: Procure a mensagem detalhada associada com o SQLCODE na mensagem de erro SQL “<mensagem de erro sql>”. Se necessário, entre em contato com o representante de serviços IBM.

GSE0009E A operação solicitada não pôde ser executada em um objeto chamado “<nome do objeto>” que já existe.

Explicação: O “<nome do objeto>” já existe no banco de dados ou no sistema operacional. Ele pode ser um arquivo, tabela, exibição, coluna, índice, disparador ou outro tipo de objeto.

Resposta do Usuário: Assegure-se de que tenha especificado o objeto corretamente ao tentar acessá-lo. Se necessário, exclua o objeto.

GSE0010E A operação solicitada não pôde ser executada em um objeto chamado “<nome do objeto>” que pode não existir.

Explicação: O “<nome do objeto>” não existe no banco de dados ou no sistema operacional. Ele pode ser um arquivo, tabela, exibição, coluna, índice, disparador ou outro tipo de objeto.

Resposta do Usuário: Assegure-se de ter a permissão correta para acessar o objeto. Caso

tenha esta permissão e o objeto não existe, então você precisa criá-lo.

GSE0011E O Spatial Extender não pôde desativar ou remover o registro do objeto “<nome do objeto>”.

Explicação: O “<nome do objeto>” é dependente de outro objeto. O “<nome do objeto>” pode ser um sistema de referência espacial, camada, geocoder ou outro tipo de objeto.

Resposta do Usuário: Consulte a documentação para localizar quais tipos de objetos “<nome do objeto>” podem ser dependentes. Em seguida remova o objeto específico “<nome do objeto>” que é dependente.

GSE0012E O Spatial Extender não pôde processar sua solicitação porque a coluna espacial completamente qualificada “<esquema da camada.nome da camada.coluna da camada>” não está registrada como uma camada de tabela.

Explicação: A coluna espacial completamente qualificada “<esquema da camada.nome da camada.coluna da camada>” deve estar registrada como uma camada de tabela antes que você possa executar certas operações associadas com a mesma (por exemplo, ativar seu índice, ativar um geocoder para preenchê-la no modo batch ou para atualizá-la automaticamente).

Resposta do Usuário: Assegure-se de que a coluna espacial completamente qualificada “<esquema da camada.nome da camada.coluna da camada>” esteja registrada como camada de tabela verificando a exibição DB2GSE.GEOMETRY_COLUMNS no catálogo do Spatial Extender. Também esteja certo de que a tabela que contém esta coluna também inclui colunas de atributo correspondente válido.

GSE0013E O banco de dados não está ativados para operações espaciais.

Explicação: O banco de dados não está ativado para operações espaciais. Portanto, o catálogo do

Spatial Extender não existe.

Resposta do Usuário: Ativar o banco de dados para operações espaciais.

GSE0014E O banco de dados foi ativado para operações espaciais.

Explicação: O banco de dados foi ativado para operações espaciais.

Resposta do Usuário: Verificar se o banco de dados foi ativado como esperado. Se necessário, desabilite o banco de dados.

GSE0498E Ocorreu o seguinte erro:
"<mensagem de erro>".

GSE0499W O Spatial Extender emitiu o seguinte aviso: "<mensagem de erro>".

GSE0500E O modo de operação que você especificou ("<modo de operação>") não é válido.

Explicação: O modo especificado não é suportado pela operação que você solicitou.

Resposta do Usuário: Consulte a documentação para localizar quais modos são suportados pela operação.

GSE1001E O Spatial Extender não pôde registrar uma camada de exibição que é chamada "<nome do esquema.nome da view.nome da coluna>" e que está baseada na coluna espacial "<nome do esquema.nome da tabela.nome da coluna>".

Explicação: A coluna espacial que você especificou ("<nome do esquema.nome da tabela.nome da coluna>") não foi registrada como uma camada de tabela.

Resposta do Usuário: Registre a coluna "<nome do esquema.nome da tabela.nome da coluna>" como uma camada de tabela.

GSE1002E O Spatial Extender foi incapaz de registrar uma camada de view que é chamada "<nome do esquema.nome da view.nome da coluna>" e que está baseada na tabela "<nome do esquema.nome da tabela>".

Explicação: A tabela que você especificou ("<nome do esquema.nome da tabela>") não suporta a view "<nome do esquema.nome da view.nome da coluna>", direta ou indiretamente.

Resposta do Usuário: Localize qual tabela base é para a view "<nome do esquema.nome da view.nome da coluna>", e especifique-a.

GSE1003E O Spatial Extender foi incapaz de acessar uma coluna chamada "<nome da coluna>" em uma tabela ou view chamada "<nome do esquema.nome do objeto>".

Explicação: A tabela ou view "<nome do esquema.nome do objeto>" não possui uma coluna chamada "<nome da coluna>".

Resposta do Usuário: Verifique a definição da tabela ou view "<nome do esquema.nome do objeto>" para localizar o nome próprio da coluna que você deseja.

GSE1004E O Spatial Extender foi incapaz de registrar a coluna espacial completamente qualificada "<nome do esquema.nome da tabela.nome da coluna>" como uma camada de tabela.

Explicação: A coluna "<nome do esquema.nome da tabela.nome da coluna>" não possui um tipo de dados espacial, ou não está associada a uma tabela base.

Resposta do Usuário: Defina um tipo de dados espacial para a coluna "<nome do esquema.nome da tabela.nome da coluna>", ou assegure-se de que esta coluna seja parte de uma tabela base local.

GSE1005E O sistema de referência espacial (“<id de referência espacial da camada da view>”) que você especificou para uma camada de view difere do sistema de referência espacial (“<id de referência espacial da camada da view>”) que é utilizado para esta camada de tabela de suporte de camadas.

Explicação: Um sistema de referência espacial de camadas de view deve ser o mesmo que o sistema de referência espacial de camadas de tabela de suporte.

Resposta do Usuário: Especifique o sistema de referência espacial da camada de tabela de suporte para a camada da view.

GSE1006E Porque a “<id de referência espacial>” é uma ID de sistema de referência espacial, o Spatial Extender foi incapaz de registrar a camada que você solicitou.

Explicação: O sistema de referência espacial que você especificou (“<id de referência espacial>”) não foi ativado ou registrado.

Resposta do Usuário: Ativar ou registrar o sistema de referência espacial. Então, submeta novamente sua solicitação para registrar a camada.

GSE1007E Um erro SQL (SQLSTATE “<sqlstate>”) pode ter ocorrido quando o Spatial Extender tentou sem sucesso incluir uma coluna espacial (“<nome da coluna>”) para a tabela “<nome do esquema.nome da tabela>”.

Resposta do Usuário: Procure a mensagem associada com o SQLSTATE “<sqlstate>”.

GSE1008E O Spatial Extender foi incapaz de registrar uma camada de view “<esquema da camada.nome da camada.coluna da camada>” porque o tipo de dados espacial “<tipo de coluna da camada>” da camada da view não coincide com o tipo de dados espacial “<tipo de coluna geométrica>” da camada de tabela suportada “<esquema de geometria.nome da geometria.coluna de geometria>”.

Explicação: O tipo de dados espacial de uma camada de view “<esquema da camada.nome da camada.coluna da camada>” deve coincidir com o tipo de dados espacial da camada de tabela que suporta a camada “<esquema de geometria.nome de geometria.coluna de geometria>”. A inconsistência entre estes dois tipos de dados causa ambigüidade quando os dados espaciais são processados.

Resposta do Usuário: Assegure-se de que os tipos de dados espaciais da camada de view e suas camadas de tabelas suportadas sejam as mesmas.

GSE1020E A “<id de referência espacial>” é uma ID de sistema de referência espacial inválida.

Explicação: Um sistema de referência espacial com um identificador de “<id de referência espacial>” não foi ativado.

Resposta do Usuário: Assegure-se de que a referência espacial especificada foi ativada.

GSE1021E O Spatial Extender não pôde ativar o sistema de referência espacial “<id de referência espacial>” porque a correspondente ID do sistema de coordenada espacial “<id de coordenada espacial>” era inválida.

Explicação: Um sistema de coordenada com um identificador de “<id de coordenada espacial>” não está definido no catálogo Spatial Extender.

Resposta do Usuário: Verifique o identificador do sistema de coordenada “<id de coordenada espacial>” conferindo a view DB2GSE.COORD_REF_SYS no catálogo do Spatial Extender.

GSE1030E Porque a “<nome do esquema.nome da tabela>” não é uma tabela base, o Spatial Extender não pôde ativar um geocoder para a mesma.

Explicação: O objeto que contém os dados fonte que você deseja geocodificar deve ser uma tabela base.

Resposta do Usuário: Esteja seguro de que as colunas que contém os dados fonte que você deseja geocodificar sejam parte de uma tabela base.

GSE1031E O Spatial Extender não pôde ativar o geocoder “<id do geocoder>” para operar automaticamente no modo criar para a camada “<esquema da camada.nome da camada.coluna da camada>”.

Explicação: As possíveis explicações são:

- O geocoder já está ativo para atualizar a camada “<esquema da camada.nome da camada.coluna da camada>” automaticamente.
- O geocoder foi temporariamente invalidado para esta camada.
- Nenhuma coluna para dados fonte foi definida para esta camada.

Resposta do Usuário: Se o geocoder foi temporariamente invalidado, ative-o para operar automaticamente no modo “Recriar”.

GSE1032E O Spatial Extender não pôde ativar o geocoder “<id do geocoder>” para operar automaticamente no modo re-criar para a camada “<esquema da camada.nome da camada.coluna da camada>”.

Explicação: As possíveis explicações são:

- O geocoder já está ativo para atualizar a camada “<esquema da camada.nome da camada.coluna da camada>” automaticamente.
- O geocoder não foi invalidado previamente para esta camada.
- Nenhuma coluna para dados fonte foi definida para esta camada.

Resposta do Usuário: Se o geocoder foi desativado previamente no modo eliminar, ou se ele nunca foi definido para esta camada, ative-o para operar automaticamente no modo “Criar”.

GSE1033E Ocorreu um erro SQL quando o Spatial Extender tentou incluir disparadores para uma tabela que contém a coluna para a camada “<esquema da camada.nome da camada.coluna da camada>” (SQLSTATE “<sqlstate>”).

Explicação: O propósito dos disparadores é manter a integridade dos dados entre as colunas de atributos de onde sucede a entrada do geocoder e a coluna espacial que suas saídas vão. O erro SQL ocorreu quando o DB2 tentou criar estes disparadores.

Resposta do Usuário: Procure a mensagem associada com o SQLSTATE “<sqlstate>”.

GSE1034E O Spatial Extender não pôde desativar o geocoder “<id do geocoder>” no modo eliminar para a camada “<esquema da camada.nome da camada.coluna da camada>”.

Explicação: As possíveis explicações são:

- O geocoder nunca foi ativado para atualizar a camada “<esquema da camada.nome da camada.coluna da camada>” automaticamente.
- O geocoder foi desativado no modo eliminar.

Resposta do Usuário: Determine o estado do geocoder antes de tentar desativá-lo. Por exemplo, ele foi registrado? foi ativado? Então decida se ele precisa ser desativado no modo eliminar. Por exemplo, caso nunca tenha sido ativado, talvez não seja preciso desativá-lo por completo.

GSE1035E O Spatial Extender não pôde desativar o geocoder “<id do geocoder>” no modo invalidar para a camada “<esquema da camada.nome da camada.coluna da camada>”.

Explicação: As possíveis explicações são:

- O geocoder nunca foi ativado para atualizar a camada “<esquema da camada.nome da camada.coluna da camada>” automaticamente.
- O geocoder foi desativado no modo invalidado ou no modo eliminar.

Resposta do Usuário: Determine o estado do geocoder antes de tentar desativá-lo. Por exemplo, ele foi registrado? Foi ativado? Então decida se ele precisa ser desativado no modo invalidar. Por exemplo, caso já tenha sido desativado no modo invalidar, talvez não seja preciso desativá-lo neste modo uma segunda vez.

GSE1036E Ocorreu um erro SQL quando o Spatial Extender tentou eliminar disparadores de uma tabela que contém a coluna para a camada “<esquema da camada.nome da camada.coluna da camada>” (SQLSTATE “<sqlstate>”).

Explicação: Os disparadores foram criados para manter a integridade dos dados entre as colunas de atributos de onde sucede a entrada do geocoder e a coluna espacial para onde as suas saídas vão. O erro SQL ocorrido quando o DB2 tentou eliminar estes disparadores.

Resposta do Usuário: Procure a mensagem

associada com o SQLSTATE “<sqlstate>”.

GSE1037E O Spatial Extender não pôde fazer o geocode dos dados fonte para a camada de tabela “<esquema da camada.nome da camada.coluna da camada>”, possivelmente porque um valor incorreto “<número de atributos>” foi assinalado para o argumento que especifica quantas são as colunas de atributo para fornecer dados fonte para esta camada/

Explicação: O número de colunas de atributo associadas a esta camada foi especificado incorretamente, ou o nome de uma ou mais destas colunas foi especificado incorretamente.

Resposta do Usuário: Assegure-se de que esta camada esteja registrada com o número correto e com os nomes das colunas de atributo associadas, ou verifique falta de exatidão dos dados de entrada e saída para o geocoder.

GSE1038E Ocorreu um erro SQL quando o Spatial Extender tentou fazer o geocode dos dados fonte para a camada de tabela “<esquema da camada.nome da camada.coluna da camada>” no modo batch (SQLSTATE “<sqlstate>”).

Resposta do Usuário:

- Procure a mensagem associada com o SQLSTATE “<sqlstate>”.
- Assegure-se de que o conteúdo e o argumento primaryUDF desta camada estejam definidos corretamente.

GSE1050E O tamanho da grade que você especificou “<tamanho da grade>”) é inválido para o primeiro nível da grade.

Explicação: Você especificou zero ou um número negativo como tamanho da grade para o primeiro nível da grade.

Resposta do Usuário: Especificar um número

positivo como tamanho da grade.

GSE1051E O tamanho da grade que você especificou “<tamanho da grade>”) é inválido para o segundo e terceiro níveis da grade.

Explicação: Você especificou um número negativo como tamanho da grade para o segundo ou o terceiro nível de grade.

Resposta do Usuário: Especificar zero ou um número positivo como o tamanho da grade.

GSE1052E Ocorreu um erro SQL quando o Spatial Extender tentou criar o índice espacial “<esquema do índice.coluna do índice>” para uma camada de tabela “<esquema da camada.nome da camada.coluna da camada>” (SQLSTATE “<sqlstate>”).

Resposta do Usuário:

- Assegure-se de que o índice espacial esteja especificado corretamente e que a coluna espacial não possui índice associado.
- Procure a mensagem que é associada ao SQLSTATE “<sqlstate>”.

GSE1500I O registro fonte “<número do registro>” foi geocodificado com sucesso.

Explicação: Um registro contendo dados de atributo foi geocodificado com sucesso.

GSE1501W O registro fonte “<número do registro>” não foi geocodificado.

Explicação: O nível de precisão foi muito alto.

Resposta do Usuário: Faça o geocode com um nível de precisão menor.

GSE1502W O registro fonte “<número do registro>” não foi encontrado.

Resposta do Usuário: Determine se o registro existe no banco de dados.

GSE2001E O Spatial Extender não pôde executar a operação pedida.

Resposta do Usuário: Consulte o administrador do banco de dados.

GSE2002E Ocorreu um erro de sistema de gerenciamento de banco de dados.

Resposta do Usuário: Consulte o administrador do banco de dados.

GSE2003E O procedimento armazenado que você invocou não conseguiu conectar-se a sua estação de trabalho.

Explicação: O procedimento armazenado não pode acessar as informações que identificam sua estação de trabalho junto a ele.

Resposta do Usuário: Consulte o administrador do banco de dados.

GSE2004E O Spatial Extender não pode validar o identificador de sistema de coordenadas especificado no arquivo de transferência SDE que está sendo importado.

Resposta do Usuário: Tente um ou mais dos seguintes métodos:

- Certifique-se de que o identificador do sistema de referências espaciais, no arquivo de transferência SDE, esteja apontando para o identificador de sistema de coordenadas correto.
- Verifique se o identificador de sistema de coordenadas correto está listado na exibição de catálogo DB2GSE.COORD_REF_SYS. Se o identificador não estiver nessa exibição, comunique o fato ao administrador do banco de dados.

- Verifique se o arquivo de transferência SDE está danificado. Se estiver, procure adquirir e importar uma cópia intacta.

GSE2005E O Spatial Extender não pode validar o arquivo que você deseja exportar.

Explicação: Pode haver uma ou mais razões para esse problema. Por exemplo, talvez você não esteja autorizado a acessar o arquivo. Ou, o Spatial Extender pode não ser capaz de encontrar ou ler o arquivo ou de reconhecer os tipos de dados que o compõem.

Resposta do Usuário: Certifique-se de especificar o caminho completo do arquivo. Garanta também que o ID do usuário com o qual o procedimento armazenado `db2gse.gse_export_shape` é executado tenha acesso, tanto de leitura quanto de gravação, a cada diretório no caminho. Verifique se o disco que contém esses diretórios está montado no mesmo nó em que o DB2 é executado e se o mesmo ponto de montagem que está especificado no caminho é utilizado. Verifique também se o Spatial Extender reconhece os tipos de dados que o arquivo contém.

Se o erro se repetir, verifique se o arquivo está danificado. Se estiver, adquira e exporte uma cópia intacta do arquivo.

GSE2006E Ocorreu um erro de E/S para um arquivo chamado “<nome do arquivo>”.

Resposta do Usuário: Verifique se o arquivo existe, se você possui o acesso apropriado para o arquivo e se o arquivo não está sendo utilizado por um outro usuário.

GSE2007E O Spatial Extender não pode validar a camada para a qual você deseja importar dados.

Explicação: O nome da coluna na qual essa camada está definida talvez tenha sido especificado incorretamente ou não esteja de acordo com as convenções de nomenclatura padrão. Igualmente, o nome da tabela à qual a

coluna pertence pode ter sido especificado incorretamente ou não está de acordo com as convenções de nomenclatura padrão.

Resposta do Usuário: Certifique-se de que a camada esteja listada na exibição de catálogo `DB2GSE.GEOMETRY_COLUMNS`, de que os nomes da coluna e da tabela aos quais a camada pertence estejam especificados corretamente e de que esses nomes estejam em conformidade com as convenções de nomenclatura padrão.

GSE2008E O Spatial Extender tentou inserir um valor nulo em uma camada que tem uma limitação NOT NULL.

Resposta do Usuário: Importe a coluna que contém valores nulos para uma camada que possa aceitar esses valores ou peça ao administrador que remova a limitação NOT NULL.

GSE2012E O Spatial Extender não pôde acessar a camada para a qual você deseja importar os dados.

Explicação: O ID do usuário com o qual você deseja acessar a camada não está autorizado a alterar a coluna na qual a camada está definida.

Resposta do Usuário: Peça a autorização necessária ao administrador do banco de dados (por exemplo, você pode precisar do privilégio de inserção, INSERT, ou de seleção, SELECT, para a tabela à qual a coluna pertence).

GSE2014E O Spatial Extender não pôde importar dados para a camada especificada ou exportar dados dela.

Explicação: O Spatial Extender não pôde localizar a camada para a importação ou exportação de dados.

Resposta do Usuário: Verifique se a camada está listada na exibição `DB2GSE.GEOMETRY_COLUMNS`. Se não estiver, use o procedimento armazenado `db2gse.gse_register_layer` ou a janela Criar

Camada, no Centro de Controle, para registrar a camada. Se a camada estiver listada em DB2GSE.GEOMETRY_COLUMNS, informe o problema ao administrador do banco de dados.

GSE2016E O Spatial Extender não pôde importar o arquivo de formato que você pediu para a camada especificada.

Explicação: O tipo de dados que você deseja importar é incompatível com o tipo de dados da camada para a qual esses dados espaciais se destinam.

Resposta do Usuário: Crie uma nova camada cujo tipo de dados seja compatível com o tipo dos dados espaciais que você deseja importar. Em seguida, importe os dados para essa nova camada. Como alternativa, importe um arquivo de formato de dados diferente, cujos dados espaciais sejam compatíveis com a camada que você deseja preencher.

GSE2021E O Spatial Extender não pôde acessar o arquivo de formato que você deseja importar.

Explicação: Há diversos motivos possíveis para esse problema. Por exemplo, o Spatial Extender talvez não conheça o caminho completo para o arquivo de formato ou não reconheça o formato do arquivo, ou o disco que contém o arquivo provavelmente não foi montado adequadamente.

Resposta do Usuário: Certifique-se de especificar o caminho completo do arquivo. Se fizer isso e o erro se repetir, verifique se o arquivo é realmente um arquivo de formato, e não outro tipo de arquivo que tenha sido especificado erroneamente como um arquivo de formato. Se for um arquivo de formato, tente uma das seguintes providências:

- Verifique se o arquivo está danificado. Se estiver, adquira e importe uma cópia intacta do arquivo.
- Se estiver acessando o arquivo em outra estação de trabalho, verifique se:
 - O disco que contém o arquivo está montado.

- O disco usa o mesmo ponto de montagem especificado no pathname do arquivo.
- O ID do usuário que você está usando em outra estação de trabalho tem acesso de leitura ao arquivo.

GSE2022E O identificador do sistema de referências espaciais não existe.

Explicação: O identificador do sistema de referências espaciais (SRID) especificado para o arquivo de formato que você deseja importar não está listado no catálogo do Spatial Extender.

Resposta do Usuário: Proceda de uma das seguintes maneiras:

- Atribua ao arquivo de formato um sistema de referências espaciais compatível, cujo SRID já esteja registrado na exibição de catálogo DB2GSE.SPATIAL_REF_SYS.
- Como alternativa, crie um novo sistema de referências espaciais para o arquivo de formato. Em seguida, verifique se o SRID do sistema foi registrado no catálogo do Spatial Extender.

Essa mensagem não está no catálogo de mensagens do DB2 Spatial Extender. Quando os erros aos quais essa mensagem está associada ocorrerem, você será notificado online de que eles não podem ser recuperados.

GSE2023E O Spatial Extender não pôde importar os dados de atributos do arquivo de formato especificado.

Explicação: A definição de uma coluna de atributos no arquivo de formato não pôde ser convertida numa definição para uma coluna correspondente na tabela para a qual você deseja importar dados.

Resposta do Usuário: Certifique-se de que o tipo de dados, o comprimento máximo e outras características dessa coluna de atributos possam ser convertidos em equivalentes ou contrapartes da coluna de atributos para a qual você está importando dados.

GSE2026E O Spatial Extender não pôde criar um arquivo para conter os dados que não conseguiu importar.

Explicação: Ao importar um arquivo de formato, o Spatial Extender coleta todos os registros desse arquivo que não puderam ser importados, para que possam ser corrigidos e importados posteriormente. Nesse caso, o Spatial Extender não dispunha de informações suficientes ou de autorização para criar um arquivo para conter os registros rejeitados.

Resposta do Usuário: Especifique um caminho completo para o arquivo que o Spatial Extender está prestes a criar para os registros rejeitados. Verifique se já não existe um arquivo com o mesmo nome e o mesmo caminho. Verifique também se o ID do usuário com o qual você está executando o procedimento armazenado db2gse.gse_import_shape tem acesso de leitura e de gravação a cada diretório no caminho. Verifique se o disco que contém esses diretórios está montado no mesmo nó em que o DB2 está sendo executado e se está usando o mesmo ponto de montagem especificado no caminho.

GSE2027E O Spatial Extender não pôde executar a operação de importação ou exportação perdida.

Explicação: Não há memória suficiente para concluir a operação. O arquivo que você está importando ou exportando pode estar danificado, provocando uma drenagem excessiva de memória.

Resposta do Usuário: Tente importar ou exportar o arquivo novamente. Se o erro continuar a ocorrer, verifique se o arquivo está danificado. Se estiver, adquira uma cópia intacta do arquivo e importe ou exporte essa cópia. Se o problema persistir, informe o administrador do banco de dados.

GSE2030E O Spatial Extender não pode importar dados para a coluna especificada.

Explicação: Essa coluna especificada não estava registrada como uma camada.

Resposta do Usuário: Se desejar importar dados SDE, use o Centro de Controle do DB2 ou o procedimento armazenado db2gse.gse_import_sde para registrar a coluna como uma camada e importar os dados. Se desejar importar dados de formato, utilize o Centro de Controle ou o procedimento armazenado db2gse.gse_import_shape para registrar a coluna como uma camada e importar os dados.

GSE2031E O Spatial Extender não pôde importar dados para a camada especificada.

Explicação: A tabela na qual a camada estava definida não existe mais.

Resposta do Usuário: Se desejar importar dados SDE, use o Centro de Controle do DB2 ou o procedimento armazenado db2gse.gse_import_sde para recriar a tabela e importar os dados. Se desejar importar dados de formato, utilize o Centro de Controle ou o procedimento armazenado db2gse.gse_import_shape para criar a tabela novamente e importar os dados.

GSE2032E O Spatial Extender não pôde importar ou exportar dados de atributos.

Explicação: Uma explicação possível é que você está tentando importar dados de atributos para uma tabela, mas uma ou mais colunas de atributos especificadas no seu arquivo de importação não possuem contrapartes nessa tabela. Outra explicação é que você está tentando exportar dados de uma tabela ou exibição, mas uma ou mais colunas de atributos nessa tabela não possuem contrapartes no arquivo para o qual deseja exportar os dados.

Resposta do Usuário: Se estiver importando dados de atributos, identifique a coluna (ou cada uma das colunas) no arquivo de importação que não possua coluna correspondente na tabela na qual o arquivo deve ser carregado. Em seguida, forneça a tabela com a coluna (ou colunas) ausente. Como alternativa, você pode alterar o destino da importação como uma camada e

definir colunas de atributos diferentes dos pretendidos originalmente.

Se estiver exportando dados de atributos de uma tabela ou exibição, identifique cada coluna de tabela ou exibição que não tenha uma coluna correspondente no arquivo de exportação. Em seguida, forneça o arquivo de importação com a coluna (ou colunas) ausente ou forneça um novo arquivo de exportação — que possua uma coluna para cada coluna de dados desejada para exportação.

GSE2033E O Spatial Extender não pôde ler totalmente o arquivo que você deseja importar.

Explicação: O arquivo pode estar danificado ou truncado.

Resposta do Usuário: Tente importá-lo novamente. Se o erro se repetir, adquira e importe uma cópia intacta do arquivo.

GSE2034E O Spatial Extender não pôde importar o arquivo de transferência SDE que você pediu.

Explicação: O tipo de dados que você deseja importar é incompatível com o tipo de dados da camada para a qual esses dados espaciais se destinam.

Resposta do Usuário: Crie uma nova camada cujo tipo de dados seja compatível com o tipo dos dados espaciais que você deseja importar. Em seguida, importe os dados para essa nova camada. Como alternativa, importe um arquivo

de transferência SDE diferente, cujos dados espaciais sejam compatíveis com a camada que você deseja preencher.

GSE2035E A coordenada especificada está fora dos limites.

Explicação: O Spatial Extender encontrou uma coordenada que é muito grande ou muito pequena para caber nos limites do sistema de coordenadas atribuído à camada especificada. Esse problema pode ser provocado pela incompatibilidade do sistema de coordenadas com os dados da camada ou com o respectivo sistema de referências espaciais. Pode também decorrer de um arquivo de transferência SDE ou de formato danificado ou, ainda, de dados anômalos inseridos na camada por engano.

Resposta do Usuário: Verifique se o identificador correto foi especificado para o sistema de referências espaciais utilizado pela camada específica. Caso positivo, esse sistema poderá ser incompatível com os dados da camada ou com o sistema de coordenadas subjacente. Em virtude dessas possibilidades, selecione ou crie um sistema de referências espaciais diferente para a camada. Se o problema se repetir, relate-o ao administrador do banco de dados.

Essa mensagem não está no catálogo de mensagens do DB2 Spatial Extender. Quando os erros aos quais essa mensagem está associada ocorrerem, você será notificado online de que eles não podem ser recuperados.

Mensagens retornadas pelas funções espaciais

Os códigos SQLCODEs das mensagens retornadas pelas funções espaciais começam com as letras "GSE", seguidos de números no intervalo de 3.001 a 3.042.

Quando uma mensagem é retornada por uma função espacial, seu valor SQLSTATE associado também é retornado, mas não seu código SQLCODE. Para obter informações sobre como encontrar o código SQLCODE, consulte a página

Note to reviewers:: Cross-reference will be inserted here.

GSE3001E Falha desconhecida do sistema.

Explicação: Ocorreu um erro inesperado.

Resposta do Usuário: Corrija a sintaxe e invoque a função novamente. Se o problema ainda for observado, entre em contato com o suporte técnico.

GSE3002E Cadeia de Texto Convencional inválida

Explicação: Uma cadeia de Texto Convencional inválida foi inserida como uma entrada para a função invocada.

Resposta do Usuário: Corrija a cadeia e invoque a função novamente. Para determinar o formato adequado para cadeias de Texto Convencional, consulte Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3003E SRID inválido

Explicação: O identificador do sistema de referências espaciais (SRID) que você tentou passar para essa função não está listado no catálogo do sistema DB2 Spatial Extender.

Resposta do Usuário: Especifique um SRID que esteja registrado atualmente na exibição de catálogo DB2GSE.SPATIAL_REF_SYS ou crie um sistema de referências espaciais que tenha o SRID que você deseja especificar.

GSE3004E Memória insuficiente

Explicação: Não há memória disponível suficiente. O DB2 Spatial Extender requer no máximo um megabyte de memória.

Resposta do Usuário: Realoque a memória para que haja maior disponibilidade para o DB2 Spatial Extender. Se não for possível realocar a memória, adicione mais memória física.

GSE3005E Os SRIDs geométricos são diferentes.

Explicação: As figuras geométricas passadas para uma função do DB2 Spatial Extender devem compartilhar o mesmo identificador do sistema

de referências espaciais (SRID).

Resposta do Usuário: Crie novamente uma das figuras geométricas para que o seu sistema de referências espaciais corresponda as do outro.

GSE3006E Cadeia binária inválida.

Explicação: Uma cadeia Binária Convencional ou Binária ESRI foi inserida como uma entrada para a função que você invocou.

Resposta do Usuário: Reconstrua a cadeia com o formato correto. Para determinar o formato correto, consulte Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3007E Figura geométrica válida não especificada.

Explicação: Não foi passado um tipo válido de figura geométrica para a função que você invocou. Os tipos válidos são figura geométrica, ponto, segmento de reta, polígono, pontos múltiplos, segmentos de reta múltiplos e polígonos múltiplos.

Resposta do Usuário: Submeta novamente a instrução SQL com um tipo de figura geométrica válido.

GSE3008E Parênteses díspares.

Explicação: O número de parênteses esquerdos na cadeia de representação de Texto Convencional não é igual ao número de parênteses diretos.

Resposta do Usuário: Insira a cadeia novamente, fornecendo um número de parênteses diretos correspondente aos esquerdos.

GSE3009E Excesso de partes especificadas.

Explicação: O número de partes indicado na cadeia de texto ou binária é maior que o número real de partes fornecidas.

Resposta do Usuário: Insira a cadeia novamente com o número correto de partes.

GSE3010E Tipo de figura geométrica incorreto.

Explicação: O tipo incorreto de figura geométrica foi passado para a função que você invocou. Por exemplo, um segmento de reta pode ter sido passado para uma função que recebe polígonos como uma entrada.

Resposta do Usuário: Passe para a função um tipo de figura geométrica que a função possa processar ou utilize uma função que aceite o tipo de figura geométrica que você deseja passar.

GSE3011E A cadeia de texto é muito longa.

Explicação: A cadeia de texto de figura geométrica excede o comprimento máximo de 4.000 caracteres.

Resposta do Usuário: A figura geométrica contém detalhes em excesso para ser convertida em texto. No entanto, você tem a alternativa de convertê-la num formato WKB ou num formato binário de formato ESRI.

GSE3012E Valor de parâmetro inválido.

Explicação: Um parâmetro inválido foi passado para a função.

Resposta do Usuário: Compare a sintaxe da função com a listada no Manual e Referência do Usuário do DB2 Spatial Extender. Corrija o parâmetro inválido e submeta a função novamente.

GSE3013E Tamanho de grade inválido.

Explicação: Uma das seguintes especificações inválidas foi feita:

- Especificou-se um número negativo como o tamanho da grade para o primeiro, segundo e terceiro nível da grade.
- Especificou-se zero como o tamanho para o primeiro nível da grade.
- O tamanho da grade especificado para o segundo nível da grade é menor do que o tamanho do segundo nível da grade.

- O tamanho especificado para o terceiro nível da grade é menor do que o tamanho do segundo nível da grade.

Resposta do Usuário: Utilize a janela Criar Índice ou o procedimento armazenado db2gse.gse_enable_idx para especificar um tamanho de grade válido. Para obter informações sobre tamanhos de grades válidos, consulte Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3014E Tamanho da grade muito pequeno.

Explicação: O tamanho da grade que foi especificado resulta em mais de 1.000 células de grade por figura geométrica.

Resposta do Usuário: Utilize a janela Criar Índice ou o procedimento armazenado db2gse.gse_enable_idx para aumentar o tamanho da grade ou adicionar outro nível de grade.

GSE3015E Figura geométrica inválida produzida.

Explicação: Os parâmetros inseridos produziram uma figura geométrica inválida. Por exemplo, os parâmetros inseridos com a função LineFromShape produzem uma figura geométrica inválida. Figura geométrica inválida é a que viola uma propriedade geométrica.

Resposta do Usuário: Corrija o parâmetro e submeta a figura geométrica novamente.

GSE3016E Erro ao submeter figura geométrica.

Explicação: A função aguardava duas figuras geométricas de um certo tipo e não as recebeu. Por exemplo, a ST_Union esperava duas figuras geométricas da mesma dimensão e recebeu um ponto e um segmento de reta, que têm dimensões diferentes.

Resposta do Usuário: Especifique figuras geométricas aceitas pela função como entradas válidas. Para determinar quais tipos de figuras geométricas são válidos para essa função,

consulte Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3017E Erro de integridade de figura geométrica.

Explicação: A função não pode processar a figura geométrica que lhe foi passada, porque uma ou mais propriedades da figura geométrica violam uma limitação de integridade.

Resposta do Usuário: Submeta a figura geométrica novamente com suas propriedades definidas de modo correto. Para obter informações sobre as propriedades das figuras geométricas, consulte o Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3018E Pontos em excesso.

Explicação: A construção de uma figura geométrica excedeu o limite de armazenamento de 1 MB; a figura geométrica tem pontos em excesso.

Resposta do Usuário: Remova os pontos desnecessários. Por considerações de desempenho e armazenamento, experimente incluir apenas os pontos necessários para produzir uma figura geométrica. Todos os pontos não-essenciais devem ser excluídos.

GSE3019E A figura geométrica é muito pequena.

Explicação: A figura geométrica retornada pela função ST_Difference, ST_Intersection, ST_SymmetricDiff, ou ST_Union é muito pequena para ser representada pelos valores do sistema de coordenadas atual.

Resposta do Usuário: Se um resultado for necessário, utilize o procedimento armazenado db2gse.gse_enable_sref para incrementar o parâmetro xyunits do sistema de referências espaciais da figura geométrica de origem. Em seguida, crie novamente a tabela na qual a figura geométrica de origem está armazenada.

GSE3020E Buffer fora dos limites.

Explicação: A função de buffer criou um buffer fora do sistema de coordenadas.

Resposta do Usuário: Reduza a distância do buffer ou altere o sistema de coordenadas da figura geométrica. Na maioria dos casos, a alteração do sistema de coordenadas requer a recarga do sistema espacial.

GSE3021E Fator de escala inválido.

Explicação: Um fator de escala (uma unidade XY, uma unidade Z ou uma unidade M) não pode ser inferior a 1.

Resposta do Usuário: Utilize o procedimento armazenado db2gse.gse_enable_sref para corrigir todos os fatores de escala na exibição de catálogo DB2GSE.SPATIAL_REF_SYS que sejam inferiores a 1.

GSE3022E Coordenada fora dos limites.

Explicação: Uma coordenada é muito grande ou muito pequena para caber nos limites do sistema de coordenadas.

Resposta do Usuário: Verifique se a coordenada está correta. Se estiver, verifique se cabe nos limites do sistema de coordenadas que você está utilizando. Para obter informações sobre esse sistema de coordenadas, consulte a exibição de catálogo DB2GSE.COORD_REF_SYS.

GSE3023E ID do sistema de coordenadas inválido.

Explicação: O Spatial Extender não pode validar o identificador do sistema de coordenadas especificado.

Resposta do Usuário: Verifique se o identificador está listado na exibição de catálogo DB2GSE.COORD_REF_SYS. Se não estiver, verifique se está correto e peça ao administrador do banco de dados que o registre no catálogo do sistema Spatial Extender.

GSE3024E Texto de anotação inválido.

Explicação: O texto de anotação que define o sistema de coordenadas especificado não pode ser convertido numa projeção válida.

Resposta do Usuário: Observe o texto de anotação para esse sistema de coordenadas na exibição de catálogo DB2GSE.COORD_REF_SYS. Verifique se o texto define o sistema adequadamente. Para obter informações de ajuda, consulte o capítulo sobre sistemas de coordenadas em Manual e Referência do Usuário do DB2 Spatial Extender.

GSE3025E Erro de Projeção

Explicação: Ocorreu um erro durante uma tentativa de projetar uma figura geométrica.

Resposta do Usuário: Certifique-se de que a figura geométrica esteja dentro do domínio legal da projeção.

GSE3026E Contornos do polígono sobrepostos.

Explicação: Os contornos de um polígono não podem se sobrepor, mas podem fazer interseção tangencial.

Resposta do Usuário: Corrija as coordenadas do polígono e o submeta novamente.

GSE3027E Muito poucos pontos.

Explicação: Um segmento de reta deve consistir em pelo menos dois pontos e os polígonos devem consistir em pelo menos quatro pontos.

Resposta do Usuário: Submeta a figura geométrica novamente com o número de pontos correto.

GSE3028E O polígono não está fechado.

Explicação: As coordenadas dos pontos inicial e final do polígono não são as mesmas.

Resposta do Usuário: Edite a lista de coordenadas do polígono, certificando-se de que

os pontos inicial e final sejam iguais, e submeta-o novamente.

GSE3029E Contorno exterior inválido.

Explicação: O contorno exterior não contém o contorno interior. O contorno interior está completamente fora do contorno exterior, sem nenhuma sobreposição.

Resposta do Usuário: Certifique-se de que as coordenadas do contorno interior estejam completamente dentro do contorno exterior. Se o contorno interior realmente representar o contorno exterior de outro polígono, insira a figura geométrica como um polígono múltiplo.

GSE3030E O polígono não tem área.

Explicação: Uma figura geométrica é um polígono somente quando suas coordenadas dividem o espaço em duas dimensões.

Resposta do Usuário: Edite as coordenadas do polígono para conterem uma área e submeta o polígono novamente. Ou, se for apropriado, submeta um segmento de reta.

GSE3031E O polígono contém um encavalamento.

Explicação: Somente os pontos inicial e final de um polígono podem ser idênticos. Todas as outras coordenadas de um contorno de polígono devem ser diferentes e, em conjunto, conter uma área.

Resposta do Usuário: Procure pares coordenados que tenham valores de X e Y idênticos. Edite esses pontos de modo que o polígono contenha uma única área e, em seguida, submeta o polígono novamente.

GSE3032E Contornos exteriores sobrepostos.

Explicação: Os contornos exteriores de um polígono múltiplo podem fazer interseção tangencial, mas não podem se sobrepor.

Resposta do Usuário: Edite as coordenadas dos contornos exteriores para que não se sobreponham e submeta o polígono novamente.

GSE3033E O polígono faz interseção com ele mesmo.

Explicação: O contorno de um polígono não pode fazer interseção com ele mesmo.

Resposta do Usuário: Edite as coordenadas do contorno que está fazendo interseção com ele mesmo e submeta o polígono novamente.

GSE3034E Número de medidas inválido.

Explicação: O parâmetro *número de medidas* da cadeia binária contém um número de medidas diferente do fornecido com a cadeia.

Resposta do Usuário: Edite o parâmetro *número de medidas* para que corresponda ao número fornecido na cadeia binária.

GSE3035E Número de partes inválido.

Explicação: O parâmetro *número de partes* da cadeia binária especificou um número de partes diferente do fornecido com a cadeia.

Resposta do Usuário: Edite o parâmetro *número de partes* para que corresponda ao número fornecido na cadeia binária.

GSE3036E Deslocamento de partes inválido.

Explicação: O parâmetro *deslocamento de partes* da cadeia binária especificou um deslocamento de partes diferente do fornecido na cadeia.

Resposta do Usuário: Edite o parâmetro *deslocamento de partes* para que corresponda aos deslocamentos de partes fornecidos com a cadeia binária.

GSE3037E Erro de projeção.

Explicação: Uma figura geométrica ilegal foi encontrada; seu separador de partes é inválido.

Resposta do Usuário: Entre em contato com o representante de serviços IBM.

GSE3038E BLOB muito pequeno.

Explicação: O número de bytes no objeto grande binário (BLOB- binary large object) é

inferior ao número de bytes no BLOB fornecido.

Resposta do Usuário: Torne o comprimento do BLOB igual ao número de bytes no BLOB e submeta a função novamente.

GSE3039E Tipo de entidade inválido.

Explicação: Uma figura geométrica ilegal foi encontrada; seu tipo de entidade associado é inválido.

Resposta do Usuário: Entre em contato com o representante de serviços IBM.

GSE3040E Ordem de bytes inválida

Explicação: A ordem de bytes deve ser 0 ou 1.

Resposta do Usuário: Edite a ordem de bytes para que seja 0 para little endian ou 1 para big endian.

GSE3041E Parte inválida.

Explicação: Um parâmetro de função indexou uma parte inexistente. Por exemplo, esse erro pode ocorrer quando a função ST_GeometryN recebeu um 3 para retornar um terceiro ponto em pontos múltiplos, quando esses contêm apenas dois pontos.

Resposta do Usuário: Edite o parâmetro e submeta a função novamente.

GSE3042E Figura geométrica vazia.

Explicação: Uma figura geométrica vazia foi passada para a função ST_AsBinary, embora não seja permitida como entrada para essa função.

Resposta do Usuário: Edite a instrução SQL que você submeteu para que somente figuras geométricas não-vazias sejam passadas para a função ST_AsBinary. Por exemplo, pode-se usar uma cláusula WHERE para desqualificar figuras geométricas vazias com a função ST_IsEmpty.

Quando uma função espacial retorna uma mensagem, o DB2 exibe a forma abreviada da mensagem e o valor SQLSTATE correspondente dentro da mensagem SQL0443N. Veja um exemplo:

```
DB21034E O comando foi processado como uma instrução SQL, porque não era um
comando válido do Processador de Linha de Comandos. Durante o processamento de SQL foi retornado:
SQL0443N A rotina "DB2GSE.ST_POINTFROMTEX" (nome específico
"SQL000503150228187") retornou um erro SQLSTATE com o texto de diagnóstico
"SRID Inválido". SQLSTATE=38601
```

Para descobrir qual SQLCODE está associado à instrução SQLSTATE retornada na mensagem SQL0443N, consulte Tabela 34. Para ver o texto completo associado ao SQLCODE, consulte esse capítulo ou emita o seguinte comando:

```
DB2 ? [SQLCODE]
```

Tabela 34. Valores SQLSTATE e SQLCODE de mensagens retornadas pelas funções espaciais

| Se o valor de SQLSTATE for: | . . . O SQLCODE será: |
|------------------------------------|------------------------------|
| 38600 | GSE3002E |
| 38601 | GSE3003E |
| 38602 | GSE3004E |
| 38603 | GSE3005E |
| 38604 | GSE3006E |
| 38605 | GSE3007E |
| 38606 | GSE3008E |
| 38607 | GSE3009E |
| 38608 | GSE3010E |
| 38609 | GSE3011E |
| 38610 | GSE3012E |
| 38612 | GSE3013E |
| 38613 | GSE3014E |
| 38800 | GSE3015E |
| 38801 | GSE3016E |
| 38802 | GSE3017E |
| 38803 | GSE3018E |
| 38804 | GSE3019E |
| 38805 | GSE3020E |
| 38806 | GSE3021E |
| 38807 | GSE3022E |
| 38808 | GSE3023E |

Tabela 34. Valores SQLSTATE e SQLCODE de mensagens retornadas pelas funções espaciais (continuação)

| Se o valor de SQLSTATE for: | . . . O SQLCODE será: |
|------------------------------------|------------------------------|
| 38809 | GSE3024E |
| 38810 | GSE3025E |
| 38811 | GSE3026E |
| 38812 | GSE3027E |
| 38813 | GSE3028E |
| 38814 | GSE3029E |
| 38815 | GSE3030E |
| 38816 | GSE3031E |
| 38817 | GSE3032E |
| 38818 | GSE3033E |
| 38819 | GSE3034E |
| 38820 | GSE3035E |
| 38821 | GSE3036E |
| 38822 | GSE3037E |
| 38823 | GSE3038E |
| 38824 | GSE3039E |
| 38825 | GSE3040E |
| 38826 | GSE3041E |
| 38827 | GSE3042E |
| 38999 | GSE3043E |

Capítulo 11. Exibições do catálogo

As exibições do catálogo do Spatial Extender contêm metadados em:

- Sistemas coordenados que você pode utilizar. Para obter mais informações, tais como anotações de texto e identificadores destes sistemas, consulte “DB2GSE.COORD_REF_SYS”.
- Colunas espaciais que foram registradas como camadas. Para obter mais informações, tais como sistemas de referência espacial associados, tipos de dados e nomes destas colunas, consulte “DB2GSE.GEOMETRY_COLUMNS” na página 144.
- Geocoders que você pode utilizar. Para informações tais como identificadores do geocoder e as regiões que contêm as localizações que os geocoders processam, consulte “DB2GSE.SPATIAL_GEOCODER” na página 144.
- Sistemas de referência espacial que podem ser usados. Para informações, tais como seus identificadores e descrições dos mesmos, consulte “DB2GSE.SPATIAL_REF_SYS” na página 145.

DB2GSE.COORD_REF_SYS

Ao ativar um banco de dados para operações espaciais, o Spatial Extender registra os sistemas coordenados que você pode usar em uma tabela de catálogo. As colunas selecionadas desta tabela compreendem a exibição de catálogo DB2GSE.COORD_REF_SYS, que é descrita na Tabela 35.

Tabela 35. Colunas na exibição de catálogo DB2GSE.COORD_REF_SYS

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|-----------|---------------|----------------|---|
| CSID | INTEGER | Não | Identificador numérico inteiro para este sistema coordenado. |
| CS_NAME | VARCHAR(64) | Não | Nome deste sistema coordenado. |
| AUTH_NAME | VARCHAR(256) | Sim | Nome da organização que compilou o sistema coordenado agrega à; por exemplo, o European Petroleum Survey Group (EPSG). |
| AUTH_SRID | INTEGER | Sim | Um identificador numérico designado para este sistema coordenado, pela organização especificada na coluna AUTH_NAME column. |
| DESC | VARCHAR(256) | Sim | Descrição deste sistema coordenado. |

Tabela 35. Colunas na exibição de catálogo DB2GSE.COORD_REF_SYS (continuação)

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|--------|---------------|----------------|---|
| SRTEXT | VARCHAR(2048) | Não | Texto de anotação para este sistema coordenado. |

DB2GSE.GEOMETRY_COLUMNS

Ao criar uma camada, o Spatial Extender registra-a gravando seu identificador e a informação relacionada à mesma em uma tabela de catálogo. As colunas selecionadas desta tabela compreendem a exibição de catálogo DB2GSE.GEOMETRY_COLUMNS, que é descrita na Tabela 36.

Tabela 36. Colunas na exibição de catálogo DB2GSE.GEOMETRY_COLUMNS

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|---------------|---------------|----------------|---|
| LAYER_CATALOG | VARCHAR(30) | Sim | NULL. Não há conceito de LAYER_CATALOG no Spatial Extender. |
| LAYER_SCHEMA | VARCHAR(30) | Não | Esquema da tabela ou da exibição que contém a coluna que foi registrada como esta camada. |
| LAYER_TABLE | VARCHAR(128) | Não | Nome da tabela ou da exibição que contém a coluna que foi registrada como esta camada. |
| LAYER_COLUMN | VARCHAR(128) | Não | Nome da coluna que foi registrada como esta camada. |
| GEOMETRY_TYPE | INTEGER | Não | Tipo de dados da coluna que foi registrada como esta camada. |
| SRID | INTEGER | Não | Identificador do sistema de referência espacial usado para os valores na coluna que foi registrada como esta camada. |
| STORAGE_TYPE | INTEGER | Sim | Informação de como o DB2 armazena os valores na coluna que foi registrada nesta camada. Por exemplo, os dados em STORAGE_TYPE podem indicar que os valores são armazenados como objetos grandes (LOBs). |

DB2GSE.SPATIAL_GEOCODER

Os geocoders disponíveis estão registrados em uma tabela do catálogo. As colunas selecionadas desta tabela compreendem a exibição de catálogo DB2GSE.SPATIAL_GEOCODER, que é descrita na Tabela 37 na página 145.

Tabela 37. Colunas na exibição de catálogo DB2GSE.SPATIAL_GEOCODER

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|-----------------|---------------|----------------|---|
| GCID | INTEGER | Não | Identificador numérico do geocoder. |
| GC_NAME | VARCHAR(64) | Não | Identificador de nome do geocoder. |
| VENDOR_NAME | VARCHAR(128) | Não | Nome do fornecedor do geocoder. |
| PRIMARY_UDF | VARCHAR(256) | Não | Nome completo do geocoder. |
| PRECISION_LEVEL | INTEGER | Não | O grau de coincidência dos dados fonte com os dados de referência correspondente a fim de ser processado com sucesso pelo geocoder. |
| VENDOR_SPECIFIC | VARCHAR(256) | Sim | Caminho e nome de um arquivo que um fornecedor pode utilizar para definir quaisquer parâmetros especiais suportados pelo geocoder. |
| GEO_AREA | VARCHAR(256) | Sim | Área geográfica contendo as localizações para serem geocodificadas. |
| DESCRIPTION | VARCHAR(256) | Sim | Descrição do geocoder. |

DB2GSE.SPATIAL_REF_SYS

Ao criar um sistema de referência espacial, o Spatial Extender registra-a gravando seu identificador e a informação relacionada à mesma em uma tabela de catálogo. As colunas selecionadas desta tabela compreendem a exibição de catálogo DB2GSE.SPATIAL_REF_SYS, que é descrita na Tabela 38.

Tabela 38. Colunas na exibição do catálogo DB2GSE.SPATIAL_REF_SYS

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|-----------|---------------|----------------|---|
| SRID | INTEGER | Não | Identificador definido pelo usuário para este sistema de referência espacial. |
| SR_NAME | VARCHAR(64) | Não | Nome deste sistema de referência espacial. |
| CSID | INTEGER | Não | Identificador numérico para o sistema coordenado que suporta este sistema de referência espacial. |
| CS_NAME | VARCHAR(64) | Não | Nome do sistema coordenado que suporta este sistema de referência espacial. |
| AUTH_NAME | VARCHAR(256) | Sim | Nome da organização que define os padrões para este sistema de referência espacial. |
| AUTH_SRID | INTEGER | Sim | O identificador que a organização especificada na coluna AUTH_NAME assinala para este sistema de referência espacial. |

Tabela 38. Colunas na exibição do catálogo DB2GSE.SPATIAL_REF_SYS (continuação)

| Nome | Tipo de Dados | Permite nulos? | Conteúdo |
|---------|---------------|----------------|--|
| SRTEXT | VARCHAR(2048) | Não | Texto de anotação para este sistema de referência espacial. |
| FALSEX | FLOAT | Não | Um número que, quando subtraído de um valor negativo da coordenada X, deixa um número não-negativo (ou seja, um número positivo ou zero). |
| FALSEY | FLOAT | Não | Um número que, quando subtraído de um valor negativo da coordenada Y, permite um número não-negativo (ou seja, um número positivo ou zero). |
| XYUNITS | FLOAT | Não | Um número que quando multiplicado por uma coordenada X decimal ou por uma coordenada Y decimal produz um inteiro que pode ser armazenado como um item de dados de 32-bits. |
| FALSEZ | FLOAT | Não | Um número que, quando subtraído de um valor negativo da coordenada Z, permite um número não-negativo (ou seja, um número positivo ou zero). |
| ZUNITS | FLOAT | Não | Um número que quando multiplicado por uma coordenada Z decimal produz um inteiro que pode ser armazenado como um item de dados de 32-bits. |
| FALSEM | FLOAT | Não | Um número que, quando subtraído de uma medida negativa, permite um número não-negativo (ou seja, um número positivo ou zero). |
| MUNITS | FLOAT | Não | Um número que quando multiplicado por uma medida decimal produz um inteiro que pode ser armazenado como um item de dados de 32-bits. |

Capítulo 12. Índices espaciais

Como as colunas espaciais contêm dados geográficos bidimensionais, as aplicações que consultam estas colunas precisam de uma estratégia de índice que identifique rapidamente todas as figuras geométricas contidas em uma determinada extensão. Por este motivo, o Spatial Extender fornece o índice espacial com três fileiras baseado em uma grade.

Este capítulo descreve este tipo de índice e fornece diretrizes sobre o uso do mesmo. Os tópicos abrangidos são:

- “Fragmento de um programa de amostra”
- “Índices B tree” na página 148
- “Formas de se criar um índice espacial” na página 148
- “Como é gerado um índice espacial” na página 149
- “Diretrizes sobre o uso de um índice espacial” na página 153

Fragmento de um programa de amostra

Considere o seguinte exemplo de como um índice é criado e utilizado no SQL. Você pode consultar a *SQL Reference* para obter maiores informações sobre os comandos CREATE INDEX e CREATE INDEX EXTENSION. Observe que depois que o índice é criado, você pode emitir instruções DDL e DML padrões que utilizam as funções e predicados espaciais.

```
create table customers (cid int, addr varchar(40), ..., loc db2gse.ST_Point)
create table stores (sid int, addr varchar(40), ..., loc db2gse.ST_Point,
    zone db2gse.ST_Polygon)

create index customersx1 on customers(loc) extend using db2gse.spatial_index(10e0,
    100e0, 1000e0)
create index storesx1 on stores(loc) extend using db2gse.spatial_index(10e0, 100e0,
    1000e0)
create index storesx2 on stores(zone) extend using db2gse.spatial_index(10e0, 100e0,
    1000e0)

insert into customers (cid, addr, loc) values
(:cid, :addr, sdeFromBinary(:loc))
insert into customers (cid, addr, loc) values
(:cid, :addr, geocode(:addr))
insert into stores (sid, addr, loc) values
(:sid, :addr, sdeFromBinary(:loc))

update stores set zone = db2gse.ST_Buffer (loc, 2)

select cid, loc from customers
    where db2gse.ST_Within(loc, :polygon) = 1
```

```

select cid, loc from customers
  where db2gse.ST_Within(loc, :circle1) = 1 OR
         db2gse.ST_Within(loc, :circle2) = 1

select c.cid, loc from customers c, stores s
  where db2gse.ST_Contains(s.zone, c.loc) = 1 selectivity 0.01

select avg(c.income) from customers c
  where not exist (select * from stores s
                  where db2gse.ST_Distance(c.loc, s.loc) < 10)

```

Índices B tree

A tecnologia de indexação espacial baseia-se no índice B tree hierárquico tradicional, mas é significativamente diferente. O índice espacial utiliza a *indexação de grade* que é projetada para indexar colunas espaciais bidimensionais. O índice B tree pode manipular apenas dados mono-dimensionais e não pode ser utilizado com informações GIS. Esta seção descreve como um índice B tree é estruturado e utilizado.

O nível mais elevado de um índice B tree, chamado de nó raiz, contém uma chave para cada nó no nível seguinte. O valor de cada chave é o maior valor de chave existente para o nó correspondente no nível seguinte. Dependendo do número de valores na tabela base, é possível que sejam necessários diversos nós intermediários. Estes nós formam uma ponte entre o nó raiz e os nós folha que retêm as efetivas IDs de linha da tabela base.

O gerenciador de banco de dados faz pesquisas em um índice B tree começando no nó raiz. Depois, ele continua pelos nós intermediários até chegar no nó folha com a ID de linha da tabela base.

O índice B tree não pode ser aplicado em uma coluna espacial pois a característica bidimensional da coluna espacial exige a estrutura de um índice espacial. Pelo mesmo motivo, não é possível aplicar um índice espacial em uma coluna não-espacial. Além disso, um índice espacial não pode ser criado a partir de várias colunas.

Formas de se criar um índice espacial

Há várias formas de se criar um índice espacial:

- Definir um a partir da janela Criar Índice Espacial. Para obter instruções, consulte “Capítulo 6. Criando índices espaciais” na página 67.

- Solicitar o procedimento armazenado `db2gse.gse_enable_idx` em um programa da aplicação. Para obter informações sobre este procedimento armazenado, consulte “Capítulo 9. Procedimentos armazenados” na página 83.
- Emitir o comando **db2 create index** com a função **spatial_index** na cláusula **USING**. Por exemplo:


```
create index storesx1 on customers (loc) using db2gse.spatial_index(10e0,
100e0, 1000e0)
```

A natureza dos dados espaciais exige que o projetista de banco de dados compreenda sua distribuição relativa de tamanho. O projetista deve determinar o tamanho satisfatório e o número de níveis de grade com os quais criar o índice espacial.

Os níveis de grade, <nível de grade 1>, <nível de grade 2> e <nível de grade 3>, são fornecidos aumentando-se o tamanho da célula. Conseqüentemente, o segundo nível deve ter um tamanho de célula maior do que o primeiro e o terceiro maior que o segundo. O primeiro nível de grade é obrigatório, mas você pode desativar o segundo e o terceiro com um valor zero de precisão dupla (0.0e0).

Como é gerado um índice espacial

Um índice espacial é gerado com o uso de *envelopes*. O envelope é uma figura geométrica em si e representa a extensão mínima e máxima de X e Y de uma figura geométrica. Para maioria das figuras geométricas, o envelope é uma caixa, mas para linhas horizontais e verticais, o envelope é uma linha com dois pontos. Para os pontos, o envelope é o próprio ponto. Para obter mais informações sobre envelopes, consulte “Envelope” na página 163.

O índice espacial é construído em uma coluna espacial produzindo uma ou mais entradas para as interseções de cada envelope da figura geométrica com a grade. Uma interseção é registrada como a ID interna da figura geométrica e das coordenadas mínimas X e Y da célula de grade atravessada. Por exemplo, o polígono na Figura 7 na página 150 intersecta a grade nas coordenadas (20,30), (30,30), (40,30), (20,40), (30,40), (40,40), (20,50), (30,50) e (40,50). Consulte Tabela 39 na página 150 para obter as coordenadas mínimas X e Y para todas as figuras geométricas na Figura 7 na página 150.

Se houver vários níveis de grade, o Spatial Extender tenta utilizar o nível de grade mais baixo possível. Quando uma figura geométrica faz interseção com quatro ou mais células da grade no nível fornecido, ela é promovida para o próximo nível mais alto. Sendo assim, um índice espacial que possui três níveis de grade de 10.0e0, 100.0e0 e 1000.0e0, o Spatial Extender intersecta primeiro cada figura geométrica com a grade de nível 10.0e0. Se uma figura geométrica intersecta quatro ou mais células de grade 10.0e0, ela é promovida

e faz interseção com a grade de nível 100.0e0. Se quatro ou mais interseções resultam no nível 100.0e0, a figura geométrica é promovida para o nível 1000.0e0. No nível 1000.0e0, as interseções devem ser fornecidas no índice espacial pois este é o nível mais alto possível.

A Figura 7 ilustra como quatro tipos diferentes de figuras geométricas intersectam uma grade 10.0e. Todas as 23 interseções para as quatro figuras geométricas são registradas no índice espacial.

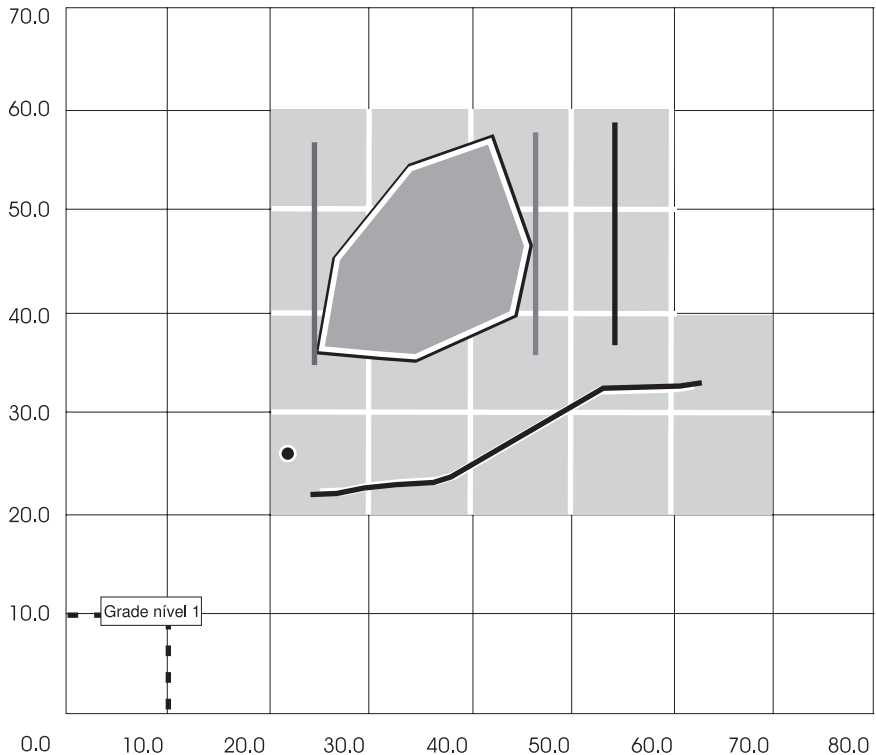


Figura 7. Aplicação de um nível de grade 10.0e0

A Tabela 39 relaciona as figuras geométricas e suas interseções de grades correspondentes. Os envelopes de quatro tipos de figuras geométricas diferentes intersectam a grade 10.0e. A coordenada mínima X e Y de cada célula de grade que ela intersecta é fornecida no índice espacial.

Tabela 39. As entradas da célula de grade 10.0e0 para os exemplos de figuras geométricas

| Figura geométrica | Grade X | Grade Y |
|-------------------|---------|---------|
| Polígono | 20.0 | 30.0 |

Tabela 39. As entradas da célula de grade 10.0e0 para os exemplos de figuras geométricas (continuação)

| Figura geométrica | Grade X | Grade Y |
|--------------------------|----------------|----------------|
| Polígono | 30.0 | 30.0 |
| Polígono | 40.0 | 30.0 |
| Polígono | 20.0 | 40.0 |
| Polígono | 30.0 | 40.0 |
| Polígono | 40.0 | 40.0 |
| Polígono | 20.0 | 50.0 |
| Polígono | 30.0 | 50.0 |
| Polígono | 40.0 | 50.0 |
| Linha vertical | 50.0 | 30.0 |
| Linha vertical | 50.0 | 40.0 |
| Linha vertical | 50.0 | 50.0 |
| Ponto | 20.0 | 20.0 |
| Linha Horizontal | 20.0 | 20.0 |
| Linha Horizontal | 30.0 | 20.0 |
| Linha Horizontal | 40.0 | 20.0 |
| Linha Horizontal | 50.0 | 20.0 |
| Linha Horizontal | 60.0 | 20.0 |
| Linha Horizontal | 20.0 | 30.0 |
| Linha Horizontal | 30.0 | 30.0 |
| Linha Horizontal | 40.0 | 30.0 |
| Linha Horizontal | 50.0 | 30.0 |
| Linha Horizontal | 60.0 | 30.0 |

A Figura 8 na página 152 mostra como o número de interseções é notavelmente reduzido para oito através do acréscimo dos níveis de grade 30.0e0 e 60.0e0. Neste caso, o polígono identificado como figura geométrica 1 é promovido para o nível de grade 30.0e0 e a linha identificada como figura geométrica 4 é promovida para o nível de grade 60.0e0. Ao invés de nove e dez interseções que as figuras geométricas tinham no nível 10.0e0, elas tem apenas duas após a promoção.

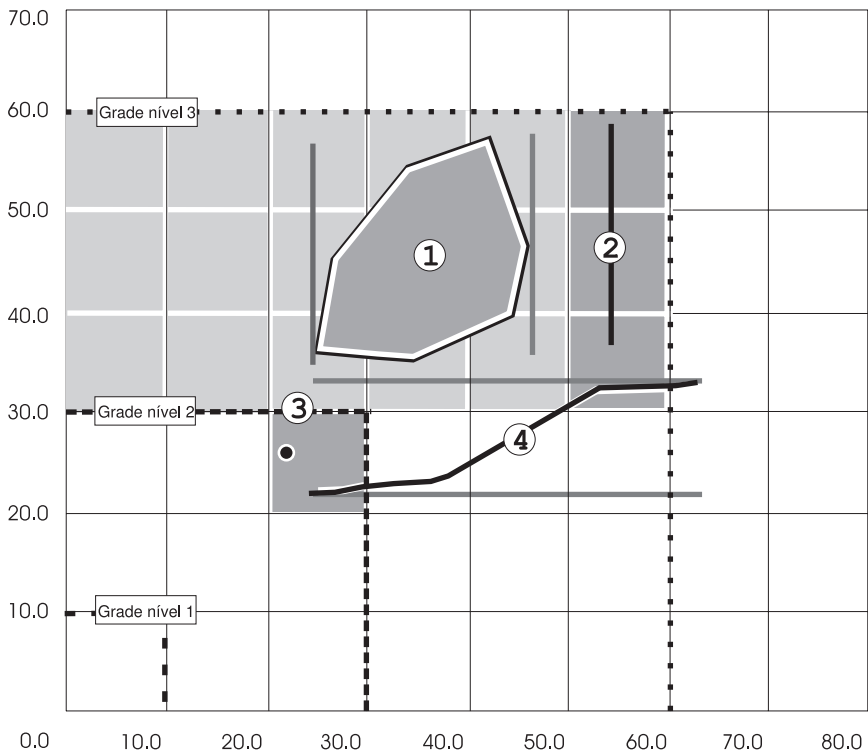


Figura 8. Efeito do acréscimo dos níveis de grade 30.0e0 e 60.0e0. O envelope do polígono identificado como a figura geométrica 1 intersecta nove células da grade. O envelope da linha vertical identificado como figura geométrica 2 intersecta três células da grade. O envelope do ponto identificado como figura geométrica 3 intersecta apenas uma célula da grade. O envelope da linha identificada como figura geométrica 4 intersecta dez células da grade.

O Spatial Extender obtém os parâmetros de nível de grade especificados na instrução CREATE INDEX e verifica cada objeto espacial para determinar as coordenadas e o número de blocos da grade em que o objeto existe. Na Figura 8, os níveis de grade 10.0e0, 30.0e0 e 60.0e0 são exibidos com pesos de linha sempre crescentes e diferentes tonalidades de cinza. As interseções da linha vertical e da célula do envelope de ponto são fornecidas no índice no nível de grade 10.0e0, pois ambas geram menos do que quatro interseções. O polígono intersecta nove células de grade 10.0e0 e é promovido para o nível de grade 30.0e0. Neste nível, o polígono intersecta duas células de grade, que são fornecidas no índice. A linha identificada como figura geométrica 4 intersecta dez células de grade 10.0e0 e é promovida para o nível de grade 30.0e0. Ainda neste nível, ela intersecta seis células de grade, e é então novamente promovida para o nível de grade 60.0e0, onde ela gera duas interseções. Depois, as interseções da grade 60.0e0 da linha são fornecidas no índice. Se a linha tivesse gerado quatro ou mais interseções neste nível, elas ainda teriam sido fornecidas no índice pois este é o nível mais alto em que uma figura geométrica pode ser promovida.

Tabela 40. As interseções das figuras geométricas no índice de três fileiras

| Figura geométrica | Grade X | Grade Y |
|---|---------|---------|
| <i>As interseções entre a linha vertical e o ponto no nível 1 (tamanho de grade 10.0e0)</i> | | |
| 2 | 50.0 | 30.0 |
| 2 | 50.0 | 40.0 |
| 2 | 50.0 | 50.0 |
| 3 | 20.0 | 20.0 |
| <i>As interseções do polígono no nível 2 (tamanho de grade 30.0e0)</i> | | |
| 1 | 0.0 | 30.0 |
| 1 | 30.0 | 30.0 |
| <i>As interseções da linha no nível 3 (tamanho de grade 60.0e0)</i> | | |
| 4 | 0.0 | 0.0 |
| 4 | 60.0 | 0.0 |

Na verdade, o Spatial Extender não cria uma estrutura de grade do polígono de tipo algum. O Spatial Extender demonstra cada nível de grade, de forma paramétrica, definindo a origem no deslocamento de X,Y do sistema de referência espacial das colunas. Depois, ele estende a grade até o espaço da coordenada positiva. Utilizando uma grade paramétrica, o Spatial Extender gera, matematicamente, as interseções.

Diretrizes sobre o uso de um índice espacial

O Spatial Extender trabalha com índices espaciais para melhorar o desempenho das consultas espaciais. Considere a consulta espacial mais básica e provavelmente a mais popular, a consulta de caixa. Esta consulta solicita que o Spatial Extender retorne todas as figuras geométricas que estão completamente ou parcialmente dentro de uma caixa definida pelo usuário. Quando não há um índice, o Spatial Extender precisa comparar todas as figuras geométricas com a caixa. No entanto, com um índice, o Spatial Extender pode localizar todas as entradas do índice que possuem uma coordenada esquerda inferior maior ou igual a caixa e uma coordenada direita superior menor que ou igual a da caixa. Como o índice é ordenado por este sistemas de coordenadas, o Spatial Extender consegue obter rapidamente uma lista das figuras geométricas candidatas. O processo que acabou de ser descrito é chamado de *primeiro passo*.

Um *segundo passo* determina se cada envelope do candidato intersecta a caixa. Uma figura geométrica que se qualifica para o primeiro passo porque seu envelope de células de grade intersecta a caixa pode ela mesmo ter um envelope que não intersecta a caixa.

Um *terceiro passo* compara as coordenadas vigentes do candidato com a caixa para determinar se alguma parte da figura geométrica está realmente dentro da caixa. Este último é um tanto complexo processo de comparação opera em uma lista de candidatos composta por um subconjunto da população total, que é significativamente reduzida pelos dois primeiros passos.

Todas as consultas espaciais realizam os três passos exceto pela função `EnvelopesIntersect`. Ela realiza apenas os dois primeiros passos. A função `EnvelopesIntersect` foi projetada para exibir operações que geralmente empregam suas próprias rotinas de recorte incorporadas e não requerem a granulosidade do terceiro passo.

Selecionando o tamanho da célula de grade

O formato irregular dos envelopes da figura geométrica dificultam a seleção do tamanho da célula de grade. Devido a esta irregularidade, alguns envelopes de figuras geométricas intersectam diversas grades, enquanto outros se ajustam dentro de uma única célula de grade. Inversamente, dependendo da distribuição espacial dos dados, algumas células de grade intersectam vários envelopes de figuras geométricas.

Para que um índice espacial funcione bem, é essencial que o número e o tamanho corretos das grades sejam selecionados. Considere uma coluna espacial que contém figuras geométricas dimensionadas uniformemente. Neste caso, um único nível de grade será suficiente. Comece com um tamanho de célula de grade que abranja o envelope médio da figura geométrica. Ao testar sua aplicação, você pode perceber que aumentar o tamanho da célula de grade melhora o desempenho de suas consultas. Isto porque cada célula de grade contém mais figuras geométricas e o primeiro passo tem condições de descartar mais rapidamente figuras geométricas não-qualificadas. No entanto, você irá perceber que à medida que você continua a aumentar o tamanho da célula, o desempenho começa a decair. Isto porque eventualmente o segundo passo terá que sustentar mais candidatos.

Selecionando o número de níveis

Se os objetos a serem indexados são aproximadamente do mesmo tamanho relativo, você pode usar um único nível de grade. Embora isto seja verdade, nem todas as colunas irão conter figuras geométricas do mesmo tamanho relativo. Geralmente, as figuras geométricas de colunas espaciais podem ser agrupadas em diversos intervalos de tamanho. Por exemplo, considere uma rede rodoviária na qual as figuras geométricas são divididas em ruas, avenidas e estradas. As ruas são quase todas do mesmo comprimento e podem ser agrupadas em um intervalo de tamanho. Isto também é verdadeiro para as avenidas e estradas. Sendo assim, as ruas, representando o um intervalo de tamanho, poderiam ser agrupadas no primeiro nível de grade, as redes rodoviárias no segundo e as estradas no terceiro nível de grade. Outro exemplo inclui uma coluna de lote de município que contém grupos de

pequenos lotes urbanos cercados por lotes rurais maiores. Neste caso, há dois intervalos de tamanho e dois níveis de grade, uma para os pequenos lotes urbanos e outro para os lotes rurais maiores. Estas situações são muito comuns e exigem o uso de uma grade de multiníveis.

Para selecionar o tamanho de célula para cada nível de grade, selecione os tamanhos da célula de grade que são um pouco maiores do que cada intervalo de tamanho. Teste o índice realizando consultas junto à coluna espacial.

Cada nível adicional requer uma pesquisa de índice extra. Tente ajustar os tamanhos de grade um pouco para cima ou para baixo para determinar se uma melhoria considerável no desempenho pode ser obtida.

Capítulo 13. Figuras geométricas e funções espaciais associadas

Este capítulo discute as unidades de informações, denominadas *figuras geométricas*, que consistem em coordenadas e simbolizam recursos geográficos. Apresenta também as funções espaciais que tomam figuras geométricas como entrada e retornam resultados para ajudá-lo a analisar recursos geográficos e deslocar dados espaciais entre sistemas informativos geográficos. Os tópicos abrangidos são:

- A natureza das figuras geométricas
- As propriedades das figura geométrica; funções que retornam informações relacionadas a estas propriedades
- figuras geométricas instanciáveis; funções que operam nelas
- Funções que:
 - Mostram relações e comparações entre recursos geográficos
 - Geram figuras geométricas
 - Convertem valores de figura geométrica em formatos para importação e exportação

Sobre figuras geométricas

O dicionário Oxford (Oxford American Dictionary) define *figura geométrica* como “ramo da matemática que estuda as propriedades e relações das linhas, ângulos, superfícies e sólidos.” Em 11 de agosto de 1997, o Open GIS Consortium Inc. (OGC) em sua publicação, *Open GIS Features for ODBC (SQL) Implementation Specification*, criou outra definição para o termo. A palavra *figura geométrica* foi selecionada para designar os recursos geométricos que, desde o último milênio, ou mais, os cartógrafos usaram para mapear o mundo. Uma definição muito abstrata deste novo significado de figura geométrica pode ser “um ponto ou conjunto de pontos que simbolizam um recurso na terra.”

No Spatial Extender, uma definição *operacional* de figura geométrica poderia ser “um modelo de recursos geográfico.” O modelo pode ser expresso em termos das coordenadas do recurso e também, em alguns casos, de um símbolo visual. O modelo transmite informações; por exemplo, as coordenadas identificam a posição do recurso com relação aos pontos fixos da referência e o símbolo descreve seu formato. Além disso, o modelo pode ser usado para produzir informações; por exemplo, a função `ST_Overlaps` pode tomar as coordenadas de duas regiões próximas como entrada e retornar informações indicando se as regiões estão sobrepostas.

As coordenadas de um recurso que uma figura geométrica simboliza são consideradas *propriedades* da figura geométrica. Vários tipos de figuras geométricas também têm outras propriedades, por exemplo:

- Um *interior* representa o conteúdo do recurso que a figura geométrica simboliza.
- Um *exterior* representa o espaço ao redor do recurso.
- Um *limite* representa a demarcação em que o conteúdo termina e o espaço adjacente começa.

Estas propriedades adicionais serão discutidas em “Propriedades e funções associadas” na página 159.

As figuras geométricas suportadas pelo Spatial Extender formam uma hierarquia, mostrada na Figura 9. Seis membros da hierarquia são instanciáveis, eles podem ser expressos como símbolos visuais, que são mostrados na figura.

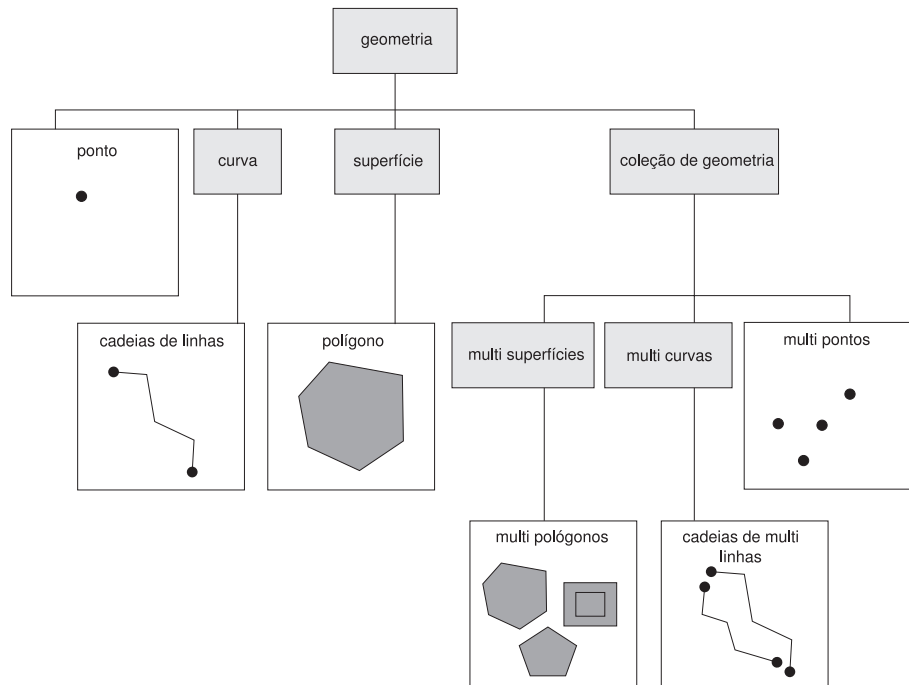


Figura 9. Hierarquia das figuras geométricas suportadas pelo Spatial Extender. As figuras geométricas instanciáveis podem ser expressas como símbolos visuais. Tais símbolos são mostrados abaixo dos nomes destas figuras geométricas.

Como Figura 9 na página 158 indica, uma superclasse chamada *figura geométrica* é a raiz da hierarquia. Os subtipos são divididos em duas categorias: os subtipos de figura geométrica base e os subtipos de coleção homogênea. As figuras geométricas base incluem:

- *Pontos*, que simbolizam recursos discretos que são captados como ocupantes do local em que uma linha da coordenada leste/oeste (como um paralelo) intercepta uma linha da coordenada norte/sul (como um meridiano). Por exemplo, supondo-se que a notação de um mapa em escala grande mostra que cada cidade no mapa está localizada na interseção de um paralelo e de um meridiano. Nesta escala, cada cidade seria representada por um ponto.
- *Cadeias de linhas*, que simbolizam recursos geográficos lineares (por exemplo, ruas, canais e pipelines).
- *Polígonos*, os quais simbolizam recursos geográficos multifacetados (por exemplo, distritos de preservação, florestas e habitats naturais).

As coleções homogêneas incluem:

- *MultiPontos*, que simbolizam recursos de várias partes cujos componentes estão localizados cada um na interseção de uma linha coordenada leste/oeste e uma linha coordenada norte/sul (por exemplo, uma cadeia de ilhas cujos membros estejam situados cada um em uma interseção de um paralelo ou meridiano).
- *Cadeias de linhas múltiplas*, que simbolizam recursos de várias partes compostos de unidades ou componentes lineares (por exemplo, sistemas fluviais e sistemas rodoviários).
- *Multipolígonos*, que simbolizam recursos de várias partes compostos de unidades ou componentes multifacetados (por exemplo, as fazendas coletivas de uma região específicas ou um sistema de lagos).

Como seu nome implica, as coleções homogêneas são coleções de figuras geométricas básicas. Além de compartilhar as propriedades da figura geométrica básica, as coleções homogêneas possuem algumas propriedades próprias também.

Os tipos de dados espaciais suportados pelo Spatial Extender são implementações das figuras geométricas mostradas na Figura 9 na página 158. Para obter uma descrição destes tipos de dados, consulte “Sobre tipos de dados espaciais” na página 45.

Propriedades e funções associadas

Esta seção descreve as propriedades das figuras geométricas e as funções espaciais que estão associadas à estas propriedades. As propriedades são:

- A classe a que uma figura geométrica pertence
- Coordenadas e medidas

- Um interior, limite e exterior da figura geométrica
- Coordenadas Z
- Medidas
- A qualidade de ser simples ou não-simples
- A qualidade de ser vazio ou não-vazio
- Um envelope de figura geométrica
- Dimensão
- O identificador de uma figura geométrica associada ao sistema de referência espacial

Classe

Cada figura geométrica pertence a uma classe na hierarquia mostrada na Figura 9 na página 158. Conforme indicado em “Sobre figuras geométricas” na página 157, seis subtipos na hierarquia — pontos, cadeias de linhas, polígonos, MultiPontos, cadeias de linhas múltiplas e multipolígonos — são instanciáveis. A superclasse e outros subtipos não são instanciáveis.

A função `ST_GeometryType` recebe uma figura geométrica e retorna um identificador de cadeia de caracteres para o subtipo instanciável. Para obter mais informações, consulte “`ST_GeometryType`” na página 253.

A função `ST_IsValid()` recebe um valor `ST_Geometry` como seu parâmetro de entrada. A função retorna 1 (VERDADEIRO) se a figura geométrica estiver válida e 0 (FALSO) se não estiver válida. Para obter mais informações, consulte “`ST_IsValid`” na página 274.

Coordenadas e medidas

Todas as figuras geométricas incluem, pelo menos, uma coordenada X e uma coordenada Y. Além disso, uma figura geométrica pode incluir uma ou mais coordenadas Z e medidas. As subseções a seguir abordam:

- Coordenadas X e Y
- Coordenadas Z e medidas
- A função `ST_CoordDim`

Coordenadas X e Y

Um *valor da coordenada X* indica um local que é relativo a um ponto de referência a leste ou oeste. Um *valor da coordenada Y* indica um local que é relativo a um ponto de referência ao norte ou sul. Para mais informações, consulte “A natureza dos dados espaciais” na página 6 e “Sobre sistemas de coordenadas e de referência espacial” na página 35.

Coordenadas Z e medidas

Esta seção aborda as coordenadas Z, medidas e a maneira que são utilizadas no DB2 Spatial Extender.

Coordenadas Z

Algumas figuras geométricas apresentam uma altitude ou profundidade associada. Cada um dos pontos que formam a figura geométrica de um recurso podem incluir uma coordenada Z opcional que representa uma altitude ou profundidade normal à superfície da terra.

Utilize a função `AsShape` para converter um valor geométrico em uma representação de formato do ESRI. Se o valor da figura geométrica incluir quaisquer coordenadas Z e medidas, elas serão retidas na representação de formato.

A função do predicado `Is3d` recebe uma figura geométrica e retorna 1 se a função tiver coordenadas Z e, do contrário, 0. Para obter mais informações, consulte “`Is3d`” na página 203.

Medidas

Uma medida é um valor que expressa informações sobre um recurso geográfico que é armazenado junto às coordenadas que definem a localização do recurso. Por exemplo, suponha que você está representando sistemas de transporte no GIS. Se desejar que sua aplicação processe seus valores de processo que indicam distâncias lineares ou postes, você pode armazenar estes valores junto às coordenadas que definem as localizações dos sistemas. As medidas são armazenadas como números de precisão dupla.

Como as coordenadas Z e as medidas são utilizadas

No DB2 Spatial Extender, as coordenadas Z e as medidas são utilizadas para transmitir informações aos aplicativos, não para definir localizações. (Por exemplo, consulte “Sistemas de coordenadas, coordenadas e medidas,” uma subseção no capítulo 13 [“Configuração de recursos”].) Desta maneira, quando a maioria das funções espaciais processam pontos que incluem coordenadas Z e/ou medidas, as funções ignoram estes valores. Outras funções espaciais realizam, entretanto, operações em coordenadas Z e medidas:

- Você pode utilizar as funções `Is3d`, `IsMeasured` e `ST_CoordDim` para descobrir se uma figura geométrica inclui uma coordenada Z, uma medida, ou ambas.
- Pode utilizar as funções `M` e `Z` para descobrir qual é a coordenada Z e a medida de um ponto.
- O predicado `IsMeasured` toma uma figura geométrica e retorna 1 (VERDADEIRO) se contiver medidas e, do contrário, 0 (FALSO). Para obter mais informações, consulte “`IsMeasured`” na página 204.

A função ST_CoordDim

ST_CoordDim retorna um valor que denota quais tipos de coordenadas uma figura geométrica possui e se ela contém também alguma medida. Este valor é chamado de *dimensão da coordenada*.

ST_CoordDim pode retornar uma dimensão de coordenada de 2, 3 ou 4:

- Se a entrada para ST_CoordDim for um ponto, um valor retornado de 2 significa que o ponto consiste em uma coordenada X e uma coordenada Y. Se a entrada for um tipo de figura geométrica diferente de um ponto, 2 significará que cada ponto nesta figura geométrica consistirá em uma coordenada X e uma coordenada Y. .
- Se a entrada para ST_CoordDim for um ponto, um valor retornado de 3 significa que o ponto consiste em uma coordenada X, uma coordenada Y e ou uma coordenada Z ou uma medida. Se a entrada for um tipo de figura geométrica diferente de um ponto, um valor de 3 significa que cada ponto nessa figura geométrica consiste em uma coordenada X, uma coordenada Y e ou uma coordenada Z ou uma medida. .
- Se a entrada para ST_CoordDim for um ponto, um valor retornado de 4 significa que o ponto consiste em uma coordenada X, uma coordenada Y, uma coordenada Z e uma medida. Se a entrada for um tipo de figura geométrica diferente de um ponto, um valor de 4 significa que cada ponto nesta figura geométrica consiste em uma coordenada X, uma coordenada Y, uma coordenada Z e uma medida. .

Interior, limite e exterior

Todas as figuras geométricas ocupam uma posição no espaço definido por seu interior, limite e exterior. O exterior de uma figura geométrica é todo o espaço não ocupado pela figura geométrica. O limite de uma figura geométrica serve como a interface entre seu interior e exterior. O interior é o espaço ocupado pela figura geométrica. Os subtipos herdam as propriedades interiores e exteriores diretamente, embora a propriedade de limite seja diferente para cada um.

A função ST_Boundary toma uma figura geométrica e retorna uma que represente o limite da figura geométrica de origem. Para obter mais informações, consulte “ST_Boundary” na página 226.

Simples ou não-simples

Alguns subtipos de figura geométrica (cadeias de linhas, Multipontos e cadeias de linhas múltiplas) são simples ou não-simples. Um subtipo é simples se obedecer a todas as regras topológicas impostas ao subtipo e não-simples se não obedecer. Uma cadeia de linhas é simples se não fizer interseção com seu interior. Um Multiponto é simples se nenhum de seus

elementos ocupar o mesmo espaço da coordenada. Uma cadeia de linhas múltiplas é simples se nenhum interior de seu elemento fizer interseção pelo seu próprio interior.

A função do predicado `ST_IsSimple` toma uma figura geométrica e retorna 1 (VERDADEIRO) se a figura geométrica for simples e, caso contrário, 0 (FALSO). Para obter mais informações, consulte “`ST_IsSimple`” na página 273.

Vazia ou não vazia

Uma figura geométrica é vazia se não possuir pontos. O envelope, o limite, o interior e o exterior de uma figura geométrica vazia são NULOS. Uma figura geométrica vazia é sempre simples e pode ter coordenadas Z ou medidas. Cadeias de linhas e cadeias de linhas múltiplas vazias têm comprimento 0. Os polígonos vazios e os multipolígonos têm área 0.

A função do predicado `ST_IsEmpty` toma uma figura geométrica e retorna 1 (VERDADEIRO) se a figura geométrica estiver vazia e, do contrário, 0 (FALSO). Para obter mais informações, consulte “`ST_IsEmpty`” na página 269.

Envelope

O envelope de uma figura geométrica é a figura geométrica limitada formada por coordenadas mínimas e máximas (X,Y). Exceto para os seguintes casos especiais, os envelopes de figuras geométricas formam um retângulo:

- O envelope de qualquer ponto é o próprio ponto, porque suas coordenadas X mínimas e máximas são as mesmas e suas coordenadas Y mínimas e máximas são as mesmas.
- O envelope de uma cadeia de linha horizontal ou vertical é uma cadeia de linha representada pelo limite (os nós de extremidade) da cadeia de linha fonte.

A função `ST_Envelope` toma uma figura geométrica e retorna uma figura geométrica limitada, que representa seu envelope. Para obter mais informações, consulte “`ST_Envelope`” na página 246.

Dimensão

Uma figura geométrica pode ter uma dimensão de 0, 1 ou 2. As dimensões estão relacionadas da seguinte forma:

- | | |
|---|------------------------------|
| 0 | Não tem comprimento nem área |
| 1 | Tem comprimento |
| 2 | Contém área |

Os subtipos ponto e MultiPonto possuem uma dimensão de zero. Os pontos representam recursos dimensionais que podem ser modelados com uma única coordenada, ao passo que os subtipos MultiPonto representam dados que devem ser modelados com um grupo de coordenadas desconectadas.

Os subtipos cadeia de linhas e cadeia de linhas múltiplas possuem uma dimensão de um. Elas armazenam segmentos de rodovia, extensões de rios e quaisquer outros recursos que sejam lineares na natureza.

Os subtipos polígono e multipolígono possuem uma dimensão de dois. Os recursos cujos perímetros englobem uma área definível, como florestas, pedaços de terra e rios podem ser cedidos pelo tipo de dados polígono ou multipolígono.

A dimensão é importante não somente como uma propriedade do subtipo, como também desempenha um papel na determinação da relação espacial de dois recursos. A dimensão dos recursos resultantes determina se operação foi bem-sucedida. O Spatial Extender examina a dimensão dos recursos para determinar como serão comparados.

A função `ST_Dimension` toma uma figura geométrica e retorna sua dimensão como um inteiro. Para obter mais informações, consulte “`ST_Dimension`” na página 240.

Identificador do sistema de referência espacial

O sistema de referência espacial identifica a transformação de coordenadas para cada figura geométrica.

Todos os sistemas de referências espaciais reconhecidos pelo banco de dados podem ser acessados através da exibição do catálogo `DB2GSE.SPATIAL_REF_SYS`. Para obter informações sobre esta exibição, consulte “`DB2GSE.SPATIAL_REF_SYS`” na página 145.

A função `ST_SRID` recebe uma figura geométrica e retorna seu identificador de referência espacial como um inteiro. Para obter mais informações, consulte “`ST_SRID`” na página 307.

A função `ST_Transform` atribui uma figura geométrica a um sistema de referências espaciais diferente do sistema de referências espaciais para o qual a figura geométrica está atualmente atribuída. Para obter mais informações, consulte “`ST_Transform`” na página 312.

figuras geométricas instanciáveis e funções associadas

Esta seção descreve os seis subtipos de figuras geométricas instanciáveis e as funções que operam nelas. Os subtipos são:

- Pontos
- Cadeias de linhas
- Polígonos
- MultiPontos

- Cadeias de linhas múltiplas
- Multipolígonos

Para obter ilustrações da hierarquia a que pertencem esses subtipos e dos símbolos visuais associados a eles, consulte Figura 9 na página 158.

Pontos

Um ponto indica uma figura geométrica dimensional zero que ocupa uma única localização no espaço da coordenada. Um ponto inclui uma coordenada X e uma coordenada Y que definem esta localização. Também pode incluir uma coordenada Z e uma medida.

Um ponto é simples e possui um limite NULL. Os pontos geralmente são utilizados para definir recursos como poços de petróleo, pontos de referência e elevações.

Funções que operam exclusivamente no subtipo de ponto:

PointFromShape

Recebe um formato do tipo ponto e um identificador do sistema de referências espaciais e retorna um ponto. Para obter mais informações, consulte “PointFromShape” na página 216.

ST_Point

Toma uma coordenada X, sua coordenada Y associada e o identificador do sistema de referência espacial, ao qual pertencem estas coordenadas, e retorna o ponto que as coordenadas definem. Para obter mais informações, consulte “ST_Point” na página 295.

ST_PointFromText

Recebe uma representação de texto binário OGC convencional de um ponto e retorna o ponto. Para obter mais informações, consulte “ST_PointFromText” na página 296.

ST_PointFromWKB

Recebe uma representação binária reconhecida do tipo polígono e um identificador do sistema de referências espaciais e retorna um polígono. Para obter mais informações, consulte “ST_PointFromWKB” na página 297.

ST_X Retorna um valor da coordenada X do tipo de dados ST_Point como um número de precisão dupla. Para obter mais informações, consulte “ST_X” na página 319.

ST_Y Retorna um valor da coordenada Y do tipo de dados ST_Point como um número de precisão dupla. Para obter mais informações, consulte “ST_Y” na página 320.

- Z** Retorna um valor da coordenada Z do tipo de dados ST_Point como um número de precisão dupla. Para obter mais informações, consulte “Z” na página 321.
- M** Retorna uma medida do tipo de dados ST_Point como um número de precisão dupla. Para obter mais informações, consulte “M” na página 211.

Cadeias de linhas

Uma cadeia de linha é um objeto unidimensional armazenando como uma seqüência de pontos que definem um caminho linear interpolado. A cadeia de linhas é simples se não fizer interseção com seu interior. Os nós de extremidade (o limite) de uma cadeia de linhas fechada ocupam o mesmo ponto no espaço. Uma cadeia de linha indica um anel se estiver fechada e seu interior não se interceptar. Além das outras propriedades herdadas da figura geométrica da superclasse, as cadeias de linhas têm comprimento. Elas são geralmente usadas para definir recursos lineares como estradas, rios e linhas de energia.

Uma cadeia de linha simples cujo ponto inicial e final sejam os mesmos é denominada *anel*.

Os nós de extremidade geralmente formam um limite de uma cadeia de linha, a menos que as linhas estejam fechadas, tornando o limite NULO. O interior de uma cadeia de linha é o caminho conectado que fica entre os nós de extremidade, a menos que esteja fechada, tornando o interior contínuo.

Funções que operam em cadeias de linhas:

ST_StartPoint

Toma uma cadeia de linha e retorna seu primeiro ponto. Para obter mais informações, consulte “ST_StartPoint” na página 308.

ST_EndPoint

Toma uma cadeia de linha e retorna seu último ponto. Para obter mais informações, consulte “ST_Endpoint” na página 245.

ST_PointN

Toma uma cadeia de linha e um índice ao ponto *n* e retorna esse ponto. Para obter mais informações, consulte “ST_PointN” na página 299.

ST_Length

Toma uma cadeia de linha e retorna seu comprimento como número de precisão dupla. Para obter mais informações, consulte “ST_Length” na página 276.

ST_NumPoints

Toma uma cadeia de linha e retorna o número de pontos na sua seqüência como um inteiro. Para obter mais informações, consulte “ST_NumPoints” na página 290.

ST_IsRing

Toma uma cadeia de linha e retorna 1 (VERDADEIRO) se a cadeia de linha for um anel e, do contrário, 0 (FALSO). Para obter mais informações, consulte “ST_IsRing” na página 271.

ST_IsClosed

Toma uma cadeia de linha e retorna 1 (VERDADEIRO) se a cadeia de linha estiver fechada e, do contrário, 0 (FALSO). Para obter mais informações, consulte “ST_IsClosed” na página 267.

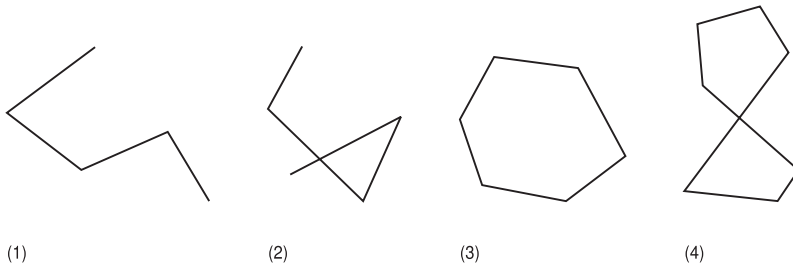


Figura 10. Objetos da cadeia de linha.

1. Uma cadeia de linha não-fechada simples.
2. Uma cadeia de linha não-simples, não-fechada.
3. Uma cadeia de linha simples fechada e, portanto, um anel.
4. Uma cadeia de linha fechada, não-simples. Não é um anel.

Polígonos

Um polígono é uma superfície bidimensional armazenada como uma seqüência de pontos definindo seu anel de limite exterior e 0 ou mais anéis interiores. Os anéis de um polígono não podem se sobrepor. Portanto, por definição, os polígonos sempre serão simples. Geralmente eles definem pedaços de terra, rios e outros recursos que possuam uma extensão espacial.

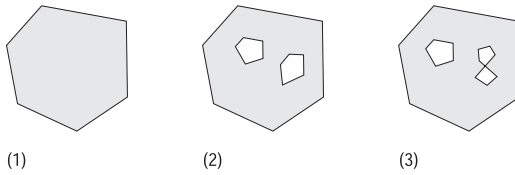


Figura 11. Polígonos.

1. Um polígono cujo limite esteja definido por um anel exterior.
2. Um polígono cujo limite esteja definido por um anel exterior e dois anéis interiores. A área dentro dos anéis interiores fazem parte do exterior dos polígonos.
3. Um polígono permitido, pois os anéis fazem interseção num único ponto da tangente.

Os anéis externo e interno definem o limite de um polígono e o espaço entre os anéis define o interior do polígono. Os anéis de um polígono podem fazer interseção num ponto da tangente mas nunca se cruzarem. Além das outras propriedades herdadas da figura geométrica da superclasse, os polígonos possuem área.

Funções que operam em polígonos:

ST_Area

Toma um polígono e retorna sua área como um número de precisão dupla. Para obter mais informações, consulte “ST_Area” na página 222.

ST_ExteriorRing

Toma um polígono e retorna seu anel externo como uma cadeia de linha. Para obter mais informações, consulte “ST_ExteriorRing” na página 250.

ST_NumInteriorRing

Toma um polígono e retorna o número de anéis internos que ele contém. Para obter mais informações, consulte “ST_NumInteriorRing” na página 289.

ST_InteriorRingN

Toma um polígono e um índice e retorna o anel interno n como uma cadeia de linha. Para obter mais informações, consulte “ST_InteriorRingN” na página 259.

ST_Centroid

Toma um polígono e retorna um ponto que é o centro da extensão do polígono. Para obter mais informações, consulte “ST_Centroid” na página 230.

ST_PointOnSurface

Toma um polígono e retorna um ponto que tem garantia de estar na superfície do polígono. Para obter mais informações, consulte “ST_PointOnSurface” na página 300.

ST_Perimeter

Toma um polígono e retorna o perímetro de sua superfície. Para obter mais informações, consulte “ST_Perimeter” na página 294.

MultiPontos

MultiPonto é uma coleção de pontos e, como seus elementos, possui uma dimensão 0. Um MultiPonto será simples se nenhum de seus elementos ocupar o mesmo espaço da coordenada. O limite de um MultiPonto é NULL. Os MultiPontos podem ser utilizados para definir fenômenos, como padrões de difusão aérea e incidentes de um surto epidêmico.

Funções que operam em MultiPontos:

ST_NumGeometries

Toma uma coleção homogênea e retorna o número de elementos da figura geométrica básica que contém. Para obter mais informações, consulte “ST_NumGeometries” na página 288.

ST_GeometryN

Toma uma coleção homogênea e um índice e retorna a figura geométrica básica nth. Para obter mais informações, consulte “ST_GeometryN” na página 252.

Cadeias de linhas múltiplas

Uma cadeia de linhas múltiplas é uma coleção de cadeias de linhas. As cadeias de linhas múltiplas são simples se elas fizerem interseção nos pontos de extremidade dos elementos da cadeia de linha. As cadeias de linhas múltiplas serão não-simples se os interiores dos elementos da cadeia de linhas fizerem interseção.

O limite de uma cadeia de linhas múltiplas será o ponto de extremidade não-interceptado dos elementos da cadeia de linhas. A cadeia de linhas múltiplas será fechada se todos os elementos da cadeia de linhas estiverem fechados. O limite de uma cadeia de linhas múltiplas será NULL se todos os pontos de extremidade de todos os elementos fizerem interseção. Além das outras propriedades herdadas da figura geométrica da superclasse, as cadeias de linhas múltiplas têm comprimento. As cadeias de linhas múltiplas são usadas para definir fluxos ou redes de estradas.

Funções que operam em cadeias de linhas múltiplas:

ST_Length

Toma uma cadeia de linhas múltiplas e retorna o comprimento cumulativo de todos os elementos da cadeia de linhas como um número de precisão dupla. Para obter mais informações, consulte “ST_Length” na página 276.

ST_IsClosed

Toma uma cadeia de linhas múltiplas e retorna 1 (VERDADEIRO) se a cadeia de linhas múltiplas estiver fechada e, do contrário, 0 (FALSO). Para obter mais informações, consulte “ST_IsClosed” na página 267.

ST_NumGeometries

Toma uma coleção homogênea e retorna o número de elementos da figura geométrica básica que contém. Para obter mais informações, consulte “ST_NumGeometries” na página 288.

ST_GeometryN

Toma uma coleção homogênea e um índice e retorna a figura geométrica básica nth. Para obter mais informações, consulte “ST_GeometryN” na página 252.

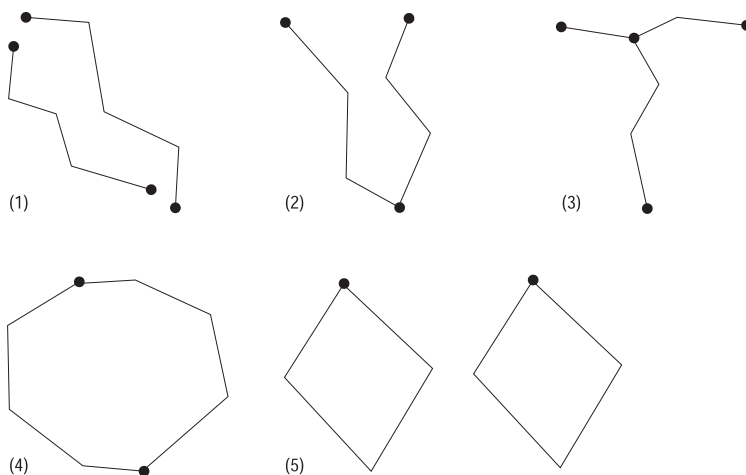


Figura 12. Cadeias de linhas múltiplas.

1. Uma cadeia de linhas múltiplas cujo limite esteja definido pelos quatro pontos de extremidade de seus dois elementos da cadeia de linhas.
2. Uma cadeia de linhas múltiplas simples somente porque os pontos de extremidade dos elementos da cadeia de linhas fizerem interseção. O limite é definido por dois pontos de extremidade que não fazem interseção.
3. Uma cadeia de linha não-simples porque o interior de um dos elementos da cadeia de linha está interceptado. O limite desta cadeia de linhas múltiplas está definido pelos quatro pontos de extremidade, inclusive o ponto de interseção.
4. Uma cadeia de linhas múltiplas não-fechada. Não está fechada porque suas cadeias de linha de elementos não estão fechadas. É simples porque nenhum interior das cadeias de linhas de elementos fazem interseção.
5. Uma cadeia de linhas múltiplas fechada. Está fechada porque todos os elementos estão fechados. É simples porque nenhum de seus elementos fazem interseção nos interiores.

Multipolígonos

O limite de um multipolígono é o comprimento cumulativo dos anéis externo e interno de seu elemento. O interior de um multipolígono é definido como o interior cumulativo dos polígonos do elemento. O limite dos elementos de um multipolígono podem fazer interseção somente no ponto de tangência. Além das outras propriedades herdadas do tipo `ST_MultiSurface`, os multipolígonos possuem área. Os multipolígonos definem recursos como uma camada da florestas ou uma parte não-contígua de terra, como uma cadeia de ilhas.

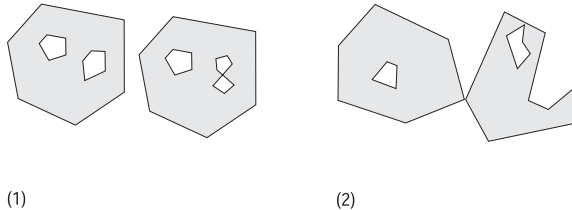


Figura 13. Multipolígonos.

1. Um multipolígono com dois elementos do polígono. O limite é definido por dois anéis externos e três internos.
2. Um multipolígono com dois elementos do polígono. O limite é definido por dois anéis externos e dois internos. Os dois elementos do polígono fazem interseção num ponto da tangente.

Funções que operam em multipolígonos:

ST_Area

Toma um multipolígono e retorna a área cumulativa dos elementos do polígono como um número de precisão dupla. Para obter mais informações, consulte “`ST_Area`” na página 222.

ST_Centroid

Toma um multipolígono e retorna um ponto que é seu centro geométrico dimensionado. Para obter mais informações, consulte “`ST_Centroid`” na página 230.

ST_NumGeometries

Toma uma coleção homogênea e retorna o número de elementos da figura geométrica básica que contém. Para obter mais informações, consulte “`ST_NumGeometries`” na página 288.

ST_GeometryN

Toma uma coleção homogênea e um índice e retorna a figura geométrica básica nth. Para obter mais informações, consulte “`ST_GeometryN`” na página 252.

Funções que mostram relações e comparações, geram figuras geométricas, e convertem formatos de valores

As seções precedentes apresentam três categorias de funções espaciais:

- Funções associadas às propriedades da figura geométrica
- Funções associadas a figuras geométricas específicas

Esta seção apresenta mais três categorias:

- Funções que determinam formas nas quais os recursos geográficos se relacionam ou se comparam
- Funções que geram novas figuras geométricas
- Funções que convertem os valores de uma figura geométrica em um formato que pode ser importado ou exportado

Funções que mostram relações ou comparações entre recursos geográficos

Várias funções espaciais retornam informações sobre formas em que os recursos geográficos se relacionam ou se comparam. A maioria destas funções retornam um valor inteiro. Esta seção descreve as funções de predicados em geral e discute cada função individualmente.

Funções do predicado

As funções de predicado retornam 1 (VERDADEIRO) se uma comparação atender aos critérios da função, ou 0 (FALSO) se a comparação falhar. Os predicados que testam uma relação espacial comparam pares de figuras geométricas que podem ser de um tipo ou dimensão diferentes.

Os predicados comparam as coordenadas X e Y das figuras geométricas submetidas. As coordenadas Z e a medida (se existirem) serão ignoradas. Isto permite que as figuras geométricas tenham coordenadas Z ou medida para comparação com aquelas que não têm.

O *Modelo de Interseção 9 Dimensionalmente Estendido (DE-9IM)*¹ é uma abordagem matemática que define a relação espacial pair-wise entre figuras geométricas de tipos e dimensões diferentes. Este modelo expressa relações espaciais entre todos os tipos de figura geométrica como interseções pair-wise de seu interior, limite e exterior e consideração com a dimensão das interseções resultantes.

1. O DE-9IM foi desenvolvido por Clementini e Felice, que estenderam dimensionalmente o Modelo de Interseção 9 de Egenhofer e Herring. O DE-9IM é colaboração de quatro autores, Clementini, Eliseo, Di Felice e van Osstrom. Eles publicaram o modelo em "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," D. Abel e B.C. Ooi (Ed.), *Advances in Spatial Database — no Terceiro Simpósio Internacional. SSD '93*. LNCS 692. Páginas 277 a 295. O modelo de Interseção 9 desenvolvido por Springer-Verlag Singapore (1993) Egenhofer M.J. e Herring, J., foi publicado em "Categorizing binary topological relationships between regions, lines, and points in geographic databases," *Tech. Report, Department of Surveying Engineering*, Universidade de Maine, Orono, ME 1991.

As figuras geométricas a e b : $I(a)$, $B(a)$, e $E(a)$ representam o interior, o limite e o exterior de a , respectivamente. $I(b)$, $B(b)$ e $E(b)$ representam o interior, o limite e o exterior de b . As interseções de $I(a)$, $B(a)$, e $E(a)$ com $I(b)$, $B(b)$ e $E(b)$ produzem uma matriz 3 por 3. Cada interseção pode resultar em figuras geométricas de dimensões diferentes. Por exemplo, a interseção dos limites de dois polígonos consiste num ponto e numa cadeia de linha. Nesse caso, a função \dim retornará a dimensão máxima de 1.

A função \dim retorna um valor de 1, 0, 1 ou 2. 1 corresponde ao conjunto nulo ou $\dim(\text{null})$, que é retornado quando nenhuma interseção é encontrada.

| | Interior | Limite | Exterior |
|-----------------|------------------------|------------------------|------------------------|
| Interior | $\dim(I(a) \cap I(b))$ | $\dim(I(a) \cap B(b))$ | $\dim(I(a) \cap E(b))$ |
| Limite | $\dim(B(a) \cap I(b))$ | $\dim(B(a) \cap B(b))$ | $\dim(B(a) \cap E(b))$ |
| Exterior | $\dim(E(a) \cap I(b))$ | $\dim(E(a) \cap B(b))$ | $\dim(E(a) \cap E(b))$ |

Os resultados dos predicados da relação espacial podem ser entendidos ou verificados através da comparação de resultados do predicado com a matriz padrão que representa os valores aceitáveis para DE-9IM.

A matriz padrão contém os valores aceitáveis para cada célula da matriz de interseção. Os valores padrão possíveis são:

- T** Deve existir uma interseção, $\dim = 0, 1$ ou 2 .
- F** Não deve existir uma interseção, $\dim = -1$.
- *** Não importa se existe uma interseção, $\dim = -1, 0, 1$ ou 2 .
- 0** Deve existir uma interseção e sua dimensão máxima deve ser 0 , $\dim = 0$.
- 1** Deve existir uma interseção e sua dimensão máxima deve ser 1 , $\dim = 1$.
- 2** Deve existir uma interseção e sua dimensão máxima deve ser 2 , $\dim = 2$.

Por exemplo a seguinte matriz padrão do predicado ST_Within inclui os valores T, F e *.

Tabela 41. Matriz para ST_Within . A matriz padrão do predicado ST_Within para combinações de figura geométrica.

| | | b | | |
|----------|-----------------|-----------------|---------------|-----------------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | F |
| | Limite | * | * | F |
| | Exterior | * | * | * |

O predicado `ST_Within` retorna VERDADEIRO quando os interiores das duas figuras geométricas fazem interseção e quando o interior e o limite de *a* não faz interseção com o exterior de *b*. Todas as outras condições não importam.

Cada predicado tem pelo menos uma matriz padrão, mas alguns exigem mais que uma para descrever as relações de várias combinações de tipo de figura geométrica.

ST_Equals

`ST_Equals` retorna 1 (VERDADEIRO) se duas figuras geométricas do mesmo tipo tiverem valores idênticos das coordenadas X,Y.





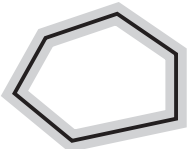
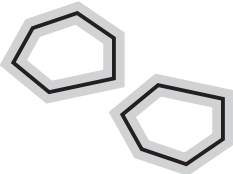
| | |
|---|---|
|  |  |
| ponto / ponto | multi ponto / multi ponto |
|  |  |
| cad. de linhas / cad. de linhas | cad.multi linhas / cad. multi linhas |
|  |  |
| polígono / polígono | multi polígono / multi polígono |

Figura 14. `ST_Equals`. As figuras geométricas serão iguais se possuírem coordenadas X,Y correspondentes.

Tabela 42. Matriz para igualdade. A matriz padrão DE-9IM para igualdade assegura que os interiores fazem interseção e que nenhuma parte ou limite interior da figura geométrica faz interseção com o exterior de outra.

| | | b | | |
|----------|-----------------|-----------------|---------------|-----------------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | F |
| | Limite | * | * | F |
| | Exterior | F | F | * |

Para obter mais informações, consulte “ST_Equals” na página 249.

ST_OrderingEquals

ST_OrderingEquals compara duas figuras geométricas e retorna 1 (VERDADEIRO) se as figuras geométricas forem iguais e as coordenadas estiverem na mesma ordem; do contrário, retornará 0 (FALSO). Para obter mais informações, consulte “ST_OrderingEquals” na página 291.

ST_Disjoint

ST_Disjoint retorna 1 (VERDADEIRO) se a interseção das duas figuras geométricas for um conjunto vazio.

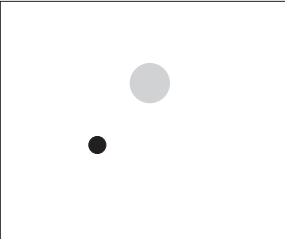
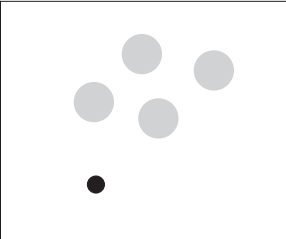
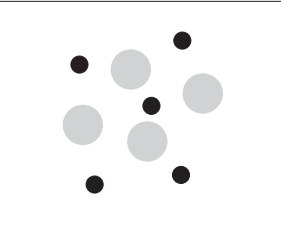
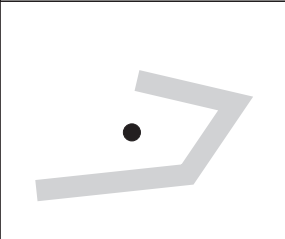
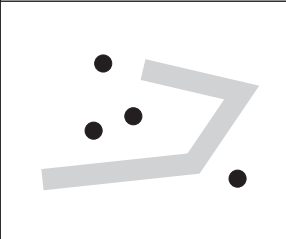

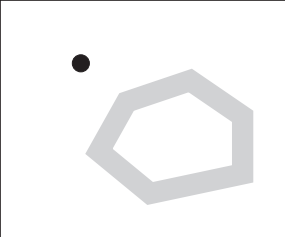
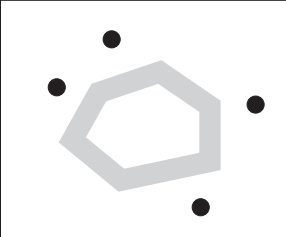
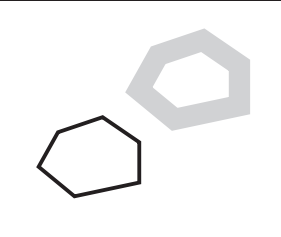
| | | |
|---|---|--|
|  |  |  |
| ponto / ponto | ponto / multi ponto | multi ponto / multi ponto |
|  |  |  |
| ponto / cadeia de linhas | cad. multi linhas / cad. de linhas | polígono / cadeia de linhas |
|  |  |  |
| ponto / polígono | multi ponto / multi polígono | polígono / polígono |

Figura 15. *ST_Disjoint*. As figuras geométricas serão desunidas se não fizerem interseção entre si de alguma forma.

Tabela 43. Matriz para *ST_Disjoint*. A matriz padrão do predicado *ST_Disjoint* simples indica que nem o interior nem o limite da figura geométrica fazem interseção.

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | F | F | * |
| | Limite | F | F | * |
| | Exterior | * | * | * |

Para obter mais informações, consulte “*ST_Disjoint*” na página 242.

ST_Intersects

ST_Intersects retorna 1 (VERDADEIRO) se a interseção não resultar em um conjunto vazio. A interseção retorna o resultado oposto exato de *ST_Disjoint*.

O predicado `ST_Intersects` retorna VERDADEIRO se as condições de quaisquer matrizes padrão seguintes retornarem VERDADEIRO.

Tabela 44. Matriz para `ST_Intersects` (1). O predicado `ST_Intersects` retorna VERDADEIRO se os interiores das duas figuras geométricas fizerem interseção.

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | * |
| | Limite | * | * | * |
| | Exterior | * | * | * |

Tabela 45. Matriz para `ST_Intersects` (2). O predicado `ST_Intersects` retorna VERDADEIRO se o limite da primeira figura geométrica fizer interseção com o limite da segunda.

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | * | T | * |
| | Limite | * | * | * |
| | Exterior | * | * | * |

Tabela 46. Matriz para `ST_Intersects` (3). O predicado `ST_Intersects` retorna VERDADEIRO se o limite da primeira figura geométrica fizer interseção com o interior da segunda.

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | * | * | * |
| | Limite | T | * | * |
| | Exterior | * | * | * |

Tabela 47. Matriz para `ST_Intersects` (4). O predicado `ST_Intersects` retorna VERDADEIRO se os limites de qualquer figura geométrica fizerem interseção.

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | * | * | * |
| | Limite | * | T | * |
| | Exterior | * | * | * |

Para obter mais informações, consulte “`ST_Intersects`” na página 266.

EnvelopesIntersect

Esta função retorna 1 (VERDADEIRO) se os envelopes das duas figuras geométricas fizerem interseção. É uma função de conveniência que implementa eficientemente `ST_Intersects` (`ST_Envelope(g1),ST_Envelope(g2)`). Para obter mais informações, consulte “`EnvelopesIntersect`” na página 199.

ST_Touches

ST_Touches retorna 1 (VERDADEIRO) se nenhum dos pontos comuns a ambas as figuras geométricas fizerem interseção com os interiores das duas. Pelo menos uma figura geométrica deve ser uma cadeia de linhas, polígono, cadeia de linhas múltiplas ou multipolígonos.

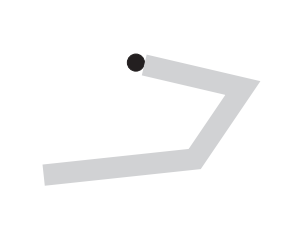
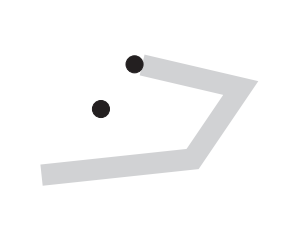

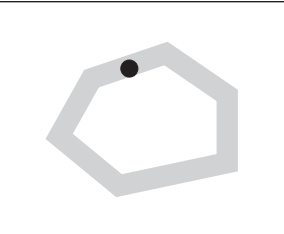
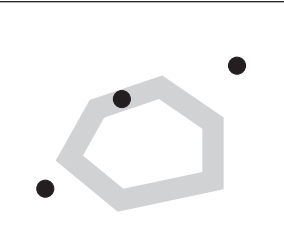
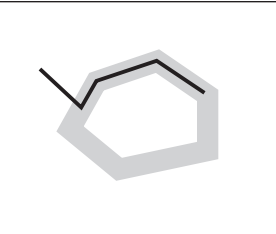
| | | |
|---|---|--|
|  |  |  |
| ponto / cadeia de linhas | multi ponto / cadeia de linhas | cad. de linhas / cad. de linhas |
|  |  |  |
| ponto / polígono | multi ponto / polígono | cadeia de linhas / polígono |

Figura 16. ST_Touches

As matrizes padrão mostram que o predicado ST_Touches retorna VERDADEIRO quando os interiores da figura geométrica não fizerem interseção e o limite de uma delas fizer interseção com o interior ou limite de outra.

Tabela 48. Matriz para ST_Touches (1)

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | F | T | * |
| | Limite | * | * | * |
| | Exterior | * | * | * |

Tabela 49. Matriz para ST_Touches (2)

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | F | * | * |
| | Limite | T | * | * |
| | Exterior | * | * | * |

Tabela 50. Matriz para ST_Touches (3)

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | F | * | * |
| | Limite | * | T | * |
| | Exterior | * | * | * |

Para obter mais informações, consulte “ST_Touches” na página 311.

ST_Overlaps

ST_Overlaps compara duas figuras geométricas da mesma dimensão. Ela retorna 1 (VERDADEIRO) se o conjunto de interseção resultar numa figura geométrica diferente de ambas, mas que tenha a mesma dimensão.

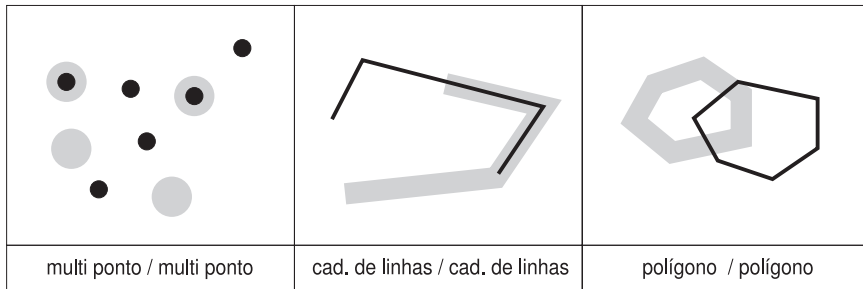


Figura 17. ST_Overlaps

A matriz padrão em Tabela 51 se aplica às sobreposições polígono/polígono, MultiPonto/MultiPonto e multipolígono/multipolígono. Para estas combinações, o predicado de sobreposição retornará VERDADEIRO se o interior de ambas figuras geométricas fizerem interseção com o interior e exterior de outras.

Tabela 51. Matriz para ST_Overlaps (1)

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | T |
| | Limite | * | * | * |
| | Exterior | T | * | * |

A matriz padrão na Tabela 52 na página 180 aplica-se a sobreposições de cadeia de linhas/cadeia de linhas e cadeia de linhas múltiplas/cadeia de linhas múltiplas. Neste caso, a interseção das figuras geométricas deverá resultar numa figura geométrica que tenha uma dimensão 1 (outra cadeia de linha). Se a dimensão da interseção dos interiores for 1, o predicado ST_Overlaps retornaria FALSO, no entanto, o predicado ST_Crosses retornaria

VERDADEIRO.

Tabela 52. Matriz para ST_Overlaps (2)

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | 1 | * | T |
| | Limite | * | * | * |
| | Exterior | T | * | * |

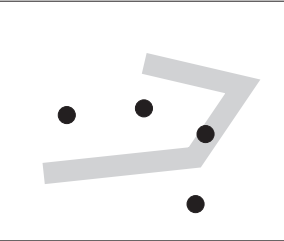
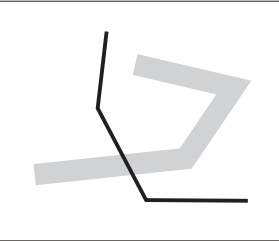
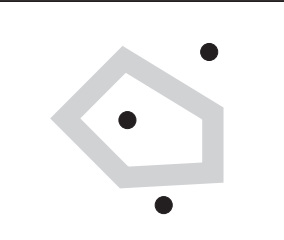

Para obter mais informações, consulte “ST_Overlaps” na página 292.

ST_Crosses

ST_Crosses recebe duas figuras geométricas e retorna 1 (TRUE) se:

- A interseção resulta em uma figura geométrica cuja dimensão é menor que a dimensão máxima das figuras geométricas de origem.
- O conjunto de interseções é interno para ambas as figuras geométricas de origem.

ST_Crosses retorna 1 (TRUE) apenas para as comparações MultiPonto/polígono, MultiPonto/cadeia de linhas, cadeia de linhas/cadeia de linhas, cadeia de linhas/polígono e cadeia de linhas/multipolígono.

| | |
|---|---|
|  |  |
| multi ponto / cadeia de linhas | cad. de linhas / cad. de linhas |
|  |  |
| multi ponto / polígono | cadeia de linhas / polígono |

A matriz padrão em Tabela 53 na página 181 se aplica ao MultiPonto/cadeia de linhas, MultiPonto/cadeia de linhas múltiplas, MultiPonto/polígono, MultiPonto/multipolígono, cadeia de linhas/polígono, cadeia de linhas/multipolígono. A matriz indica que os interiores devem fazer

interseção e que o interior da primária (figura geométrica a) deve fazer interseção com o interior da secundária (figura geométrica b).

Tabela 53. Matriz para $ST_Crosses$ (1)

| | | b | | |
|----------|-----------------|-----------------|---------------|-----------------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | T |
| | Limite | * | * | * |
| | Exterior | * | * | * |

A matriz padrão na Tabela 54 aplica-se à cadeia de linhas/cadeia de linhas, cadeia de linhas/cadeia de linhas múltiplas e cadeia de linhas múltiplas/cadeia de linhas múltiplas. A matriz indica que a dimensão da interseção dos interiores deve ser 0 (interseção num ponto). Se a dimensão desta interseção for 1 (interseção numa cadeia de linha), o predicado $ST_Crosses$ retornará FALSO; no entanto, o predicado $ST_Overlaps$ retornará VERDADEIRO.

Tabela 54. Matriz para $ST_Crosses$ (2)

| | | b | | |
|----------|-----------------|-----------------|---------------|-----------------|
| | | Interior | Limite | Exterior |
| a | Interior | 0 | * | * |
| | Limite | * | * | * |
| | Exterior | * | * | * |

Para obter mais informações, consulte “ $ST_Crosses$ ” na página 237.

ST_Within

ST_Within retorna 1 (VERDADEIRO) se a primeira figura geométrica estiver completamente dentro da segunda. ST_Within retorna o resultado oposto exato de $ST_Contains$.

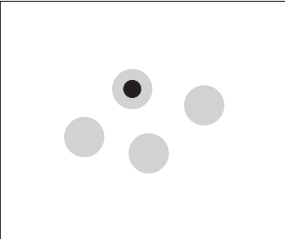
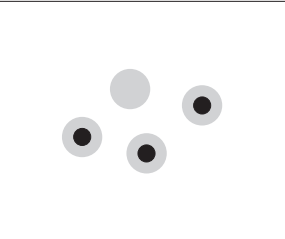
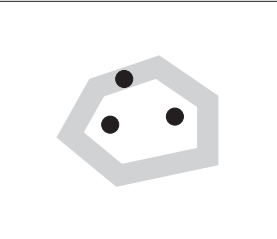
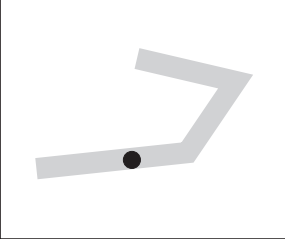
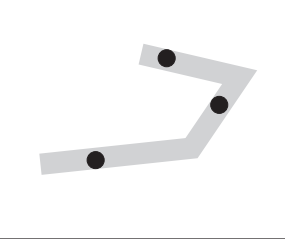
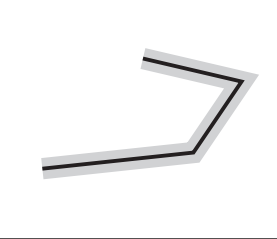
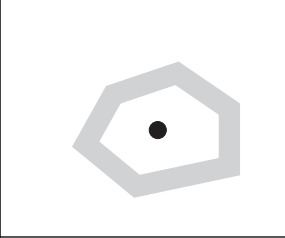
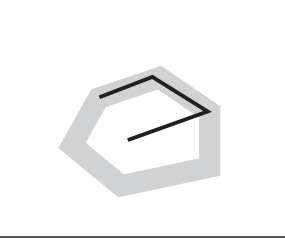
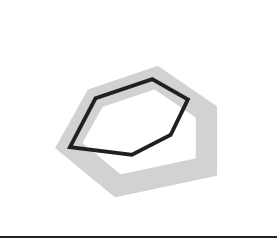
| | | |
|---|---|--|
|  |  |  |
| ponto / multi ponto | multi ponto / multi ponto | multi ponto / polígono |
|  |  |  |
| ponto / cadeia de linhas | multi ponto / cadeia de linhas | cad. de linhas / cad. de linhas |
|  |  |  |
| ponto / polígono | cadeia de linhas / polígono | polígono / polígono |

Figura 18. *ST_Within*

A matriz do predicado *ST_Within* indica que os interiores das duas figuras geométricas devem fazer interseção e que o interior e o limite da figura geométrica primária (figura geométrica *a*) não deve fazer interseção com o exterior da segunda (figura geométrica *b*).

Tabela 55. Matriz para *ST_Within*

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | F |
| | Limite | * | * | F |
| | Exterior | * | * | * |

Para obter mais informações, consulte “*ST_Within*” na página 315.

ST_Contains

ST_Contains retorna 1 (VERDADEIRO) se a segunda figura geométrica estiver completamente dentro da primeira. O predicado ST_Contains retorna o resultado oposto exato do predicado ST_Within.










| | | |
|--|--|---|
|  |  |  |
| multi ponto / ponto | multi ponto / multi ponto | polígono / multi ponto |
|  |  |  |
| cadeia de linhas / ponto | cadeia de linhas / multi ponto | cad. de linhas / cad. de linhas |
|  |  |  |
| polígono / ponto | polígono / cadeia de linhas | polígono / polígono |

Figura 19. ST_Contains

A matriz padrão do predicado ST_Contains indica que o interior das duas figuras geométricas deve fazer interseção e que o interior e limite da secundária (figura geométrica *b*) não deve fazer interseção com o exterior da primária (figura geométrica *a*).

Tabela 56. Matriz para ST_Contains

| | | b | | |
|---|----------|----------|--------|----------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | * |
| | Limite | * | * | * |
| | Exterior | F | F | * |

Para obter mais informações, consulte “ST_Contains” na página 231.

ST_Relate

A função ST_Relate compara duas figuras geométricas e retornará 1 (VERDADEIRO) se as figuras geométricas atenderem às condições especificadas pela cadeia de matrizes padrão DE-91M, do contrário, a função retornará 0 (FALSO). Para obter mais informações, consulte “ST_Relate” na página 305.

ST_Distance

A função ST_Distance informa a distância mínima que separa dois recursos desunidos. Se os recursos não estiverem desunidos, a função irá informar uma distância mínima de 0.

Por exemplo, ST_Distance poderia informar a menor distância entre dois locais que uma aeronave deveria viajar. A Figura 20 ilustra estas informações.



Figura 20. Distância mínima entre duas cidades. ST_Distance pode tomar as coordenadas para as cidades de Los Angeles e Chicago como entrada e retornar um valor indicando a distância mínima entre estes locais.

Para obter mais informações, consulte “ST_Distance” na página 244.

Funções que geram novas figuras geométricas a partir de existentes

O Spatial Extender fornece predicados e funções de transformação que geram novas figuras geométricas a partir de existentes.

ST_Intersection

A função ST_Intersection retorna o conjunto de interseção de duas figuras geométricas. Ele sempre retorna como uma coleção que indica a dimensão mínima das figuras geométricas de origem. Por exemplo, para uma cadeia de linhas que faça interseção com um polígono, a função de interseção retorna uma cadeia de linhas múltiplas composta dessa parte da cadeia de linhas

comum ao interior e ao limite do polígono. A cadeia de linhas múltiplas conterá mais de uma cadeia de linhas se a cadeia de linhas de origem fizer interseção com o polígono com dois ou mais segmentos descontínuos. Se as figuras geométricas não fizerem interseção ou se a interseção resultar numa dimensão menor que ambas as figuras geométricas de origem, será retornada uma figura geométrica vazia.

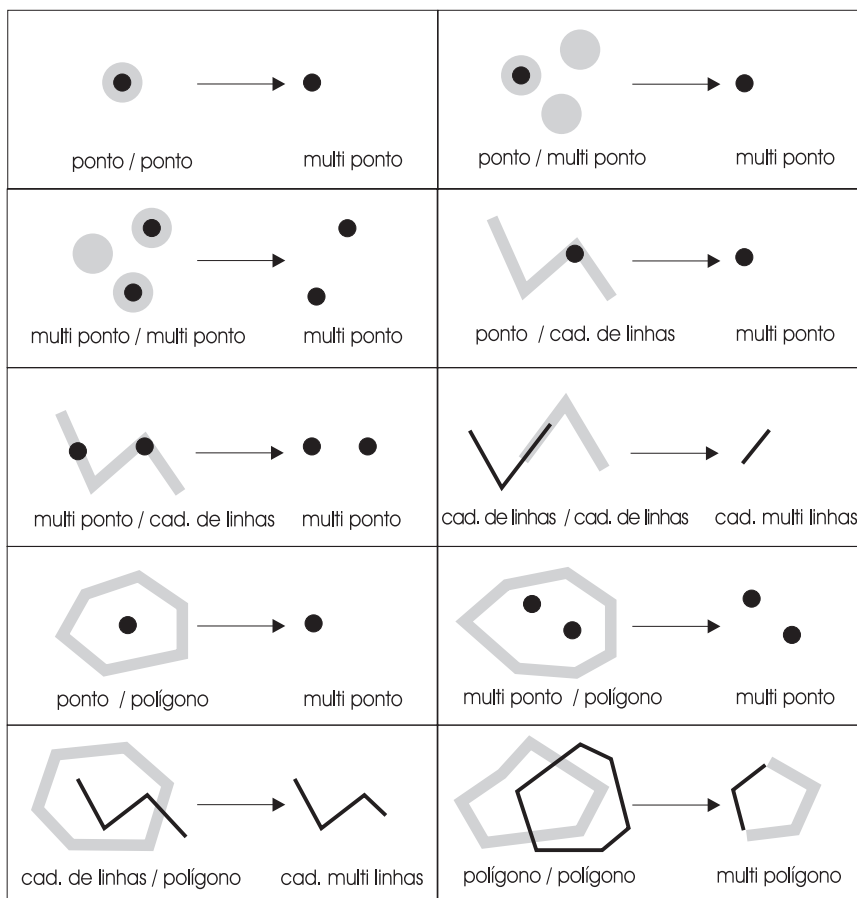


Figura 21. *ST_Intersection*. Exemplos da função *ST_Intersection*.

Para obter mais informações, consulte “*ST_Intersection*” na página 264.

ST_Difference

ST_Difference recebe duas figuras geométricas como entrada. A primeira é chamada *figura geométrica primária*; a segunda, a *figura geométrica secundária*. A função *ST_Difference* retorna a parte da figura geométrica primária que não fez interseção com a segunda figura geométrica. Este é o E NÃO (AND NOT) lógico de espaço. A função *ST_Difference* opera somente em figuras

geométricas de dimensão semelhante e retorna uma coleção que tem a mesma dimensão que as figuras geométricas de origem. No caso de as figuras geométricas serem iguais, será retornado uma figura geométrica vazia. Se as dimensões das figuras geométricas fornecidas ao ST_Difference como entrada não corresponderem, o ST_Difference retornará um nulo.

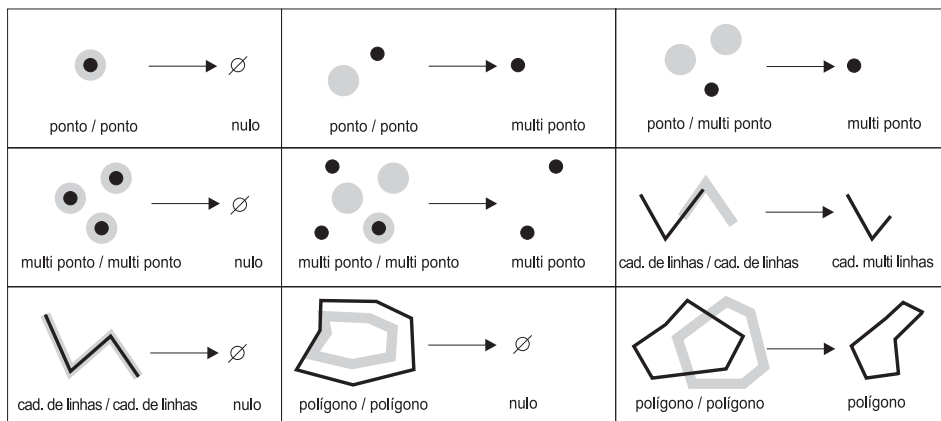


Figura 22. ST_Difference

Para obter mais informações, consulte “ST_Difference” na página 239.

ST_Union

A função ST_Union retorna o conjunto de união de duas figuras geométricas. Este é o OU (OR) lógico de espaço. As figuras geométricas de origem deve ser de dimensão semelhante. ST_Union sempre retorna o resultado como uma coleção.

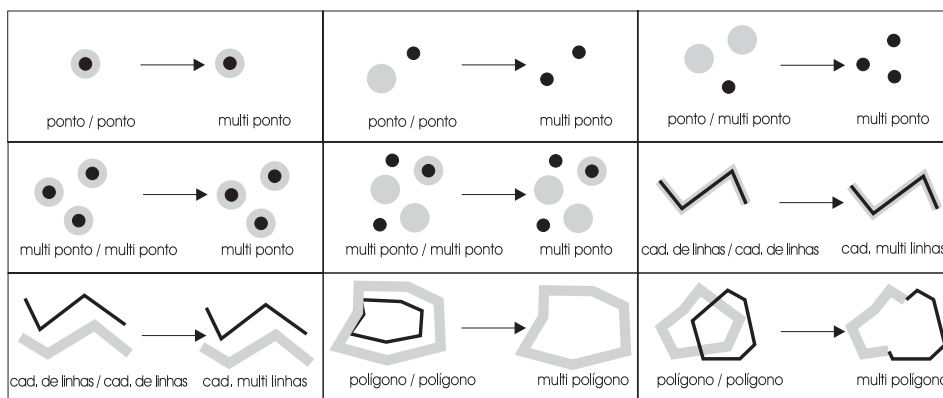


Figura 23. ST_Union

Para obter mais informações, consulte “ST_Union” na página 314.

ST_SymmetricDiff

A função `ST_SymmetricDiff` retorna a diferença simétrica (a lógica booleana XOR de espaço) de duas figuras geométricas de intersecção que possuem a mesma dimensão. Se essas figuras geométricas forem iguais, o `ST_SymmetricDiff` retornará uma figura geométrica vazia. Se forem diferentes, uma parte de uma delas ou de ambas ultrapassará a área de intersecção. O `ST_SymmetricDiff` retornará a parte de não-intersecção ou as partes como uma coleção; por exemplo, como um multipolígono.

Se `ST_SymmetricDiff` fornecer figuras geométricas de diferentes dimensões como entrada, ele retornará um nulo.

ST_Buffer

A função `ST_Buffer` gera uma figura geométrica englobando uma figura geométrica numa distância especificada. Um polígono resultará quando uma figura geométrica primária estiver em buffer, sempre que os elementos de uma coleção estiverem próximos o suficiente, de modo que todos os polígonos do buffer estejam sobrepostos. No entanto, quando há separação suficiente entre os elementos de uma coleção em buffer, resultarão polígonos individuais do buffer indicando, nesse caso, que a função `ST_Buffer` retorna multipolígonos.

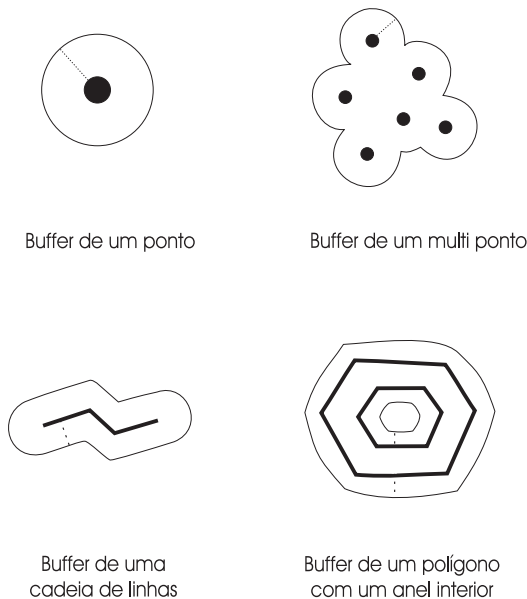


Figura 24. `ST_Buffer`

A função `ST_Buffer` aceita distâncias positivas e negativas, no entanto, somente figuras geométricas com dimensão dois (polígonos e multipolígonos) aplicam um buffer negativo. O valor absoluto da distância do buffer é usado

sempre que a dimensão da figura geométrica de origem for menor que 2 (todas as figuras geométricas que não sejam polígonos ou multipolígonos).

Em geral, para anéis externos, as distâncias do buffer positivo geram anéis de polígonos que estejam distantes do centro da figura geométrica de origem; distâncias de buffer negativo geram anéis de polígonos ou multipolígonos em direção ao centro. Para anéis internos de um polígono ou multipolígonos, a distância de um buffer positivo gera um anel de buffer em direção ao centro e uma distância de um buffer negativo gera um anel de buffer a partir do centro.

O processo de buffer combina polígonos que se sobrepõem. As distâncias negativas maiores que metade da largura máxima do interior de um polígono resultam numa figura geométrica vazia.

Para obter mais informações, consulte “ST_Buffer” na página 228.

LocateAlong

Para figuras geométricas que contêm medidas, a localização de uma determinada medida pode ser encontrada com a função `LocateAlong`. `LocateAlong` retorna a localização como um `MultiPonto`. Se a dimensão da figura geométrica de origem for zero 0 (por exemplo, um ponto e um `MultiPonto`), será necessária uma correspondência exata e esses pontos que têm um valor de medida correspondente serão retornados como um `MultiPonto`. Contudo, para figuras geométricas de origem cuja dimensão seja maior que 0, a localização será interpolada. Por exemplo, se o valor de medida digitado for 5,5 e as medidas nos vértices de uma cadeia de linha forem 3, 4, 5, 6 e 7 respectivamente, o ponto interpolado que ficar exatamente entre os vértices com valores de medida 5 e 6 retornarão.

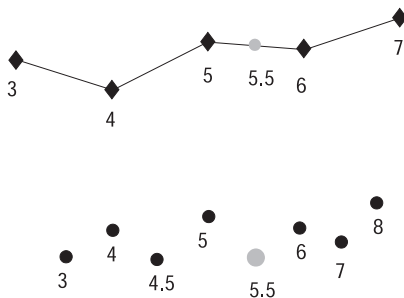


Figura 25. `LocateAlong`

Para obter mais informações, consulte “`LocateAlong`” na página 207.

LocateBetween

A função `LocateBetween` retorna o conjunto de caminhos ou localizações que ficam entre dois valores de medida de uma figura geométrica de origem que tenha medidas. Se a dimensão da figura geométrica de origem for 0, `LocateBetween` retornará um `MultiPonto` contendo todos os pontos cujas medidas estejam entre as duas medidas de origem. Para figuras geométricas de origem cuja dimensão é maior que 0, `LocateBetween` retornará uma cadeia de linhas múltiplas se for possível interpolar um caminho; do contrário, `LocateBetween` retornará um `MultiPonto` contendo as localizações do ponto. Um ponto vazio retornará sempre que `LocateBetween` não conseguir interpolar um caminho ou encontrar uma localização entre as medidas. `LocateBetween` executa uma pesquisa inclusiva das figuras geométricas; portanto as medidas das figuras geométricas devem ser maiores ou iguais à medida de *origem* e menores ou iguais à medida de *destino*.

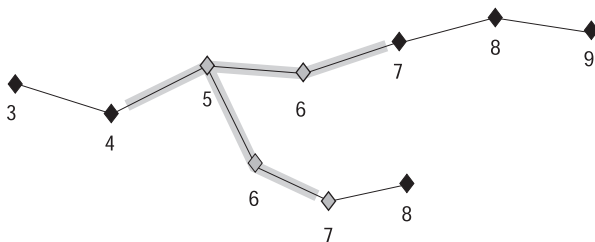


Figura 26. `LocateBetween`

Para obter mais informações, consulte “`LocateBetween`” na página 209.

ST_ConvexHull

A função `ST_ConvexHull` retorna o polígono de casca convexa de qualquer figura geométrica que tenha pelo menos três vértices formando um convexo. Se os vértices da figura geométrica não formarem um convexo, `ST_ConvexHull` retornará um nulo. `ST_ConvexHull` geralmente é a primeira etapa em mosaico utilizada para criar uma rede TIN a partir de um conjunto de pontos.

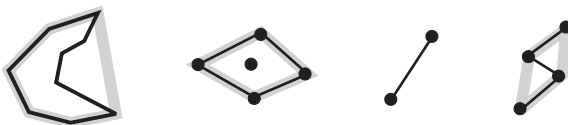


Figura 27. `ST_ConvexHull`

Para obter mais informações, consulte “`ST_ConvexHull`” na página 233.

ST_Polygon

Gera um polígono a partir de uma cadeia de linha. Para obter mais informações, consulte “ST_Polygon” na página 304.

Funções que convertem o formato dos valores de uma figura geométrica

O Spatial Extender suporta três formatos de troca de dados GIS:

- Representação de texto reconhecida
- Representação do modo binário reconhecida
- Representação de formato ESRI

Representação de texto reconhecida

O Spatial Extender possui várias funções que geram figuras geométricas a partir de descrições de textos.

ST_WKTToSQL

Cria uma figura geométrica a partir da representação de texto de qualquer tipo de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “ST_WKTToSQL” na página 318.

ST_GeomFromText

Cria uma figura geométrica a partir da representação de texto de qualquer tipo de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “ST_GeomFromText” na página 255.

ST_PointFromText

Cria um ponto a partir de uma representação de texto de ponto. Para obter mais informações, consulte “ST_PointFromText” na página 296.

ST_LineFromText

Cria uma cadeia de linhas a partir de uma representação de texto da cadeia de linhas. Para obter mais informações, consulte “ST_LineFromText” na página 278.

ST_PolyFromText

Cria um polígono a partir de uma representação do texto do polígono. Para obter mais informações, consulte “ST_PolyFromText” na página 301.

ST_MPointFromText

Cria um MultiPonto a partir de uma representação do MultiPonto. Para obter mais informações, consulte “ST_MPointFromText” na página 284.

ST_MLineFromText

Cria uma cadeia de linhas múltiplas a partir de uma representação da cadeia de linhas múltiplas. Para obter mais informações, consulte “ST_MLineFromText” na página 281.

ST_MPolyFromText

Cria um multipolígono a partir de uma representação de multipolígono. Para obter mais informações, consulte “ST_MPolyFromText” na página 286.

A representação de texto é uma cadeia ASCII. Ela permite que a figura geométrica seja trocada no formato de texto ASCII. Estas funções não exigem a definição de estruturas de programas especiais para mapear uma representação do modo binário. Portanto, podem ser usadas num programa 3GL ou 4GL.

A função ST_AsText converte o valor de uma figura geométrica existente em representação de texto. Para obter mais informações, consulte “ST_AsText” na página 225.

Para obter uma descrição detalhada de representações de texto reconhecidas, consulte “As representações de texto reconhecidas de OGC” na página 335.

Representação do modo binário reconhecida

O Spatial Extender possui várias funções que geram figuras geométricas a partir de representações do modo binário reconhecidas (WKB).

ST_WKBTtoSQL

Cria uma figura geométrica a partir de uma representação do modo binário reconhecida de qualquer tipo de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “ST_WKBTtoSQL” na página 316.

ST_GeomFromWKB

Cria uma figura geométrica a partir de uma representação do modo binário reconhecida de qualquer tipo de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “ST_GeomFromWKB” na página 257.

ST_PointFromWKB

Cria um ponto a partir de uma representação do modo binário reconhecida de um ponto. Para obter mais informações, consulte “ST_PointFromWKB” na página 297.

ST_LineFromWKB

Cria uma cadeia de linha a partir de uma representação do modo binário reconhecida de uma cadeia de linha. Para obter mais informações, consulte “ST_LineFromWKB” na página 279.

ST_PolyFromWKB

Cria um polígono a partir de uma representação do modo binário

reconhecida de um polígono. Para obter mais informações, consulte “ST_PolyFromWKB” na página 302.

ST_MPointFromWKB

Cria um MultiPonto a partir de uma representação binária convencional de um MultiPonto. Para obter mais informações, consulte “ST_MPointFromWKB” na página 285.

ST_MLineFromWKB

Cria uma cadeia de linhas múltiplas a partir de uma representação do modo binário reconhecida de uma cadeia de linhas múltiplas. Para obter mais informações, consulte “ST_MLineFromWKB” na página 282.

ST_MPolyFromWKB

Cria um multipolígono a partir de uma representação do modo binário reconhecida de um multipolígono. Para obter mais informações, consulte “ST_MPolyFromWKB” na página 287.

A representação do modo binário reconhecida é um fluxo contíguo de bytes. Ela permite que a figura geométrica sejam trocada entre um cliente ODBC e um banco de dados SQL no modo binário. Estas funções de figura geométrica exigem a definição das estruturas C para mapear a representação do modo binário. Portanto, elas destinam-se ao uso dentro de um programa 3GL e não se adequam a um ambiente 4GL.

A função ST_AsBinary converte um valor de figura geométrica existente em representação do modo binário reconhecida. Para obter mais informações, consulte “ST_AsBinary” na página 224.

Para obter uma descrição detalhada de representações do modo binário reconhecidas, consulte “As representações (WKB) do modo binário reconhecidas” na página 340.

Representação de formato ESRI

O Spatial Extender possui várias funções que geram figuras geométricas a partir de uma representação de formato ESRI. A representação de formato ESRI suporta coordenadas Z e medidas, além das representações bidimensionais suportadas pelas representações de texto e do modo binário reconhecidas.

ShapeToSQL

Cria uma figura geométrica a partir de um formato de qualquer tipo de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “ShapeToSQL” na página 220.

GeometryFromShape

Cria uma figura geométrica a partir de um formato de qualquer tipo

de figura geométrica. Nenhum identificador do sistema de referência espacial deverá ser especificado. Para obter mais informações, consulte “GeometryFromShape” na página 201.

PointFromShape

Cria um ponto a partir de um formato. Para obter mais informações, consulte “PointFromShape” na página 216.

LineFromShape

Cria uma cadeia de linhas a partir de um formato polilinha. Para obter mais informações, consulte “LineFromShape” na página 205.

PolyFromShape

Cria um polígono a partir de um formato de polilinha. Para obter mais informações, consulte “PolyFromShape” na página 218.

MPointFromShape

Cria um MultiPonto a partir de um formato do MultiPonto. Para obter mais informações, consulte “MPointFromShape” na página 214.

MLineFromShape

Cria uma cadeia de linhas múltiplas a partir de um formato de polilinha de várias partes. Para obter mais informações, consulte “MLine FromShape” na página 212.

MPolyFromShape

Cria um multipolígono a partir de um formato de polígono de multipartes. Para obter mais informações, consulte “MPolyFromShape” na página 215.

A sintaxe geral destas funções é a mesma. O primeiro argumento é a representação de formato fornecida como tipo de dados BLOB. O segundo argumento é o identificador de referência espacial que será atribuído à figura geométrica. Por exemplo, a função GeometryFromShape apresenta a seguinte sintaxe:

```
GeometryFromShape(shapegeometry, SRID)
```

Para mapear a representação do modo binário, estas funções de formato exigem a definição de estruturas C. Portanto, elas destinam-se ao uso dentro de um programa 3GL e não se adequam a um ambiente 4GL.

A função AsBinary converte um valor de figura geométrica em uma representação de formato ESRI. Para obter mais informações, consulte “AsShape” na página 198.

Para obter uma descrição detalhada de representações de formato, consulte “As representações de formato ESRI” na página 344.

Capítulo 14. Funções espaciais para consultas SQL

Este capítulo relaciona as funções disponíveis que você pode chamar quando consultar os dados espaciais. Cada uma das funções é descrita em uma seção que mostra a sintaxe, o tipo de retorno e os exemplos de código. Alguns dos exemplos deste capítulo incluem uma instrução CREATE TABLE em que uma ou mais colunas são definidas como colunas espaciais.

Como observado em "Sobre os tipos de dados espaciais" no Capítulo 4 ("Definição de colunas espaciais, registrando-as como camadas e ativando um geocódigo para mantê-las"), os tipos de dados espaciais formam uma hierarquia, com ST_Geometry como raiz. Quando o texto indica que um valor de um tipo subordinado nessa hierarquia pode ser utilizado como entrada para uma função, o leitor deve entender que, alternativamente, um valor de qualquer tipo subordinado a esse tipo superior pode ser utilizado como uma entrada.

Por exemplo, no Capítulo 14, a seção "Is3d" indica que a função Is3d recebe um valor do tipo de dado ST_Geometry como entrada. O leitor deve compreender que, alternativamente, a entrada em Is3d pode ser um valor de qualquer tipo de dado subordinado a ST_Geometry: ST_Point, ST_Curve, ST_LineString e assim por diante. Para ver um gráfico de todos os tipos de dados subordinados suportados, consulte a Figura 6 no *Manual e Referência do Usuário*.

As considerações a seguir se aplicam às funções espaciais:

- Os exemplos deste capítulo estão qualificados com o nome da biblioteca db2gse. Em vez de qualificar explicitamente cada uma das funções espaciais e tipo com db2gse, você pode definir o caminho da função para que inclua o db2gse.
- Se ST_Union receber como entrada dois pontos que tenham as mesmas coordenadas, irá retornar apenas um ponto. Se esse ponto for passado a ST_IsSimple como uma entrada,, ST_IsSimple retornará um 1 (TRUE; isto é, sim, trata-se de um ponto simples).
- Se uma função espacial receber um valor nulo como parâmetro de entrada, retornará um valor nulo como parâmetro de saída.
- Para poder inserir os dados em uma coluna espacial:
 - Talvez você precise aumentar o parâmetro udf_mem_sz. A definição inicial sugerida é 2048. Se o valor 2048 for inadequado, aumente o parâmetro udf_mem_sz em quantidades de 256.

- É preciso registrá-lo como uma camada. Para obter mais informações sobre o registro de uma coluna espacial como uma camada, consulte “Capítulo 4. Definindo colunas espaciais, registrando-as como camadas e ativando um geocoder para mantê-las” na página 45.

Aninhamento

As funções geométricas de aninhamento, em conjunto, podem apresentar problemas quando a função aninhada interna retornar um tipo de figura geométrica não aceito pela função de chamada externa. Esse problema poderá ser solucionado se o tipo de figura geométrica da função aninhada externa puder ser convertido em um tipo aceitável pela função de chamada externa.

Conversão de ST_Geometry em um subtipo

O tipo de figura geométrica das funções que retornam o supertipo ST_Geometry pode ser convertido num subtipo que use a função Tratar (TREAT). Por exemplo, a função ST_Union retorna valores de ST_Geometry. Quando ST_Union é aninhada na função ST_PointOnSurface, ST_PointOnSurface retorna o seguinte erro:

```
SQL00440N Nenhuma função com o nome "ST_POINTONSURFACE"
que tenha argumentos compatíveis foi encontrada no caminho
da função.      SQLSTATE=42884
```

A função ST_PointOnSurface considera o tipo de figura geométrica ST_Polygon ou ST_MultiPolygon, mas não o ST_Geometry retornado por ST_Union, embora o valor retornado pela função ST_Union seja ST_MultiPolygon. Portanto, nesse caso, é necessário converter o tipo de figura geométrica da função ST_Union em ST_MultiPolygon.

Por exemplo, se a tabela COUNTIES fosse auto-anexada pela união de sua coluna de polígono COUNTY, a função Treat deveria ser aplicada ao resultado da função ST_Union para converter o tipo ST_Geometry em ST_MultiPolygon, antes que a função ST_PointOnSurface pudesse ser aplicada.

```
SELECT ST_Astext(ST_PointOnSurface(
    TREAT ( ST_Union(c1.county, c2.county) AS ST_MultiPolygon)))
FROM   counties AS c1, counties AS c2;
```

Se a função ST_Union retornar um valor de ST_MultiPolygon, a função Treat a converterá num tipo de dado ST_MultiPolygon. Se a função ST_Union não retornar um valor ST_MultiPolygon, a função Tratar retornará um erro de tempo de execução.

Para obter mais informações sobre o tratamento de um subtipo, consulte a *Referência de SQL*.

Conversão de uma coleção em uma figura geométrica básica

Utilize a função `ST_GeometryN` para converter um elemento de uma coleção de figuras geométricas em uma figura geométrica básica requerida pela função de chamada externa.

Por exemplo, o valor retornado pela função `ST_Union` é sempre uma coleção de figuras geométricas retornada como uma `ST_Geometry`. Utilize a função `Treat` para converter o tipo `ST_Geometry` em um subtipo, que pode ser um dos seguintes: `ST_MultiPoint`, `ST_MultiLineString`, `ST_MultiPolygon`, `ST_GeomCollection`, `ST_MultiCurve` ou `ST_MultiSurface`. Aplique a função `ST_GeometryN` à saída da função `Treat` que está convertendo a coleção de figuras geométricas em uma figura geométrica básica.

Por exemplo, para utilizar a função `ST_ExteriorRing` nos resultados da função `ST_Union` do exemplo da seção anterior, utilize primeiro a função `ST_GeometryN` para extrair um elemento de polígono.

```
SELECT ST_AsText(ST_ExteriorRing(ST_GeometryN(
    TREAT ( ST_Union(c1.county, c2.county) AS ST_MultiPolygon ), 1)))
FROM    counties AS c1, counties AS c2;
```

O cálculo é necessário somente ao se passar, na hierarquia, de um supertipo para um subtipo. Para obter mais informações sobre o tratamento de subtipos, consulte a *Referência de SQL*.

AsShape

O AsShape recebe um objeto geométrico e retorna um BLOB.

Sintaxe

```
db2gse.AsShape(g db2gse.ST_Geometry)
```

Tipo de retorno

BLOB(1m)

Exemplos

O seguinte fragmento de código ilustra como a função AsShape converte os polígonos de zona da tabela SENSITIVE_AREAS em polígonos de formato. Tais polígonos de formato são passados para a função de desenho do polígono (draw_polygon) exibição.

```
/* Criar a expressão SQL. */
strcpy(sqlstmt, "select db2gse.AsShape (zone) from SENSITIVE_AREAS
where db2gse.EnvelopesIntersect(zone, db2gse.PolyFromShape(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Preparar as instruções SQL. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* Definir o comprimento de pcbvalue1 do formato. */
pcbvalue1 = blob_len;

/* Ligar o parâmetro do formato */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* Executar a consulta */
rc = SQLExecute (hstmt);

/* Atribuir os resultados da consulta (os polígonos da Zone) a
  fetched_binary variable. */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* Buscar cada polígono dentro da janela de exibição e exibi-lo. */

while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
  draw_polygon(fetched_binary);
```

EnvelopesIntersect

EnvelopesIntersect retorna 1 (VERDADEIRO) se os envelopes de duas figuras geométricas tiverem interseção; caso contrário retornará 0 (FALSO).

Sintaxe

```
db2gse.EnvelopesIntersect(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A função da janela `get_` recupera as coordenadas da janela de exibição da aplicação. O parâmetro da janela é, na verdade, uma estrutura de formato do polígono, que contém uma cadeia de coordenadas que representam o polígono de exibição. A função `PolyFromShape` converte o formato da janela de exibição em um polígono do Spatial Extender que a função `EnvelopesIntersect` utiliza como seu envelope de interseção. Todos os polígonos da zona `SENSITIVE_SAREAS` que fazem interseção com o interior ou com o limite da janela de exibição são retornados. Cada polígono é obtido do conjunto de resultados e passado à função `draw_polygon`.

```
/* Obter as coordenadas da janela de exibição como um formato de polígono.
get_window(&window)
```

```
/* Criar a expressão SQL. A função db2gse.EnvelopesIntersect
   será usada para limitar o conjunto de resultados somente nesses
   polígonos da zona
   que fazem interseção com o envelope da janela de exibição. */
strcpy(sqlstmt, "select db2gse.AsShape(zone) from SENSITIVE_AREAS where
db2gse.EnvelopesIntersect (zone, db2gse.PolyFromShape(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");
```

```
/* Definir blob_len para o comprimento do byte de um polígono de formato
de 5 pontos. */ blob_len = 128;
```

```
/* Preparar as instruções SQL. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);
```

```
/* Definir pcbvalue1 para o formato da janela */
pcbvalue1 = blob_len;
```

```
/* Ligar o parâmetro do formato */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB,
blob_len, 0, window, blob_len, &pcbvalue1);
```

```
/* Executar a consulta */
rc = SQLExecute (hstmt);
```

```
/* Atribuir os resultados da consulta, (os polígonos da Zona) para a
   fetched_binary variable. */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);
```

```
/* Buscar cada polígono dentro da janela de exibição e exibi-lo. */  
while(SQL_SUCCESS == (rc = SQLFetch(hstmt))  
      draw_polygon(fetched_binary);
```

GeometryFromShape

GeometryFromShape recebe um formato e um identificador de sistema de referência espacial para retornar um objeto de figura geométrica.

Sintaxe

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O seguinte fragmento de código C contém funções ODBC incorporadas às funções SQL do Spatial Extender que inserem dados na tabela LOTS.

A tabela LOTS foi criada com duas colunas: a coluna LOT_ID, que identifica cada lote exclusivamente e a coluna polígono do LOT, que contém a figura geométrica de cada lote.

```
CREATE TABLE LOTS ( lot_id integer,
                    lot      db2gse.ST_Polygon);
```

A função GeometryFromShape converte formatos em figuras geométricas do Spatial Extender. A instrução INSERT inteira é copiada em shp_sql. A instrução INSERT contém marcadores do parâmetro para aceitar os dados do LOT_ID e os dados do LOT dinamicamente.

```
/* Crie a instrução insert do SQL para preencher o id do lot_ e as
colunas do lote. Os pontos de interrogação são marcadores de parâmetros que
indicam a id do_lote e os valores do lote que serão recuperados no
tempo de execução.*/ strcpy (shp_sql,"insert into
LOTS (lot_id, lot) values (?,
db2gse.GeometryFromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");
```

```
/* Alocar memória para o tratamento da instrução SQL e associar o
tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);
```

```
/* Ligar o valor inteiro da chave ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Ligar o formato ao segundo parâmetro. */
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

Is3d

Is3d toma um objeto de figura geométrica e retorna 1 (VERDADEIRO) se o objeto possuir coordenadas 3D; caso contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.Is3d(g db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela THREEED_TEST, a qual possui duas colunas: a coluna GID do tipo inteiro e a coluna de figura geométrica G1.

```
CREATE TABLE THREEED_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

As instruções INSERT inserem dois pontos na tabela THREEED_TEST. O primeiro ponto não contém coordenadas Z, ao passo que o segundo sim.

```
INSERT INTO THREEED_TEST  
VALUES(1, db2gse.ST_PointFromText('point (10 10)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO THREEED_TEST  
VALUES (2, db2gse.ST_PointFromText('point z (10.92 10.12 5)',  
db2gse.coordref()..srid(0)))
```

A instrução SELECT a seguir relaciona o conteúdo da coluna GID aos resultados da função Is3d. A função retorna 0 para a primeira linha, a qual não possui coordenadas Z e 1 para a segunda linha, que possui coordenadas Z.

```
SELECT gid, db2gse.Is3d (g1) "Is it 3d?" FROM THREEED_TEST
```

O seguinte conjunto de resultados retorna.

| gid | Is it 3d? |
|-----|-----------|
| 1 | 0 |
| 2 | 1 |

IsMeasured

IsMeasured toma um objeto de figura geométrica e retorna 1 (VERDADEIRO) se o objeto possuir medidas; caso contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.IsMeasured(g db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela MEASURE_TEST, a qual possui duas colunas. A coluna GID identifica exclusivamente as linhas e a coluna G1 armazena as figuras geométricas do ponto.

```
CREATE TABLE MEASURE_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

A seguinte instrução INSERT insere dois registros na tabela MEASURE_TEST. O primeiro registro armazena um ponto que não possui uma medida. O segundo ponto do registro não possui uma medida.

```
INSERT INTO MEASURE_TEST  
VALUES(1, db2gse.ST_PointFromText('point (10 10)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO MEASURE_TEST  
VALUES (2, db2gse.ST_PointFromText('point m (10.92 10.12 5)',  
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostram a coluna GID com os resultados da função IsMeasured. A função IsMeasured retorna 0 para a primeira linha porque o ponto não contém uma medida. Ela retorna 1 para a segunda linha porque o ponto tem medidas.

```
SELECT gid, db2gse.IsMeasured (g1) "Has measures?" FROM MEASURE_TEST  
gid      Contém medidas
```

```
-----  
1          0  
2          1
```

LineFromShape

LineFromShape recebe um formato do tipo ponto e um identificador do sistema de referência espacial e retorna um segmento de reta.

Sintaxe

```
db2gse.Line FromShape(ShapeLineString Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_LineString
```

Exemplos

O seguinte fragmento de código preenche a tabela SEWERLINES com id, classe de tamanho e geometria exclusivos de cada linha sewer.

A instrução CREATE TABLE cria a tabela SEWERLINES, a qual possui três colunas. A primeira coluna, SEWER_ID, identifica exclusivamente cada linha do sewer. A segunda coluna, CLASS, do tipo inteiro identifica o tipo de linha sewer, que geralmente está associada a uma capacidade da linha. A terceira coluna, SEWER, do tipo cadeia de linhas armazena a figura geométrica da linha sewer.

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer,
                          sewer db2gse.ST_LineString);

/* Criar a instrução insert do SQL para preencher a id do sewer, o tamanho e
   a cadeia de linha sewer. Os pontos de interrogação são marcadores de
   parâmetros que
   indicam a id do sewer, a classe e os valores de figura geométrica do sewer
   que serão
   recuperados no tempo de execução. */
strcpy (shp_sql,"insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.Line FromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Ligar o valor inteiro da chave ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* Ligar o valor da classe de inteiro ao segundo parâmetro. */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);;

/* Ligar o formato ao terceiro parâmetro. */
```

```
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, sewer_shape, blob_len, &pcbvalue3);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);
```

LocateAlong

LocateAlong toma um objeto de figura geométrica e uma medida para retornar como um multiponto do conjunto de pontos encontrados na medida.

Se LocateAlong receber um ponto múltiplo e uma medida como entradas e se o ponto múltiplo não incluir essa medida, o LocateAlong retornará POINT EMPTY.

Sintaxe

```
db2gse.LocateAlong(g db2gse.ST_Geometry, [measure] Double)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela LOCATEALONG_TEST. LOCATEALONG_TEST apresenta duas colunas: a coluna GID, que identifica exclusivamente cada linha e a coluna da figura geométrica G1, que armazena a figura geométrica de amostra.

```
CREATE TABLE LOCATEALONG_TEST (gid integer, g1 db2gse.ST_Geometry)
```

A seguinte instrução INSERT insere duas linhas. A primeira é uma cadeia de linhas múltiplas; a segunda é de multipontos.

```
INSERT INTO db2gse.LOCATEALONG_TEST VALUES(
1, db2gse.ST_MLineFromText('multilinestring m ((10.29 19.23 5,23.82 20.29 6,
30.19 18.47 7,
45.98 20.74 8), (23.82 20.29 6,30.98 23.98 7,42.92 25.98 8))',
db2gse.coordref().srid(0))
```

```
INSERT INTO db2gse.LocateAlong_TEST VALUES(
2, db2gse.ST_MPointFromText('multipoint m (10.29 19.23 5,23.82 20.29 6,
30.19 18.47 7,45.98 20.74 8,23.82 20.29 6,30.98 23.98 7,42.92 25.98)',
db2gse.coordref().srid(0))
```

Na seguinte instrução SELECT e no conjunto de resultados correspondente, a função LocateAlong está direcionada para localizar pontos cuja medida seja 6,5. A primeira linha retorna um multiponto que contém dois pontos. No entanto, a segunda linha retornou um ponto vazio. Para recursos lineares (figura geométrica com dimensão maior que 0), LocateAlong pode interpolar o ponto, no entanto, para multipontos, a medida de destino deve corresponder exatamente.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.LocateAlong (g1,6.5)) AS
varchar(96))
"Geometry"
FROM LOCATEALONG_TEST
```

```

GID          Geometry
-----
          1 MULTIPOINT M ( 27.01000000 19.38000000 6.50000000, 27.40000000
22.14000000 6.50000000)
          2 POINT EMPTY

2 record(s) selected.

```

Na seguinte instrução SELECT e no conjunto de resultados correspondente, a função LocateAlong retorna multipontos para ambas as linhas. A medida de destino de 7 corresponde às medidas tanto nos dados de origem da cadeia de linhas múltiplas como de multipontos.

```

SELECT gid,CAST(db2gse.ST_AsText(db2gse.LocateAlong (g1,7)) AS varchar(96))
"Geometry"
FROM LOCATEALONG_TEST

```

```

GID          Geometry
-----
          1 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000
23.98000000 7.00000000)
          2 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000
23.98000000 7.00000000)

2 record(s) selected.

```

LocateBetween

O `LocateBetween` pega um objeto de figura geométrica e duas localizações de medida e retorna uma figura geométrica que representa o conjunto de caminhos desconectados entre as duas localizações de medida.

Sintaxe

```
db2gse.LocateBetween(g db2gse.ST_Geometry, [measure] Double, [measure]5 Double)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A seguinte instrução `CREATE TABLE` cria a tabela `LOCATEBETWEEN_TEST`. `LOCATEBETWEEN_TEST` apresenta duas colunas: a coluna `GID`, que identifica exclusivamente cada linha e a coluna `G1` de cadeia de multilinha, que armazena a figura geométrica de amostra.

```
CREATE TABLE LOCATEBETWEEN_TEST (gid integer, g1 db2gse.ST_Geometry)
```

A seguinte instrução `INSERT` insere duas linhas na tabela `LOCATEBETWEEN_TEST`. A primeira linha é uma cadeia de linhas múltiplas e a segunda é um multiponto.

```
INSERT INTO db2gse.LOCATEBETWEEN_TEST
VALUES(1,db2gse.ST_MLineFromText('multilinestring m ((10.29 19.23 5,
                                     23.82 20.29 6, 30.19 18.47 7,45.98 20.74 8),
                                     (23.82 20.29 6,30.98 23.98 7,
42.92 25.98 8))'),
db2gse.coordref(..srid(0)))
```

```
INSERT INTO db2gse.LOCATEBETWEEN_TEST
VALUES(2, db2gse.ST_MPointFromText('multipoint m (10.29 19.23 5,23.82 20.29 6,
30.19 18.47 7,45.98 20.74 8,23.82 20.29 6,
30.98 23.98 7,42.92 25.98 8)'),
db2gse.coordref(..srid(0)))
```

A seguinte instrução `SELECT` e o conjunto de resultados correspondente mostram a função `LocateBetween` localizando medidas entre as 6,5 e 7,5 inclusive. A primeira linha retorna uma cadeia de linhas múltiplas com várias cadeias de linhas. A segunda linha retorna um multiponto porque os dados de origem eram multipontos. Quando os dados de origem têm dimensão de 0 (ponto ou multiponto), é solicitada uma correspondência exata.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.LocateBetween (g1,6.5,7.5))
                AS varchar(96)) "Geometry"
FROM LOCATEBETWEEN_TEST
```

```
GID          Geometry
-----
1 MULTILINESTRING M ( 27.01000000 19.38000000 6.50000000, 31.19000000
```

```
18.47000000 7.00000000,38.09000000 19.61000000 7.50000000),(27.40000000 22.1400  
0000 6.50000000, 30.98000000 23.98000000 7.00000000,36.95000000 24.98000000 7.5  
00000000)  
2 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000 23.9  
8000000 7.00000000)
```

2 record(s) selected.

M

M toma um ponto e retorna sua medida.

Sintaxe

```
db2gse.M(p db2gse.ST_Point)
```

Tipo de retorno

Duplo

Exemplos

A seguinte instrução CREATE TABLE cria a tabela M_TEST. M_TEST apresenta duas colunas: a coluna de inteiros GID, que identifica exclusivamente a linha e a coluna de pontos PT1, que armazena a figura geométrica de amostra.

```
CREATE TABLE M_TEST (gid integer, pt1 db2gse.ST_Point)
```

As seguintes instruções INSERT inserem uma linha que contém um ponto com medidas e uma linha que contém um ponto sem medidas.

```
INSERT INTO db2gse.M_TEST
VALUES(1, db2gse.ST_PointFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
INSERT INTO db2gse.M_TEST
VALUES(2, db2gse.ST_PointFromText('point zm(10.02 20.01 5.0 7.0)',
db2gse.coordref()..srid(0)))
```

Na seguinte instrução SELECT e no conjunto de resultados correspondente, a função M lista os valores de medida dos pontos. Como o primeiro ponto não tem medidas, a função M retorna um NULO.

```
SELECT gid, db2gse.M (pt1) "The measure" FROM M_TEST
```

```
GID          The measure
-----
1              -
2    +7.00000000000000E+000
```

2 record(s) selected.

MLine FromShape

MLine FromShape recebe um formato do tipo segmento de reta e um identificador do sistema de referências espaciais e retorna um segmento de reta múltiplo.

Sintaxe

```
db2gse.MLineFromShape(ShapeMultiLineString Blob(1M), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiLineString
```

Exemplos

O seguinte fragmento de código preenche a tabela WATERWAYS com uma id exclusiva, um nome e uma cadeia de linhas múltiplas de água.

A tabela WATERWAYS é criada com as colunas ID e NAME que identificam cada sistema de córregos e rios armazenados na tabela. A coluna ÁGUA é uma cadeia de linhas múltiplas pois os sistemas de rios e córregos são geralmente uma agregação de várias cadeias de linhas.

```
CREATE TABLE WATERWAYS (id      integer,  
                          name    varchar(128),  
                          water    db2gse.ST_MultiLineString);  
  
/* Criar a instrução insert do SQL para preencher a id, nome e  
   a cadeia de linhas múltiplas. Os pontos de interrogação são  
   marcadores de parâmetros que  
   indicam a id, o nome e os valores da água que serão recuperados no  
   tempo de execução.*/  
strcpy (shp_sql,"insert into WATERWAYS (id,name,water)  
values (?,?, db2gse.MLineFromShape (cast(? as blob(1m)),  
db2gse.coordref()..srid(0)))");  
  
/* Alocar memória para o tratamento da instrução SQL e associar o  
   tratamento da instrução ao tratamento da conexão. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Preparar as instruções SQL para execução. */  
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);  
  
/* Ligar o valor inteiro do id ao primeiro parâmetro.*/  
  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);  
/* Ligar o valor do nome varchar ao segundo parâmetro. */  
  
pcbvalue2 = name_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,  
SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);
```

```
/* Ligar o formato ao terceiro parâmetro. */  
pcbvalue3 = blob_len;  
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);  
  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

MPointFromShape

MPointFromShape recebe um formato do tipo ponto múltiplo e um identificador do sistema de referências espaciais e retorna um ponto múltiplo.

Sintaxe

```
db2gse.MPointFromShape(ShapeMultiPoint (1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPoint
```

Exemplos

Este fragmento de código preenche a tabela SPECIES_SITINGS de um biólogo.

A tabela SPECIES_SITINGS foi criada com três colunas. As colunas espécies e gêneros identificam exclusivamente cada linha, os pontos que o multiponto de locais armazena as localizações dos locais das espécies.

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                               genus varchar(32),
                               sitings db2gse.ST_MultiPoint);

/* Criar a instrução insert do SQL para preencher as espécies, gêneros e
   sítios. Os pontos de interrogação são marcadores de parâmetros que
   indicam o
   nome e os valores da água que serão recuperados no tempo de execução. */
strcpy (shp_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.MPointFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Ligar o valor de espécies varchar ao primeiro parâmetro. */
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, species_len, 0, species, species_len, &pcbvalue1);

/* Ligar valor do gênero varchar ao segundo parâmetro. */
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, genus_len, 0, name, genus_len, &pcbvalue2);

/* Ligar o formato ao terceiro parâmetro. */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, sitings_len, 0, sitings_shape, sitings_len, &pcbvalue3);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);
```

MPolyFromShape

MPolyFromShape recebe um formato do tipo polígono múltiplo e um identificador do sistema de referências espaciais para retornar um polígono múltiplo.

Sintaxe

```
db2gse.MPolyFromShape(ShapeMultiPolygon Blob(1m), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPolygon
```

Exemplos

Este fragmento de código preenche a tabela LOTS.

A tabela LOTS armazena a id do lote que identifica exclusivamente cada lote e o multipolígono do lote que contém a figura geométrica de linha do lote.

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );

/* Criar a instrução insert do SQL para preencher a lot_id e o lote. Os
   pontos de interrogação são marcadores de parâmetro que indicam a lot_id e
   os valores do lote que serão recuperados no tempo de execução. */
strcpy (shp_sql,"insert into LOTS (lot_id,lot)
values (?, db2gse.MPolyFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Ligar o valor inteiro do id do lote ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* Ligar o formato do lote ao segundo parâmetro. */
pcbvalue2 = lot_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, lot_len, 0, lot_shape, lot_len, &pcbvalue2);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);
```

PointFromShape

PointFromShape recebe um formato do tipo ponto e um identificador do sistema de referências espaciais para retornar um ponto.

Sintaxe

```
db2gse.PointFromShape(db2gse.ShapePoint blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

O fragmento do programa preenche a tabela HAZARDOUS_SITES.

Os locais arriscados estão armazenados na tabela HAZARDOUS_SITES criada com a instrução CREATE TABLE que segue. O nome da localização, definido como um ponto, armazena uma localização que é o centro geográfico de cada ponto arriscado.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);

/* Criar a instrução insert do SQL para preencher a id do local, nome e
   localização. Os pontos de interrogação são marcadores de parâmetros
   que indicam a
   id do local, nome e os valores de localização que serão recuperados no
   tempo de execução. */
strcpy (shp_sql,"insert into HAZARDOUS_SITES (site_id, name, location)
values (?,?, db2gse.PointFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Ligar o valor inteiro do id do local ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* Ligar o valor varchar do nome ao segundo parâmetro. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* Ligar o formato de localização ao terceiro parâmetro. */
pcbvalue3 = location_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, location_len, 0, location_shape, location_len, &pcbvalue3);  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

PolyFromShape

PolyFromShape recebe um formato do tipo polígono e um identificador do sistema de referências espaciais para retornar um polígono.

Sintaxe

db2gse.PolyFromShape (ShapePolygon Blob(1M), SRID db2gse.coordref)

Tipo de retorno

db2gse.ST_Polygon

Exemplos

O fragmento do programa preenche a tabela SENSITIVE_AREAS. Os pontos de interrogação representam marcadores de parâmetros para o id, nome, tamanho, tipo e valores da zona que serão recuperados no tempo de execução.

A tabela SENSITIVE_AREAS contém várias colunas que descrevem as instituições ameaçadas além da coluna zona que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

```
/* Criar a instrução insert do SQL para preencher a id, nome, tamanho, tipo e
   zona. Os pontos de interrogação são marcadores de parâmetro que
   indicam o id, nome, tamanho, tipo e valores da zona que serão
   recuperados no tempo de execução. */
```

```
strcpy (shp_sql, "insert into SENSITIVE AREAS (id, name, size, type, zone)
values (?, ?, ?, ?, db2gse.PolyFromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");
```

```
/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);
```

```
/* Ligar o valor inteiro do id ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);
/* Ligar o valor varchar do nome ao segundo parâmetro. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);
```

```
/* Ligar a dimensão flutuante ao terceiro parâmetro. */
pcbvalue3 = 0;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
```



```

        SQL_REAL, 0, 0, &size, 0, &pcbvalue3);

/* Ligar o tipo varchar ao quarto parâmetro. */
pcbvalue4 = type_len;
rc = SQLBindParameter (hstmt, 4, SQL_PARAM_INPUT, SQL_C_CHAR,
        SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);

/* Ligar o polígono de zona ao quinto parâmetro. */
pcbvalue5 = zone_len;
rc = SQLBindParameter (hstmt, 5, SQL_PARAM_INPUT, SQL_C_BINARY,
        SQL_BLOB, zone_len, 0, zone_shp, zone_len, &pcbvalue5);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);

```

ShapeToSQL

ShapeToSQL constrói um valor de `db2gse.ST_Geometry`, fornecendo sua representação de formato. O valor 0 de SRID é usado automaticamente.

Sintaxe

```
db2gse.ShapeToSQL(ShapeGeometry blob(1M))
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O seguinte fragmento de código C contém funções ODBC incorporadas às funções SQL do Spatial Extender que inserem dados na tabela LOTS. A tabela LOTS foi criada com duas colunas: a coluna do ID do `lot_`, que identifica cada lote com exclusividade, e a coluna de polígonos múltiplos do lote, que contém a figura geométrica de cada lote.

```
CREATE TABLE lots (lot_id integer,  
                  lot      db2gse.ST_MultiPolygon);
```

A função `ShapeToSQL` converte formatos em figura geométrica do Spatial Extender. A instrução `INSERT` inteira é copiada em `shp_sql`. A instrução `INSERT` contém marcadores do parâmetro para aceitar os dados da id do lote e os dados do LOT dinamicamente.

```
/* Criar a instrução insert do SQL para preencher o id do lote e o  
   multipolígono do lote. Os pontos de interrogação são marcadores  
   do parâmetro que  
   indicam a id do lote e os valores do lote  
   que serão recuperados no tempo de execução.*/
```

```
strcpy (shp_sql,"insert into lots (lot_id, lot) values(?,  
db2gse.ShapeToSQL(cast(? as blob(1m))))");
```

```
/* Alocar memória para o tratamento da instrução SQL e associar o  
   tratamento da instrução ao tratamento da conexão. */
```

```
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Preparar as instruções SQL para execução. */
```

```
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);
```

```
/* Ligar o valor inteiro da chave ao primeiro parâmetro.*/
```

```
pcbvalue1 = 0;
```

```
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Ligar o formato ao segundo parâmetro. */
```

```
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);

/* Executar a instrução insert . */

rc = SQLExecute (hstmt);
```

ST_Area

ST_A área toma um polígono ou multipolígonos e retorna sua área.

Sintaxe

```
db2gse.ST_Area(s db2gse.ST_Surface)
db2gse.ST_Surface
db2gse.ST_Polygondb2gse.ST_MultiSurface
db2gse.ST_MultiPolygon
```

Tipo de retorno

Duplo

Exemplos

O engenheiro da cidade precisa de uma lista de áreas de construção. Para obtê-la, um técnico GIS seleciona a ID da construção e a área de cada base do prédio.

As bases da construção são armazenadas na tabela BUILDINGFOOTPRINTS que foi criada com a seguinte instrução CREATE TABLE:

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

Para atender ao pedido do engenheiro da cidade, o técnico utiliza a seguinte instrução SELECT para selecionar a chave exclusiva, a id da construção e a área de cada base de construção a partir da tabela BUILDINGFOOTPRINTS:

```
SELECT building_id, db2gse.ST_Area (footprint) "Area"
FROM BUILDINGFOOTPRINTS;
```

A instrução SELECT retorna o seguinte conjunto de resultados:

| ID_da construção | Área |
|------------------|-------------------------|
| 506 | +1.407680000000000E+003 |
| 1208 | +2.557590000000000E+003 |
| 543 | +1.807860000000000E+003 |
| 178 | +2.086710000000000E+003 |
| . | . |
| . | . |
| . | . |

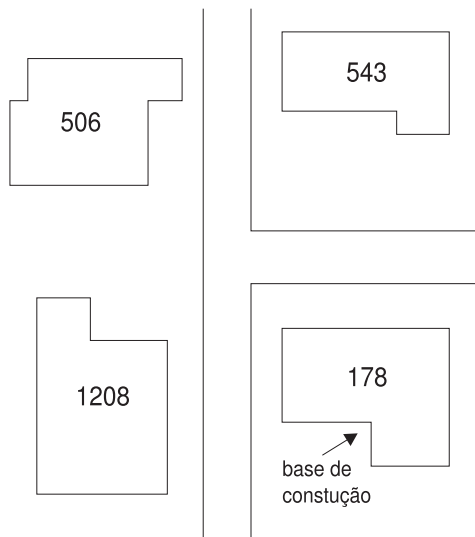


Figura 28. Utilizando a área para encontrar a base da construção. Quatro das bases de construção rotuladas com seus números de ID de construção são exibidas junto à rua adjacente.

Nota: Como indicado anteriormente, `ST_Area` pode receber um polígono ou um polígono múltiplo como entrada. Assim, o tipo de dados do parâmetro de entrada pode ser:

- Ou os tipos de dados utilizados para os polígonos (`db2gse.ST_Surface` ou `db2gse.ST_Polygon`)
- Ou os tipos de dados utilizados para os polígonos múltiplos (`db2gse.ST_MultiSurface` ou `db2gse.ST_MultiPolygon`)

ST_AsBinary

ST_AsBinary toma um objeto de figura geométrica e retorna sua representação de modo binário reconhecida. ST_AsBinary não pode receber uma figura geométrica vazia como entrada (SQLSTATE 38827).

Sintaxe

```
db2gse.ST_AsBinary(g db2gse.ST_Geometry)
```

Tipo de retorno

BLOB(1m)

Exemplos

O seguinte fragmento de código ilustra como a função ST_AsBinary converte os multipolígonos da base da tabela BUILDINGFOOTPRINTS em multipolígonos WKB. Tais multipolígonos são passados para a função de desenho do polígono (draw_polygon) para exibição.

```
/* Criar a expressão SQL. */
strcpy(sqlstmt, "select db2gse.ST_AsBinary (footprint) from BUILDINGFOOTPRINTS
where db2gse.EnvelopesIntersect(footprint, db2gse.ST_PolyFromWKB
(cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Preparar as instruções SQL. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* Definir o comprimento de pcbvalue1 do formato. */
pcbvalue1 = blob_len;

/* Ligar o parâmetro do formato */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* Executar a consulta */
rc = SQLExecute (hstmt);

/* Atribuir os resultados da consulta (os polígonos da Zone) a
  fetched_binary variable.
  */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* Buscar cada polígono dentro da janela de exibição e exibi-lo. */
while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
  draw_polygon(fetched_binary);
```

ST_AsText

O `db2gse.ST_AsText` pega um objeto de figura geométrica e retorna sua representação de texto reconhecida.

Sintaxe

```
db2gse.ST_AsText(g db2gse.ST_Geometry)
```

Tipo de retorno

```
Varchar(4000)
```

Exemplos

No cenário a seguir, a função `db2gse.ST_AsText` converte o ponto da localização `HAZARDOUS_SITES` na descrição de seu texto:

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                                name      varchar(40),
                                location  db2gse.ST_Point);

INSERT INTO HAZARDOUS_SITES
VALUES (102,
        'W. H. Kleenare Chemical Repository',
        db2gse.ST_PointFromText('point (1020.12 324.02)',
        db2gse.coordref().srid));

SELECT site_id, name, cast(db2gse.ST_AsText(location) as varchar(40))
       "Location"
FROM HAZARDOUS_SITES;
```

A instrução `SELECT` retorna o seguinte conjunto de resultados:

| ID do_LOCAL | Nome | Localização |
|-------------|------------------------------------|------------------------------------|
| 102 | W. H. Kleenare Chemical Repository | POINT (1020.00000000 324.00000000) |

ST_Boundary

ST_Boundary toma um objeto de figura geométrica e retorna seu limite combinado como um objeto de figura geométrica. Os pontos e os pontos múltiplos resultam sempre em um limite, que é uma figura geométrica vazia com dimensão 0 (não uma dimensão 1).

Sintaxe

```
db2gse.ST_Boundary(g db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

No seguinte fragmento de código, é criada uma tabela chamada BOUNDARY_TEST. BOUNDARY_TEST tem duas colunas: GEOTYPE, que está definida como varchar e G1, definida como figura geométrica de superclasse. As instruções INSERT que se seguem inserem cada uma das figuras geométricas de subclasse. A função ST_Boundary recupera o limite de cada subclasse que está armazenada na coluna de figura geométrica G1. Observe que a dimensão da figura geométrica resultante é sempre uma menor que a figura geométrica de entrada. Os pontos e multipontos sempre resultam num limite que é uma figura geométrica vazia, dimensão 1. As cadeias de linhas e cadeias de linhas múltiplas retornam um limite multiponto, dimensão 0. Um polígono ou multipolígono sempre retorna um limite de cadeia de linhas múltiplas, dimensão 1.

```
CREATE TABLE BOUNDARY_TEST (GEOTYPE varchar(20), G1 db2gse.ST_Geometry)
```

```
INSERT INTO BOUNDARY_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
      db2gse.coordref()..srid(0)))
```

```
INSERT INTO BOUNDARY_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO BOUNDARY_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94, 10.02 20.01))',
      db2gse.coordref()..srid(0)))
```

```
INSERT INTO BOUNDARY_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```



```

INSERT INTO BOUNDARY_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64), (9.55 23.75,15.36 30.11))',
      db2gse.coordref()..srid(0)))

INSERT INTO BOUNDARY_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
      ((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))',
      db2gse.coordref()..srid(0)))

SELECT GEOTYPE,
       CAST(db2gse.ST_AsText(db2gse.ST_Boundary (G1)) as varchar(280))
       "The boundary"
FROM BOUNDARY_TEST

```

| GEOTYPE | The boundary |
|----------------------------|--|
| Point | POINT EMPTY |
| Linestring 25.64000000) | MULTIPOINT (10.02000000 20.01000000, 11.92000000 |
| Polygon | MULTILINESTRING ((10.02000000 20.01000000, 19.15000000 |
| | 33.94000000, 25.02000000 34.15000000, 11.92000000 35.64000000, 10.02000000 |
| | 20.01000000)) |
| Multipoint | POINT EMPTY |
| Multilinestring | MULTIPOINT (9.55000000 23.75000000, 10.02000000 |
| | 20.01000000, 11.92000000 25.64000000, 15.36000000 30.11000000) |
| Multipolygon | MULTILINESTRING ((51.71000000 21.73000000, 73.36000000 |
| | 27.04000000, 71.52000000 32.87000000, 52.43000000 31.90000000, 51.71000000 |
| | 21.73000000),(10.02000000 20.01000000, 19.15000000 33.94000000, 25.02000000 |
| | 34.15000000, 11.92000000 35.64000000, 10.02000000 20.01000000)) |

6 record(s) selected.

ST_Buffer

ST_Buffer toma um objeto de figura geométrica e distância e retorna o objeto de figura geométrica que está ao redor do objeto de origem.

Sintaxe

```
db2gse.ST_Buffer(g db2gse.ST_Geometry , [measure] Double)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O Supervisor do Município precisa de uma lista de locais arriscados num raio de 8 km que abranja áreas sensíveis, como escolas, hospitais e casas de repouso. As áreas sensíveis são armazenadas na tabela SENSITIVE_AREAS que é criada com a seguinte instrução CREATE TABLE. A coluna ZONE é definida como um polígono, que está armazenado como o contorno de cada linha sensível.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

As áreas de risco são armazenadas na tabela HAZARDOUS_SITES que foi criada com a seguinte instrução CREATE TABLE. A coluna LOCATION, definida como um ponto, armazena uma localização que é o centro geográfico de cada local arriscado.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);

INSERT INTO HAZARDOUS_SITES
VALUES (102, 'Allied Chemicals',
       db2gse.ST_PointFromText('Point(157000 475000)', coordref()..srid(0)))
```

As tabelas SENSITIVE_AREAS e HAZARDOUS_SITES são unidas pela função db2gse.ST_Overlaps. A função retorna 1 (VERDADEIRO) para todas as linhas SENSITIVE_AREAS cujos polígonos da zona sobrepõem o raio de buffer de 8 km do ponto de localização do HAZARDOUS_SITES.

```
SELECT sa.name "Sensitive Areas", hs.name "Hazardous Sites"
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Overlaps(sa.zone, db2gse.ST_Buffer (hs.location,(5 * 5280)))
=1;
```

(5 * 5280 representa cinco milhas (oito quilômetros). Isso decorre do fato de haver 5280 pés em uma milha e de um pé ser a unidade linear do sistema de coordenadas ao qual as coordenadas especificadas na instrução VALUES pertencem).

Na Figura 29, algumas áreas sensíveis neste distrito administrativo estão dentro do buffer de 8 km das localizações de pontos arriscados. Os dois buffers de 8 km fazem interseção com o hospital e um deles faz interseção com a escola. Contudo, a casa de repouso está localizada com segurança fora dos dois raios.

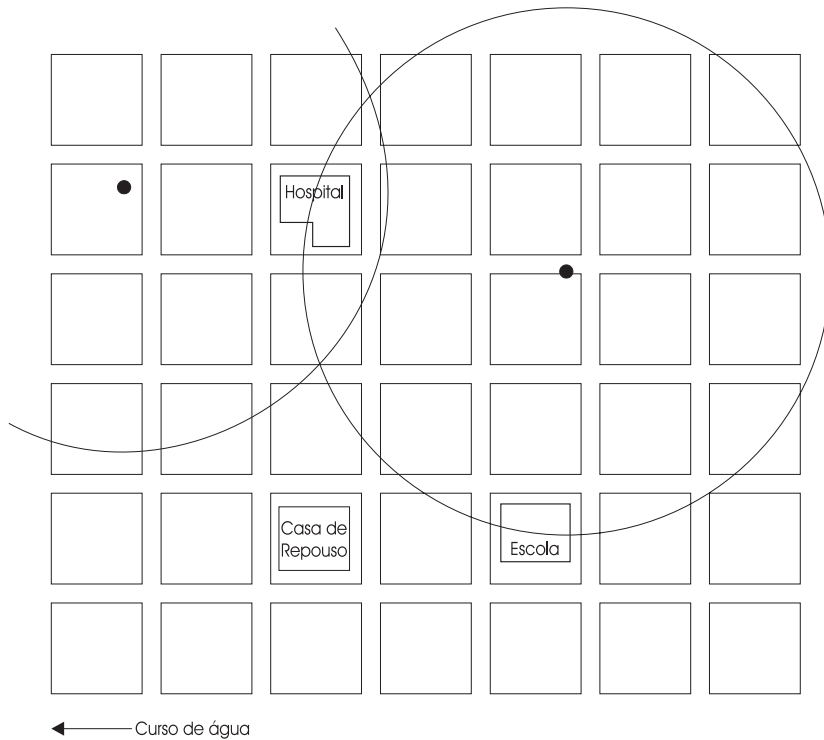


Figura 29. Um buffer com um raio de 8 km é aplicado a um ponto

ST_Centroid

ST_Centroid toma um polígono ou um multipolígono e retorna seu centro geométrico como um ponto.

Sintaxe

```
db2gse.ST_Centroid(s db2gse.ST_Surface)
db2gse.ST_Centroid(ms db2gse.ST_MultiSurface)
```

Tipo de retorno

Na superfície: db2gse.ST_Point

Exemplos

O técnico em GIS da cidade deseja exibir os multipolígonos das bases da construção como pontos únicos num gráfico de densidade de construção.

As bases da construção são armazenadas na tabela BUILDINGFOOTPRINTS que foi criada com a seguinte instrução CREATE TABLE.

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

A função ST_Centroid retorna a id de centro de cada multipolígono da base da construção. A função AsShape converte o ponto do ID do centro em um formato, a representação externa que é reconhecida pelo aplicativo.

```
SELECT building_id,
       CAST(db2gse.AsShape(db2gse.ST_Centroid (footprint)) as blob(1m))
  "Centroid"
FROM BUILDINGFOOTPRINTS;
```

ST_Contains

ST_Contains toma dois objetos de figura geométrica e retorna 1 (VERDADEIRO) se o primeiro objeto incluir completamente o segundo; caso contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Contains(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

No exemplo abaixo são criadas duas tabelas. Uma tabela contém as bases de construção da cidade, ao passo que a outra contém seus lotes. O engenheiro da cidade deseja verificar se todas as bases de construção estão inteiramente dentro de seus lotes.

Nas duas tabelas o tipo de dados de multipolígono armazena a figura geométrica das bases e lotes de construção. O projetista de bancos de dados selecionou multipolígonos para ambos os recursos. O projetista imaginou que os lotes podem ser desmembrados por recursos naturais, como um rio e que as bases de construção podem ser geralmente constituídas de vários edifícios.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                   lot_id         integer,
                                   footprint      db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (  lot_id integer, lot db2gse.ST_MultiPolygon );
```

O engenheiro da cidade primeiro seleciona as construções que não estão inteiramente dentro de um lote.

```
SELECT building_id
   FROM BUILDINGFOOTPRINTS, LOTS
WHERE db2gse.ST_Contains(lot, footprint) = 0;
```

O engenheiro da cidade imagina que a primeira consulta retornará uma lista de todas as IDs de construção que possuem bases fora de um polígono do lote. Mas o engenheiro da cidade também sabe que estas informações não indicarão se as outras construções possuem a ID de lote correta atribuída a eles. Esta segunda consulta executa uma verificação de integridade dos dados na id do lote da tabela BUILDINGFOOTPRINTS.

```
SELECT bf.building_id "building id", bf.lot_id "buildings lot_id",
       LOTS.lot_id "LOTS lot_id"
   FROM BUILDINGFOOTPRINTS bf, LOTS
WHERE db2gse.ST_Contains(lot, footprint) = 1 AND LOTS.lot_id <> bf.lot_id;
```

Na Figura 30 na página 232, as bases de construção rotuladas com suas IDs de construção estão dentro de seus lotes. As linhas do lote estão ilustradas com

linhas pontilhadas. Embora não ilustradas, estas linhas ampliam a linha central da rua e incluem totalmente os lotes e as bases de construção dentro delas.

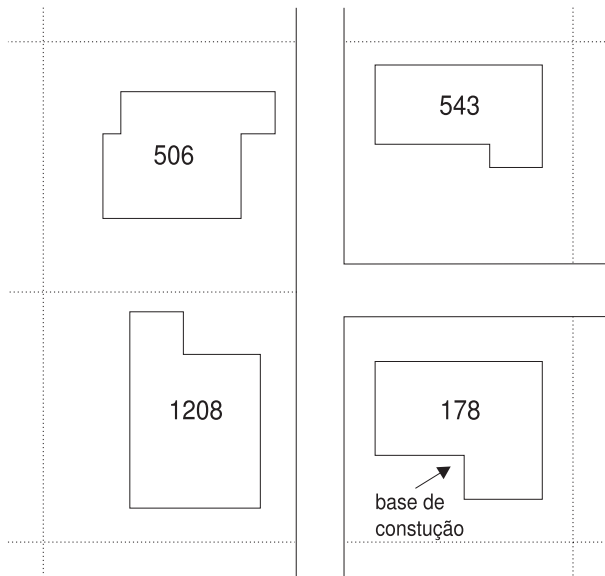


Figura 30. Utilizando `ST_Contains` para garantir que todas as construções fiquem dentro de seus lotes

ST_ConvexHull

ST_ConvexHull toma um objeto de figura geométrica e retorna a carcaça convexa.

Sintaxe

```
db2gse.ST_ConvexHull(g db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O exemplo cria a tabela CONVEXHULL_TEST que tem duas colunas: GEOTYPE e G1. A coluna GEOTYPE, varchar(20), armazenará o nome da subclasse de figura geométrica que está armazenada em G1, a qual está definida como figura geométrica.

```
CREATE TABLE CONVEXHULL_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Cada instrução INSERT insere uma figura geométrica de cada tipo de subclasse na tabela CONVEXHULL_TEST.

```
INSERT INTO CONVEXHULL_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
INSERT INTO CONVEXHULL_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))

INSERT INTO CONVEXHULL_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94,10.02 20.01))',
db2gse.coordref()..srid(0))

INSERT INTO CONVEXHULL_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))

INSERT INTO CONVEXHULL_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64),(9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0))

INSERT INTO CONVEXHULL_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01))),
```

```
((51.71 21.73,73.36 27.04,71.52 32.87,  
52.43 31.90,51.71 21.73)))',
```

```
db2gse.coordref(..srid(0)))
```

A seguinte instrução SELECT relaciona o nome da subclasse armazenado na coluna GEOTYPE e a carcaça convexa. A carcaça convexa gerada pela função ST_ConvexHull é convertida em texto pela função ST_AsText. É então feita a conversão em um varchar(256) porque a saída padrão de ST_AsText é varchar(4000).

```
SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_ConvexHull(G1)))  
as varchar(256) "The convexhull"  
FROM CONVEXHULL_TEST
```

ST_CoordDim

ST_CoordDim retorna as dimensões coordenadas do valor ST_Geometry. Para obter uma explicação sobre as dimensões coordenadas, consulte o “Pontos” na página 165.

Sintaxe

```
db2gse.ST_CoordDim(g1 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A tabela coorddim_test é criada com as colunas geotype e g1. A coluna geotype armazena o nome da subclasse de figura geométrica armazenada na coluna de figura geométrica g1.

```
CREATE TABLE coorddim_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

As instruções INSERT inserem uma subclasse de amostra na tabela coorddim_test.

```
INSERT INTO coorddim_test VALUES(
  'Point', db2gse.ST_PointFromText('point (10.02 20.01)',
  db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
  'Linestring',
  db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,11.92 25.64)',
  db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
  'Polygon', db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,
  25.02 34.15, 19.15 33.94,10.02 20.01))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
  'Multipoint', db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
  11.92 25.64)', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
  'Multilinestring', db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,
  10.32 23.98,11.92 25.64),(9.55 23.75,15.36 30.11))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
  'Multipolygon',
  MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,25.02 34.15,
  19.15 33.94,10.02 20.01)),((51.71 21.73,73.36 27.04,71.52 32.87,
  52.43 31.90,51.71 21.73)))', db2gse.coordref()..srid(0))
)
```

A instrução `SELECT` relaciona o nome da subclasse armazenada na coluna `geotype` à dimensão coordenada desse `geotype`.

```
SELECT geotype, db2gse.ST_coordDim(g1)' coordinate_dimension'  
FROM coorddim_test
```

| GEOTYPE | coordinate_dimension |
|--------------------|----------------------|
| ST_Point | 2 |
| ST_Linestring | 2 |
| ST_Polygon | 2 |
| ST_Multipoint | 2 |
| ST_Multilinestring | 2 |
| ST_Multipolygon | 2 |

6 record(s) selected.

ST_Crosses

ST_Crosses toma dois objetos de figura geométrica e retorna 1 (VERDADEIRO) se sua interseção resultar num objeto de figura geométrica cuja dimensão seja menor que a dimensão máxima dos objetos de origem. O objeto de interseção contém pontos que são interiores para as duas figuras geométricas de origem e não é igual a nenhum dos objetos de origem. Do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Crosses(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A prefeitura do município está considerando uma nova lei que declara que todas as estações de armazenamento de lixo tóxico dentro do município não poderão estar dentro de 8 km de qualquer curso de água. O gerente GIS do município possui uma representação exata de rios e córregos, que está armazenada como cadeias de linhas múltiplas na tabela WATERWAYS. Mas, o gerente GIS apresenta somente uma única localização de ponto para cada estação de armazenamento de lixo tóxico.

```
CREATE TABLE WATERWAYS (id      integer,
                          name    varchar(128),
                          water    db2gse.ST_MultiLineString);
```

```
CREATE TABLE HAZARDOUS_SITES ( site_id integer,
                                name    varchar(128),
                                location db2gse.ST_Point);
```

Para determinar se o supervisor do município precisa ser alertado das estações que violam o regulamento proposto, o gerente GIS deve ter buffer dos locais arriscados e verificar se algum rio ou córrego cruza o polígono do buffer. O predicado ST_Crosses compara os HAZARDOUS_SITES com buffer aos WATERWAYS. Assim, retornam somente esses registros nos quais o curso da água cruza o raio regulamentado proposto pelo município.

```
SELECT ww.name "River or stream", hs.name "Hazardous site"
FROM WATERWAYS ww, HAZARDOUS_SITES hs
WHERE db2gse.ST_Crosses(db2gse.ST_Buffer(hs.location,(5 * 5280)),ww.water)
=1;
```

Na Figura 31 na página 238, o buffer de 8 km dos locais de lixo tóxico cruza a rede de córregos que passa pelo distrito administrativo do município. A rede de córregos foi definida como uma cadeia de linhas múltiplas. Portanto, o conjunto de resultados inclui segmentos da cadeia de linha que fazem parte destes segmentos que cruzam o raio.

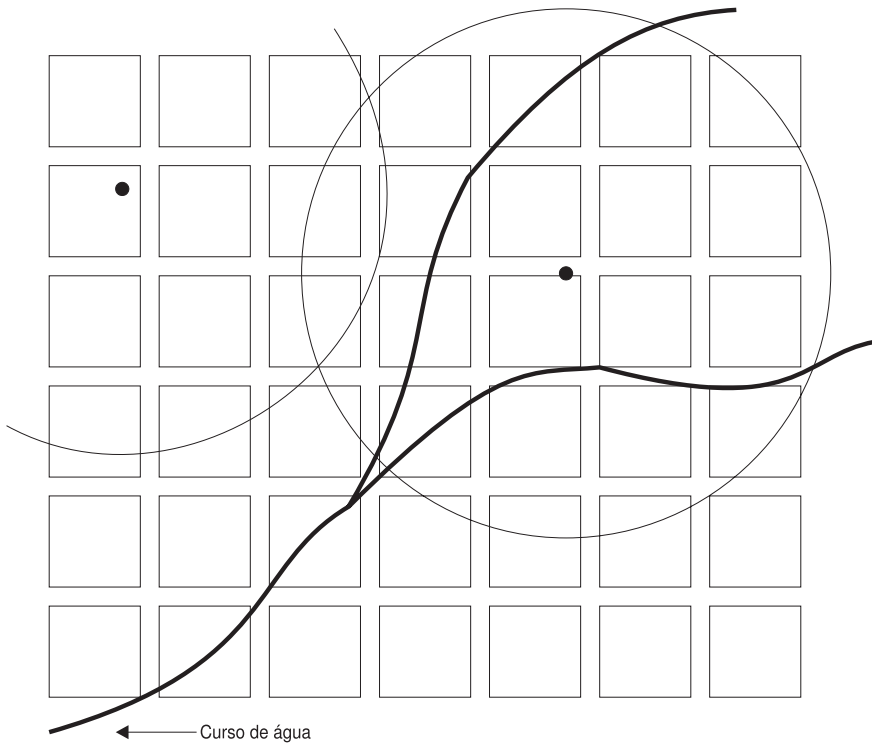


Figura 31. Utilizando `ST_Crosses` para encontrar cursos de água que passam por área de lixo tóxico

ST_Difference

ST_Difference toma dois objetos de figura geométrica e retorna um objeto de figura geométrica que é a diferença dos objetos de origem. As duas figuras geométricas de entrada de ST_Difference devem ter a mesma dimensão ou um valor nulo será retornado.

Sintaxe

```
db2gse.ST_Difference(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O engenheiro da cidade precisa saber a área total da área de lotes da cidade não abrangida por uma construção. Ou seja, o engenheiro da cidade deseja a soma da área de lotes após a remoção da área de construção.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                     lot_id         integer,
                                     footprint      db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id   integer,
                    lot      db2gse.ST_MultiPolygon);
```

O engenheiro da cidade reúne a tabela BUILDINGFOOTPRINTS e LOTS no lot_id. O engenheiro então soma a área de diferença dos lotes menos as bases de construção.

```
SELECT SUM(db2gse.ST_Area(db2gse.ST_Difference(lot,footprint)))
FROM BUILDINGFOOTPRINTS bf, LOTS
WHERE bf.lot_id = LOTS.lot_id;
```

ST_Dimension

ST_Dimension toma um objeto de figura geométrica e retorna sua dimensão.

Sintaxe

```
db2gse.ST_Dimension(g1 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A tabela DIMENSION_TEST é criada com as colunas GEOTYPE e G1. A coluna GEOTYPE armazena o nome da subclasse de figura geométrica armazenada na coluna de figura geométrica G1.

```
CREATE TABLE DIMENSION_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

As instruções INSERT inserem uma subclasse de amostra na tabela DIMENSION_TEST.

```
INSERT INTO DIMENSION_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
```

```
db2gse.coordref()..srid(0))
```

```
INSERT INTO DIMENSION_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94,10.02 20.01))',
```

```
db2gse.coordref()..srid(0))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64),(9.55 23.75,15.36 30.11))',
```

```
db2gse.coordref()..srid(0))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
```

```

                25.02 34.15,19.15 33.94,10.02 20.01)),
                ((51.71 21.73,73.36 27.04,71.52 32.87,
                 52.43 31.90,51.71 21.73)))',
db2gse.coordref)..srid(0)))

```

A seguinte instrução `SELECT` relaciona o nome da subclasse armazenado na coluna `GEOTYPE` à dimensão desse geotipo.

```

SELECT geotype, db2gse.ST_Dimension(g1) "A dimensão"
FROM DIMENSION_TEST

```

O seguinte conjunto de resultados retorna.

| GEOTYPE | The dimension |
|--------------------|---------------|
| ----- | ----- |
| ST_Point | 0 |
| ST_Linestring | 1 |
| ST_Polygon | 2 |
| ST_Multipoint | 0 |
| ST_Multilinestring | 1 |
| ST_Multipolygon | 2 |

6 record(s) selected.

ST_Disjoint

ST_Disjoint toma duas figuras geométricas e retorna 1 (VERDADEIRO) se a interseção de duas figuras geométricas produzir um conjunto vazio; do contrário retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Disjoint(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

Uma companhia de seguros precisa orçar a cobertura de seguro para os hospitais, casas de repouso e escolas de uma cidade. Parte deste processo inclui a determinação da ameaça que os locais de lixo tóxico representam para cada instituição. A companhia de seguro pretende considerar somente as instituições que estejam com risco de contaminação. O consultor GIS contratado pela companhia de seguros foi encarregado de localizar todas as instituições que não estejam dentro do raio de 8 km de um depósito de lixo tóxico.

A tabela SENSITIVE_AREAS contém várias colunas que descrevem as instituições ameaçadas além da coluna ZONE, que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

A tabela HAZARDOUS_SITES armazena o identificador dos locais nas colunas SITE_ID e NAME, ao passo que a localização geográfica real de cada local é armazenada na coluna LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

A seguinte instrução SELECT relaciona os nomes de todas as áreas sensíveis que não estejam dentro do raio de 8 km de um local de lixo tóxico. A função ST_Intersects pode substituir a função ST_Disjoint nesta consulta se o resultado da função estiver definido igual a 0 em vez de 1. Isto porque ST_Intersects e ST_Disjoint retornam o resultado oposto exato.

```
SELECT sa.name
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Disjoint(db2gse.ST_Buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```


Na Figura 32, são sensíveis os locais comparados ao raio de 8 km do lixo tóxico. A casa de repouso é a única área sensível em que a função `ST_Disjoint` retornará 1 (VERDADEIRO). A função `ST_Disjoint` retorna 1 sempre que duas figuras geométricas não fazem interseção de alguma forma.

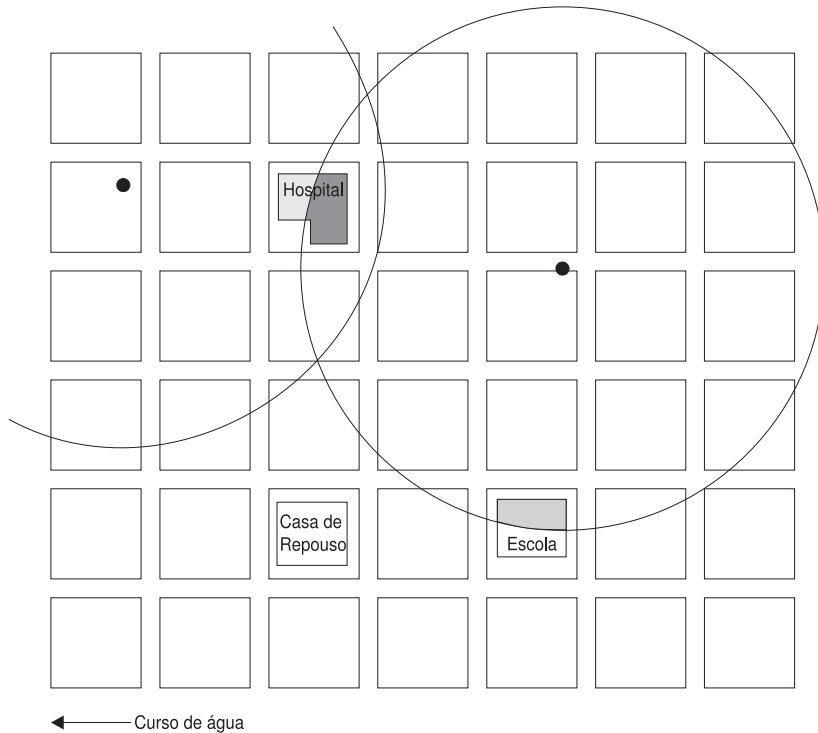


Figura 32. Usando `ST_Disjoint` para encontrar as construções que não estejam dentro de (interseção) alguma área de lixo tóxico

ST_Distance

ST_Distance toma duas figuras geométricas e retorna a distância mais próxima que as separa.

Sintaxe

```
db2gse.ST_Distance(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Duplo

Exemplos

O engenheiro da cidade precisa de uma lista de todas as construções que estejam dentro de 30 cm de qualquer linha do lote.

A coluna BUILDINGI_ID da tabela BUILDINGFOOTPRINTS identifica exclusivamente cada construção. A coluna LOT_ID identifica o lote a que cada construção pertence. O multipolígono da base armazena a figura geométrica de cada base da construção.

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

A tabela LOTS armazena a ID do lote que identifica exclusivamente cada lote e o multipolígono do lote que contém a figura geométrica de linha do lote.

```
CREATE TABLE LOTS (
    lot_id integer,
    lot     db2gse.ST_MultiPolygon);
```

A consulta retorna uma lista de IDs de construção que estejam dentro de 30 cm das linhas do lote. A função ST_Distance executa uma junção espacial entre as bases e o limite dos multipolígonos do lote. No entanto, a equijunção entre bf.lot_id LOTS.lot_id assegura que somente os multipolígonos pertencentes ao mesmo lote sejam comparados pela função ST_Distance.

```
SELECT bf.building_id
FROM BUILDINGFOOTPRINTS bf, LOTS
WHERE bf.lot_id = LOTS.lot_id AND
      db2gse.ST_Distance(bf.footprint, db2gse.ST_Boundary(LOTS.lot)) <= 1.0;
```

ST_Endpoint

ST_Endpoint toma uma cadeia de linha e retorna um ponto que é o último da cadeia.

Sintaxe

```
db2gse.ST_Endpoint(c db2gse.ST_Curve)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

A tabela ENDPOINT_TEST armazena a coluna inteira de GID que identifica exclusivamente cada linha na cadeia de linha LN1 que armazena cadeias de linhas.

```
CREATE TABLE ENDPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

As instruções INSERT inserem cadeias de linhas na tabela ENDPOINT_TEST. A primeira não possui coordenadas Z ou medidas; a segunda possui.

```
INSERT INTO ENDPOINT_TEST
VALUES( 1,
       db2gse.ST_LineFromText('linestring (10.02 20.01,23.73 21.92,
                               30.10 40.23)', db2gse.coordref()..srid(0))
```

```
INSERT INTO ENDPOINT_TEST
VALUES (2,
       db2gse.ST_LineFromText('linestring zm (10.02 20.01 5.0 7.0,
                               23.73 21.92 6.5 7.1, 30.10 40.23 6.9 7.2)',
```

```
db2gse.coordref()..srid(0))
```

A seguinte instrução SELECT relaciona a coluna GID à saída da função ST_Endpoint. A função ST_Endpoint gera uma figura geométrica de ponto que é convertida em texto pela função ST_AsText. A função CAST é usada para abreviar o valor varchar(4000) da função ST_AsText para um varchar(60).

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_Endpoint(ln1)) AS varchar(60))
"Endpoint"
FROM ENDPOINT_TEST
```

O seguinte conjunto de resultados retorna.

```
GID      Endpoint
-----
1 POINT ( 30.10000000 40.23000000)
2 POINT ZM ( 30.10000000 40.23000000 7.00000000 7.20000000)
```

```
2 record(s) selected.
```

ST_Envelope

ST_Envelope toma um objeto de figura geométrica e retorna seu quadro delimitador como uma figura geométrica.

Sintaxe

```
db2gse.ST_Envelope(g db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A coluna GEOTYPE na tabela ENVELOPE_TEST armazena o nome da subclasse de figura geométrica armazenada na coluna de figura geométrica G1.

```
CREATE TABLE ENVELOPE_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

As seguintes instruções INSERT inserem cada subclasse de figura geométrica na tabela ENVELOPE_TEST.

```
INSERT INTO ENVELOPE_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))

INSERT INTO ENVELOPE_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.01 20.01, 10.01 30.01,
10.01 40.01)', db2gse.coordref()..srid(0)))

INSERT INTO ENVELOPE_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)',
db2gse.coordref()..srid(0)))

INSERT INTO ENVELOPE_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94,10.02 20.01))',
db2gse.coordref()..srid(0)))

INSERT INTO ENVELOPE_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32
23.98,11.92 25.64)',
db2gse.coordref()..srid(0)))

INSERT INTO ENVELOPE_TEST
```

```

VALUES('Multilinestring',
       db2gse.ST_MLineFromText('multilinestring ((10.01 20.01,20.01 20.01,
       30.01 20.01), (30.01 20.01,40.01 20.01,50.01 20.01))'),
       db2gse.coordref()..srid(0))

INSERT INTO ENVELOPE_TEST
VALUES('Multilinestring',
       db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
       11.92 25.64), ( 9.55 23.75,15.36 30.11))'),
       db2gse.coordref()..srid(0))

INSERT INTO ENVELOPE_TEST
VALUES('Multipolygon',
       db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
       25.02 34.15, 19.15 33.94,10.02 20.01)),
       ((51.71 21.73,73.36 27.04,71.52 32.87,
       52.43 31.90,51.71 21.73)))'),
       db2gse.coordref()..srid(0))

```

A seguinte instrução SELECT relaciona o nome da subclasse ao lado de seu envelope. Como a função ST_Envelope retorna um ponto, uma cadeia de linha ou um polígono, sua saída será convertida em texto com a função ST_AsText. A função CAST converte o resultado padrão varchar(4000) da função ST_AsText para um varchar(280).

```

SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_Envelope(g1)) AS varchar(280))
"The envelope"
FROM ENVELOPE_TEST

```

O seguinte conjunto de resultados retorna.

| GEOTYPE | The envelope |
|--|---|
| Point | POINT (10.02000000 20.01000000) |
| Linestring 40.01000000) | LINESTRING (10.01000000 20.01000000, 10.01000000 |
| Linestring 20.01000000, 11.92000000 25.64000000, 10.02000000 25.64000000, 10.02000000 20.01000000)) | POLYGON ((10.02000000 20.01000000, 11.92000000 |
| Polygon 20.01000000, 25.02000000 35.64000000, 10.02000000 35.64000000, 10.02000000 20.01000000)) | POLYGON ((10.02000000 20.01000000, 25.02000000 |
| Multipoint 20.01000000, 11.92000000 25.64000000, 10.02000000 25.64000000, 10.02000000 20.01000000)) | POLYGON ((10.02000000 20.01000000, 11.92000000 |
| Multilinestring 20.01000000) | LINESTRING (10.01000000 20.01000000, 50.01000000 |
| Multilinestring 20.01000000, 15.36000000 30.11000000, 9.55000000 30.11000000, 9.55000000 20.01000000)) | POLYGON ((9.55000000 20.01000000, 15.36000000 |
| Multipolygon | POLYGON ((10.02000000 20.01000000, 73.36000000 |

20.01000000, 73.36000000 35.64000000, 10.02000000 35.64000000, 10.02000000
20.01000000))

8 record(s) selected.

ST_Equals

ST_Equals compara duas figuras geométricas e retorna 1 (VERDADEIRO) se as figuras geométricas são idênticas; do contrário, retorna 0 (FALSO).

Sintaxe

```
db2gse.ST_Equals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

O técnico GIS da cidade suspeita que alguns dados na tabela BUILDINGFOOTPRINTS foram duplicados de alguma forma. O técnico consulta a tabela para determinar se algum multipolígono da base é igual.

A tabela BUILDINGFOOTPRINTS foi criada com a seguinte instrução. A coluna BUILDING_ID identifica as construções; a coluna LOT_ID identifica o lote da construção e a coluna FOOTPRINT armazena a figura geométrica da construção.

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

A tabela BUILDINGFOOTPRINTS é unida a si própria espacialmente pelo predicado ST_Equals, que retorna 1 sempre que encontra dois multipolígonos que são iguais. A condição bf1.building_id <> bf2.building_id é necessária para eliminar a comparação da mesma figura geométrica.

```
SELECT bf1.building_id, bf2.building_id
FROM BUILDINGFOOTPRINTS bf1, BUILDINGFOOTPRINTS bf2
WHERE db2gse.ST_Equals(bf1.footprint,bf2.footprint) = 1
      e bf1.building_id <> bf2.building_id;
```

ST_ExteriorRing

ST_ExteriorRing toma um polígono e retorna seu anel externo como uma cadeia de linha.

Sintaxe

```
db2gse.ST_ExteriorRing(s db2gse.ST_Polygon)
```

Tipo de retorno

```
db2gse.ST_LineString
```

Exemplos

Um ornitologista que está estudando a população de pássaros em várias ilhas marinhas do sul, sabe que a zona de alimentação de uma determinada espécie de pássaro restringe-se ao litoral. Para calcular a capacidade de transporte da ilha, o ornitologista solicita o perímetro da ilha. Embora algumas ilhas tenham vários lagos, as linhas de contorno dos lagos são habitadas exclusivamente por outras espécies de pássaros mais agressivas. Portanto, ele solicita o perímetro do anel externo das ilhas.

As colunas ID e NAME da tabela ISLANDS identificam cada ilha e a coluna LAND do tipo ST_Polygon armazena a figura geométrica de cada uma.

```
CREATE TABLE ISLANDS (id    integer,  
                        name  varchar(32),  
                        land  db2gse.ST_Polygon);
```

A função ST_ExteriorRing extrai o anel externo de cada polígono da ilha como uma cadeia de linha. O comprimento da cadeia de linha é estabelecido pela função comprimento. Os comprimentos das cadeias de linha são resumidos pela função SUM.

```
SELECT SUM(db2gse.ST_length(db2gse.ST_ExteriorRing (land))) FROM ISLANDS;
```

Na Figura 33 na página 251, os anéis externos das ilhas representam a interface ecológica que cada ilha compartilha com o mar. Algumas ilhas têm lagos, que são representados pelos anéis interiores dos polígonos.

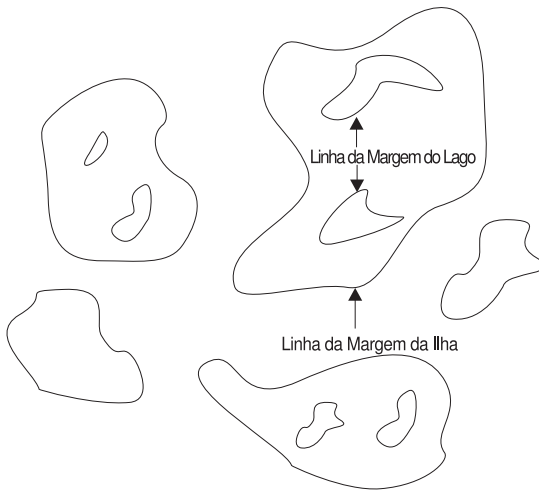


Figura 33. Utilizando `ST_ExteriorRing` para determinar o comprimento do contorno da ilha

ST_GeometryN

ST_GeometryN toma uma coleção e um índice inteiro e retorna o objeto de figura geométrica *n* na coleção.

Sintaxe

```
db2gse.ST_GeometryN(g db2gse.ST_Geometry, n Integer)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O engenheiro da cidade precisa saber se as bases da construção estão todas dentro do primeiro polígono do multipolígono do lote.

A coluna BUILDING_ID identifica exclusivamente cada linha da tabela BUILDINGFOOTPRINTS. A coluna LOT_ID identifica o lote da construção. A coluna BASE armazena a figura geométrica da construção.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                   lot_id      integer,
                                   footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id   integer,
                    lot      db2gse.ST_MultiPolygon);
```

A consulta lista BUILDINGFOOTPRINTS building_id e lot_id para todas as bases da construção que estejam dentro do primeiro polígono do lote. A função ST_GeometryN retorna o primeiro polígono de um lote na matriz de polígonos.

```
SELECT bf.building_id,bf.lot_id
FROM BUILDINGFOOTPRINTS bf,LÔTS
WHERE db2gse.ST_Within(footprint, db2gse.ST_GeometryN (lot,1)) = 1
      AND bf.lot_id = LOTS.lot_id;
```

ST_GeometryType

O ST_GeometryType pega um objeto ST_Geometry e apresenta o tipo de sua figura geométrica como a cadeia.

Sintaxe

```
db2gse.ST_GeometryType (g db2gse.ST_Geometry)
```

Tipo de retorno

```
Varchar(4000)
```

Exemplos

A tabela GEOMETRYTYPE_TEST contém a coluna de figura geométrica G1.

```
CREATE TABLE GEOMETRYTYPE_TEST(g1 db2gse.ST_Geometry)
```

As seguintes instruções INSERT inserem cada subclasse de figura geométrica na coluna G1.

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES (db2gse.ST_GeomFromText('linestring (10.01 20.01, 10.01 30.01,
10.01 40.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_Geometrytype_test values(db2gse.ST_GeomFromText('polygon
((10.02 20.01,11.92 35.64,25.02 34.15,19.15 33.94, 10.02 20.01))',
db2gse.coordref()..srid(0))))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multipoint (10.02
20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64),
(9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))',
db2gse.coordref()..srid(0)))
```

A seguinte instrução INSERT insere o tipo de figura geométrica de cada subclasse que está armazenada na coluna de figura geométrica G1.

```
SELECT db2gse.ST_GeometryType(g1) "Geometry type" FROM GEOMETRYTYPE_TEST
```

O seguinte conjunto de resultados retorna.

```
Geometry type
```

```
-----
```

```
ST_Point
```

```
ST_LineStringST_Polygon
```

```
ST_MultiPointST_MultiLineStringST_MultiPolygon
```

```
6 record(s) selected.
```

ST_GeomFromText

ST_GeomFromText recebe uma representação de um texto convencional e um identificador do sistema de referências espaciais e retorna um objeto de figura geométrica.

Sintaxe

```
db2gse.ST_GeomFromText(geometryTaggedText Varchar(4000), SRID
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A tabela GEOMETRY_TEST contém a coluna GID, que identifica exclusivamente cada linha e a coluna G1, que armazena a figura geométrica.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

As instruções INSERT inserem os dados nas colunas GID e G1 da tabela GEOMETRY_TEST. A função ST_GeomFromText converte a representação de texto de cada figura geométrica na sua subclasse instanciável correspondente Spatial Extender.

```
INSERT INTO GEOMETRY_TEST
VALUES(1, db2gse.ST_GeomFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES (2,
        db2gse.ST_GeomFromText('linestring (10.01 20.01, 10.01 30.01,
        10.01 40.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(3,
        db2gse.ST_GeomFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
        19.15 33.94,10.02 20.01))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(4,
        db2gse.ST_GeomFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(5,
        db2gse.ST_GeomFromText('multilinestring ((10.02 20.01,10.32 23.98,
        11.92 25.64),
        ( 9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(6,
      db2gse.ST_GeomFromText('multipolygon (((10.02 20.01,11.92 35.64,
      25.02 34.15, 19.15 33.94,10.02 20.01)),
      ((51.71 21.73,73.36 27.04,71.52 32.87,
      52.43 31.90,51.71 21.73)))',
      db2gse.coordref()..srid(0)))
```

ST_GeomFromWKB

ST_GeomFromWKB recebe uma representação de binário convencional e um identificador do sistema de referências espaciais e retorna um objeto de figura geométrica.

Sintaxe

```
db2gse.ST_GeomFromWKB(WKBGeometry Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O seguinte fragmento de código C contém funções ODBC incorporadas às funções SQL do Spatial Extender que inserem dados na tabela LOTS.

A tabela LOTS foi criada com duas colunas: a coluna LOT_ID, que identifica cada um dos lotes exclusivamente, e a coluna multipolígono do LOT, que contém a figura geométrica de cada lote.

```
CREATE TABLE LOTS ( lot_id integer,
                    lot      db2gse.ST_MultiPolygon);
```

A função ST_GeomFromWKB converte representações de WKB em figura geométrica do Spatial Extender. A instrução INSERT inteira será copiada na cadeia wkb_sql char. A instrução INSERT contém marcadores do parâmetro para aceitar os dados do LOT_ID e os dados do LOT dinamicamente.

```
/* Criar a instrução insert do SQL para preencher o id do lote e o
   multipolígono do lote. Os pontos de interrogação são marcadores
   do parâmetro que
   indicam a id do_lote e os valores do lote
   que serão recuperados no tempo de execução.*/
strcpy (wkb_sql,"insert into LOTS (lot_id, lot) values (?,
   db2gse.ST_GeomFromWKB

   (cast(? as blob(1m)), db2gse.coordref(..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Ligar o valor inteiro da chave ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
   SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* Ligar o formato ao segundo parâmetro. */
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```


ST_InteriorRingN

Retorna o n anel interior de um polígono como uma cadeia de linha. Os anéis não estão organizados por orientação geométrica. Estão organizados de acordo com as regras definidas pelas rotinas de verificação de figura geométrica interna. Assim, não é possível predefinir a ordem dos anéis.

Sintaxe

ST_InteriorRingN(p ST_Polygon, n Integer)

Tipo de retorno

db2gse.ST_LineString

Exemplos

Um ornitologista, que está estudando a população de pássaros em várias ilhas marinhas do sul, sabe que a zona de alimentação de uma determinada espécie passiva restringe-se à costa. Algumas ilhas têm vários lagos. As praias dos lagos são habitadas exclusivamente por outras espécies mais agressivas. O ornitologista sabe que para cada ilha, se o perímetro do lago exceder um determinado limite, as espécies agressivas se tornarão mais numerosas e ameaçarão as espécies passivas do litoral. Portanto, ele solicita o perímetro agregado dos anéis internos das ilhas.

Na Figura 34, os anéis externos das ilhas representam a interface ecológica que cada ilha compartilha com o mar. Algumas ilhas têm lagos, que são representados pelos anéis interiores dos polígonos.

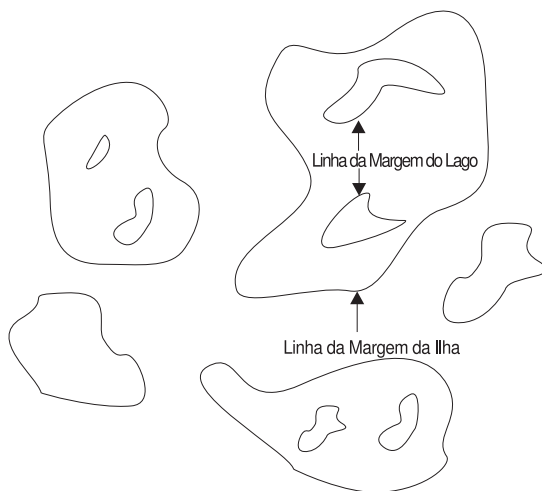


Figura 34. Usando ST_InteriorRingN para determinar o comprimento das margens dos lagos dentro de cada ilha

As colunas ID e NAME da tabela ISLANDS identificam cada ilha, ao passo que a coluna polígono de terra armazena a figura geométrica da ilha.

```
CREATE TABLE ISLANDS (id    integer,
                       name  varchar(32),
                       land  db2gse.ST_Polygon);
```

O seguinte programa ODBC utiliza a função ST_InteriorRingN para extrair o anel interior (lago) de cada polígono da ilha como uma cadeia de linhas. O perímetro da cadeia de linhas que a função comprimento retorna é somado e exibido junto à ID da ilha.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sg.h"
#include "sgerr.h"
#include "sqlcli1.h"

/**          ***
*** Altere estas constantes ***
***          ***/

#define USER_NAME    "sdetest" /* nome de usuário */
#define USER_PASS    "acid.rain" /* sua senha de usuário */
#define DB_NAME      "mydb" /* banco de dados ao qual se conectar */

static void check_sql_err (SQLHDBC  handle,
                          SQLHSTMT hstmt,
                          LONG      rc,
                          CHAR      *str);

void main( argc, argv )
int  argc;
char *argv[];
{
    SQLHDBC  handle;
    SQLHENV  henv;
    CHAR     sql_stmt[256];
    LONG     rc,
            total_perimeter,
            num_lakes,
            lake_number,
            island_id,
            lake_perimeter;
    SQLHSTMT island_cursor,
            lake_cursor;
    SDWORD   pcbvalue,
            id_ind,
            lake_ind,
            length_ind;

    /* Alocar memória para a manipulação de ambiente ODBC henv e
```

```

inicializar a aplicação. */

    rc = SQLAllocEnv (&henv);
    if (rc != SQL_SUCCESS)
    {
        printf ("SQLAllocEnv failed with %d\n", rc);
        exit(0);
    }

/* Alocar memória para o manipulador da conexão dentro do ambiente henv. */

    rc = SQLAllocConnect (henv, &handle);
    if (rc != SQL_SUCCESS)
    {
        falha de printf ("SQLAllocConnect com %d\n", rc);
        exit(0);
    }

/* Carregar o controlador ODBC e conectar-se à fonte de dados
   identificada pelo banco de dados, usuário e senha.*/

    rc = SQLConnect (handle,
                    (UCHAR *)DB_NAME,
                    SQL_NTS,
                    (UCHAR *)USER_NAME,
                    SQL_NTS,
                    (UCHAR *)USER_PASS,
                    SQL_NTS);

    check_sql_err (handle, NULL, rc, "SQLConnect");

/* Alocar memória para a instrução SQL manipular island_cursor. */

    rc = SQLAllocStmt (handle, &island_cursor);
    check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* Preparar e executar a consulta para obter as IDs e número da ilha de
   lagos (anéis interiores) */

    strcpy (sql_stmt, "select id, db2gse.ST_NumInteriorRings(land) from ISLANDS");

    rc = SQLExecDirect (island_cursor, (UCHAR *)sql_stmt, SQL_NTS);
    check_sql_err (NULL, island_cursor, rc, "SQLExecDirect");

/* Ligar a coluna ID da tabela island à variável island_id */

    rc = SQLBindCol (island_cursor, 1, SQL_C_SLONG, &island_id, 0, &id_ind);
    check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* Ligar o resultado de numinteriorrings(land) à variável num_lakes. */

    rc = SQLBindCol (island_cursor, 2, SQL_C_SLONG, &num_lakes, 0, &lake_ind);
    check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* Alocar memória para o manipulador da instrução SQL lago_cursor. */

```

```

rc = SQLAllocStmt (handle, &lake_cursor);
check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* Preparar a consulta para obter o comprimento de um anel interior.*/

strcpy (sql_stmt,
        "select Length(db2gse.ST_InteriorRingN(land, cast (? as
integer))) from ISLANDS where id = ?");

rc = SQLPrepare (lake_cursor, (UCHAR *)sql_stmt, SQL_NTS);
check_sql_err (NULL, lake_cursor, rc, "SQLPrepare");

/* Ligar a variável lake_number como primeiro parâmetro de entrada */

pcbvalue = 0;
rc = SQLBindParameter (lake_cursor, 1, SQL_PARAM_INPUT, SQL_C_LONG,
        SQL_INTEGER, 0, 0, &lake_number, 0, &pcbvalue);
check_sql_err (NULL, lake_cursor, rc, "SQLBindParameter");

/* Ligar o id da ilha como segundo parâmetro de entrada */

pcbvalue = 0;
rc = SQLBindParameter (lake_cursor, 2, SQL_PARAM_INPUT, SQL_C_LONG,
        SQL_INTEGER, 0, 0, &island_id, 0, &pcbvalue);
check_sql_err (NULL, lake_cursor, rc, "SQLBindParameter");

/* Ligar o resultado de Length(db2gse.ST_InteriorRingN(land, cast
(? como inteiro ))) à variável lake_perimeter */

rc = SQLBindCol (lake_cursor, 1, SQL_C_SLONG, &lake_perimeter, 0,
        &length_ind);
check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* Loop externo, obter as ids da ilha e número de lagos
(anéis interiores) */

while (SQL_SUCCESS == rc)
{
    /* Buscar uma ilha */

    rc = SQLFetch (island_cursor);

    if (rc != SQL_NO_DATA)
    {
        check_sql_err (NULL, island_cursor, rc, "SQLFetch");

        /* Loop interno, desta ilha, obter o perímetro de todos os
        seus lagos (anéis internos) */

        para (perímetro_total = 0, lake_number = 1;
            lake_number <= num_lakes;
            lake_number++)
        {
            rc = SQLExecute (lake_cursor);

```

```

        check_sql_err (NULL, lake_cursor, rc, "SQLExecute");

        rc = SQLFetch (lake_cursor);
        check_sql_err (NULL, lake_cursor, rc, "SQLFetch");

        total_perimeter += lake_perimeter;

        SQLFreeStmt (lake_cursor, SQL_CLOSE);
    }
}

/* Exibir a id da ilha e o perímetro total de seus lagos. */

    printf ("Island ID = %d, Total lake perimeter = %d\n",
            island_id, total_perimeter);

}

SQLFreeStmt (lake_cursor, SQL_DROP);
SQLFreeStmt (island_cursor, SQL_DROP);
    SQLDisconnect (handle);
    SQLFreeConnect (handle);
SQLFreeEnv (henv);

printf( "\nTest Complete ...\n" );

}

static void check_sql_err (SQLHDBC handle, SQLHSTMT hstmt, LONG rc,
CHAR      *str)
{

    SDWORD dbms_err = 0;
    SWORD  length;
    UCHAR  err_msg[SQL_MAX_MESSAGE_LENGTH], state[6];

    if (rc != SQL_SUCCESS)
    {
        SQLError (SQL_NULL_HENV, handle, hstmt, state, &dbms_err,
                err_msg, SQL_MAX_MESSAGE_LENGTH - 1, &length);
        printf ("%s ERROR (%d): DBMS code:%d, SQL state: %s, message:
                \n %s\n", str, rc, dbms_err, state, err_msg);

        if (handle)
        {
            SQLDisconnect (handle);
            SQLFreeConnect (handle);
        }
        exit(1);
    }
}
}

```

ST_Intersection

ST_Intersection toma dois objetos de figura geométrica e retorna o conjunto de interseção como um objeto de figura geométrica.

Se ST_Intersection receber um polígono e um segmento de reta como parâmetros de entrada e se forem aplicáveis as seguintes condições:

- Um ponto do polígono é o ponto inicial do segmento de reta, e
 - o polígono e o segmento de reta não têm outros pontos em comum,
- então ST_Intersection retornará a cadeia de texto, POINT EMPTY.

Sintaxe

```
db2gse.ST_Intersection(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O oficial encarregado do serviço contra incêndios deve obter as áreas de hospitais, escolas e casas de repouso que fazem interseção com o raio de uma possível contaminação por lixo tóxico.

As áreas sensíveis são armazenadas na tabela SENSITIVE_AREAS que é criada com a seguinte instrução CREATE TABLE. A coluna ZONE é definida como um polígono, que armazena a descrição de cada área sensível.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

As áreas de risco são armazenadas na tabela HAZARDOUS_SITES que foi criada com a seguinte instrução CREATE TABLE. A coluna LOCATION, definida como um ponto, armazena uma localização que é o centro geográfico de cada local arriscado.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

A função de buffer gera um buffer de 8 km ao redor das áreas do lixo tóxico. A função ST_Intersection gera polígonos da interseção dos polígonos de locais de lixo tóxico em buffer e das áreas sensíveis. A função ST_Area retorna a área de polígono da interseção, que é resumida para cada local de risco pela função SUM. A cláusula GROUP BY direciona a consulta para agregar as áreas de interseção pela ID do local do lixo tóxico.

```

SELECT hs.name,SUM(db2gse.ST_Area(db2gse.ST_Intersection (sa.zone,
db2gse.ST_buffer hs.location,(5 * 5280))))
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
GROUP BY hs.site_id;

```

Na Figura 35, os círculos representam os polígonos em buffer ao redor de locais de lixo tóxico. A interseção destes polígonos em buffer com os polígonos da área sensível produz três outros polígonos. O hospital no canto superior esquerdo faz interseção duas vezes, ao passo que a escola no canto inferior direito faz apenas uma vez.

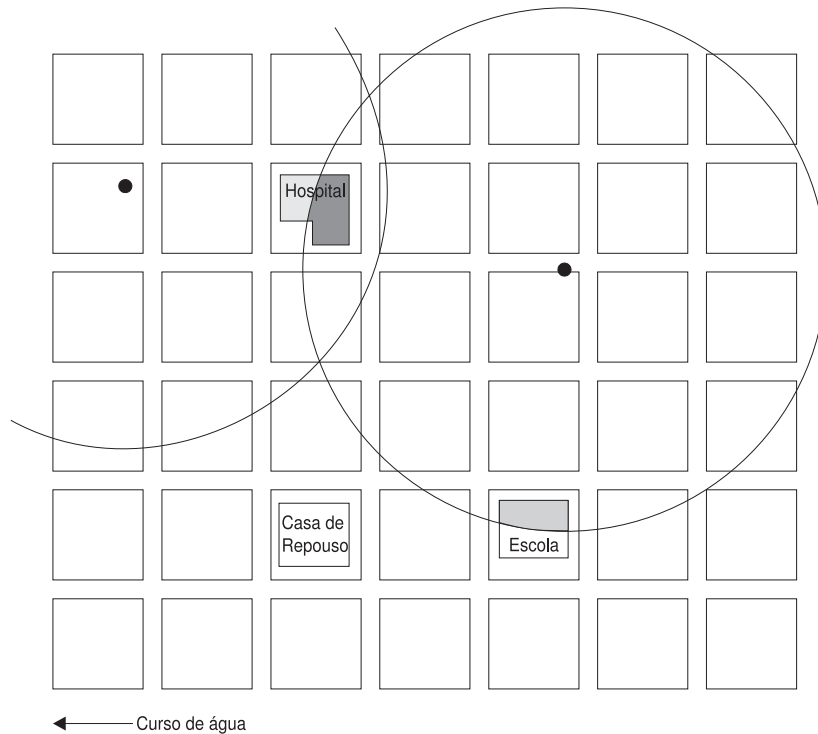


Figura 35. Utilizando ST_Intersection para determinar o quanto uma área em cada construção poderá ser afetada pelo lixo tóxico

ST_Intersects

ST_Intersects toma duas figuras geométricas e retorna 1 (VERDADEIRO), se a interseção das duas figuras geométricas não resultar num conjunto vazio. Do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Intersects(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

O oficial encarregado do serviço contra incêndios precisa de uma lista de todas as áreas sensíveis dentro de um raio de 8 km de uma área de lixo tóxico.

As áreas sensíveis são armazenadas na tabela SENSITIVE_AREAS que é criada com a seguinte instrução CREATE TABLE. A coluna ZONE é definida como um polígono, que armazena a descrição de cada área sensível.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

As áreas de risco são armazenadas na tabela HAZARDOUS_SITES criada com a seguinte instrução CREATE TABLE. A coluna LOCATION, definida como um ponto, armazena uma localização que é o centro geográfico de cada local arriscado.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name        varchar(128),
                               location   db2gse.ST_Point);
```

A consulta retorna uma lista de áreas sensíveis e nomes de áreas de risco para as áreas sensíveis que fazem interseção com o buffer de 8 km dos locais de risco.

```
SELECT sa.name, hs.name
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Intersects(db2gse.ST_Buffer(hs.location,(5 * 5280)),sa.zone)
=1;
```

ST_IsClosed

ST_IsClosed toma uma cadeia de linhas ou cadeia de linhas múltiplas e retorna 1 (VERDADEIRO) se estiver fechado; do contrário, retorna 0 (FALSO).

Sintaxe

```
db2gse.ST_IsClosed(c db2gse.ST_Curve)
db2gse.ST_IsClosed(mc db2gse.ST_MultiCurve)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela CLOSED_LINestring, que tem uma coluna de cadeia de linhas simples.

```
CREATE TABLE CLOSED_LINestring (ln1 db2gse.ST_LineString)
```

A seguinte instrução INSERT insere dois registros na tabela CLOSED_LINestring. O primeiro registro não é uma cadeia de linha fechada, ao passo que o segundo sim.

```
INSERT INTO CLOSED_LINestring
VALUES(db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO CLOSED_LINestring
VALUES(db2gse.ST_LineFromText('linestring (10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)',
```

```
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostra os resultados da função ST_IsClosed. A primeira linha retorna 0 porque a cadeia de linha não está fechada, ao passo que a segunda linha retorna 1 porque a cadeia de linha está fechada.

```
SELECT db2gse.ST_IsClosed(ln1) "Is it closed" FROM CLOSED_LINestring
```

```
Is it closed
-----
           0
           1
```

2 record(s) selected.

A seguinte instrução CREATE TABLE cria a tabela CLOSED_MULTILINESTRING, que tem uma coluna de cadeia de linhas múltiplas.

```
CREATE TABLE CLOSED_MULTILINESTRING (mln1 db2gse.ST_MultiLineString)
```

A seguinte instrução INSERT insere dois registros em CLOSED_MULTILINESTRING, um registro de cadeia de linhas múltiplas não está fechado e outro está.

```
INSERT INTO CLOSED_MULTILINESTRING
VALUES(db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64), (9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0)))

INSERT INTO CLOSED_MULTILINESTRING
VALUES(db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01),
(51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73))',
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostra os resultados da função ST_IsClosed. A primeira linha retorna 0 porque a cadeia de linhas múltiplas não está fechada, ao passo que a segunda linha retorna 1 porque a cadeia de linhas múltiplas está fechada. Uma cadeia de linhas múltiplas será fechada se todos os elementos da cadeia de linhas estiverem fechados.

```
SELECT db2gse.ST_IsClosed(mln1) "Is it closed" FROM CLOSED_MULTILINESTRING
```

```
Is it closed
-----
          0
          1
```

```
2 record(s) selected.
```

ST_IsEmpty

ST_IsEmpty toma um objeto de figura geométrica e retorna 1 (VERDADEIRO) se estiver vazio; do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_IsEmpty(g db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela EMPTY_TEST, com duas colunas. A coluna GEOTYPE armazena o tipo de dados das classes filhas que estão armazenadas na coluna de figura geométrica G1.

```
CREATE TABLE EMPTY_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

A seguinte instrução INSERT insere dois registros para o ponto, cadeia de linha e polígono da classes filhas de figura geométrica. Um registro está vazio e outro não.

```
INSERT INTO EMPTY_TEST
VALUES('Point', db2gse.ST_PointFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
INSERT INTO EMPTY_TEST
VALUES('Point', db2gse.ST_PointFromText('point empty',
db2gse.coordref()..srid(0)))
INSERT INTO EMPTY_TEST
VALUES('Linestring', db2gse.ST_LineFromText('linestring (10.02 20.01,
10.32 23.98, 11.92 25.64)',
db2gse.coordref()..srid(0)))

INSERT INTO EMPTY_TEST
VALUES('Linestring', db2gse.ST_LineFromText('linestring empty',
db2gse.coordref()..srid(0)))
INSERT INTO EMPTY_TEST
VALUES('Polygon', db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,
25.02 34.15,19.15 33.94,10.02 20.01))',
db2gse.coordref()..srid(0)))

INSERT INTO EMPTY_TEST
VALUES('Polygon', db2gse.ST_PolyFromText('polygon empty',
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostram a coluna tipo de figura geométrica e os resultados da função ST_IsEmpty.

```
SELECT geotype, db2gse.ST_IsEmpty(g1) "It is empty" FROM EMPTY_TEST

GEOTYPE                It is empty
```

```
-----  
ST_Point          0  
ST_Point          1  
ST_Linestring     0  
ST_Linestring     1  
ST_Polygon        0  
ST_Polygon        1
```

6 record(s) selected.

ST_IsRing

ST_IsRing toma uma cadeia de linhas e retorna 1 (VERDADEIRO) se for um anel (a saber, a cadeia de linhas é fechada e simples); caso contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_IsRing(c db2gse.ST_Curve)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela RING_LINestring, que tem uma coluna de cadeia de linhas simples denominada LN1.

```
CREATE TABLE RING_LINestring (ln1 db2gse.ST_LineString)
```

As seguintes instruções INSERT inserem três cadeias de linhas na coluna LN1. A primeira linha contém uma cadeia de linha que não é fechada e portanto não é um anel. A segunda linha contém uma cadeia de linhas que é fechada e é simples, portanto, é um anel. A terceira linha contém uma cadeia de linhas que está fechada mas não é simples porque faz interseção com seu próprio interior e, portanto, não é um anel.

```
INSERT INTO RING_LINestring
VALUES(db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO RING_LINestring
VALUES(db2gse.ST_LineFromText('linestring (10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94, 10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO RING_LINestring
VALUES(db2gse.ST_LineFromText('linestring (15.47 30.12,20.73 22.12,10.83 14.13,
16.45 17.24,21.56 13.37,11.23 22.56,
19.11 26.78,15.47 30.12)',
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostra os resultados da função ST_IsRing. A primeira e a terceira linha retornam 0. Isto porque as cadeias de linhas não são anéis, ao passo que a segunda linha retorna 1 porque é um anel.

```
SELECT db2gse.ST_IsRing(ln1) "Is it ring" FROM RING_LINestring
```

```
Is it ring
-----
0
```

1
0

3 record(s) selected.

ST_IsSimple

ST_IsSimple toma um objeto de figura geométrica e retorna 1 (VERDADEIRO) se o objeto for simples; do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_IsSimple(g db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela ISSIMPLE_TEST, a qual possui duas colunas. A coluna PID, que é smallint, contém o identificador exclusivo para cada linha. A coluna de figura geométrica G1 armazena as amostras de figura geométrica simples e não-simples.

```
CREATE TABLE ISSIMPLE_TEST (pid smallint, g1 db2gse.ST_Geometry)
```

A seguinte instrução INSERT insere dois registros na tabela ISSIMPLE_TEST. O primeiro é simples porque é uma cadeia de linha que não faz interseção com seu interior. O segundo é não-simples porque faz interseção com seu interior.

```
INSERT INTO ISSIMPLE_TEST
VALUES (1, db2gse.ST_LineFromText('linestring (10 10, 20 20, 30 30)',
db2gse.coordref()..srid(0)))
INSERT INTO ISSIMPLE_TEST
VALUES (2, db2gse.ST_LineFromText('linestring (10 10, 20 20,20 30,10 30,10 20,
20 10)', db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostram os resultados da função ST_IsSimple. O primeiro registro retorna 1 porque a cadeia de linha é simples, ao passo que o segundo registro retorna 0 porque a cadeia de linha não é simples.

```
SELECT ST_IsSimple(g1)
FROM ISSIMPLE_TEST
```

```
g1
-----
1
0
```

ST_IsValid

ST_IsValid toma um ST_Geometry e retorna 1 (VERDADEIRO) se for válido, do contrário, retornará 0 (FALSO). Uma figura geométrica inserida num banco de dados do DB2 sempre será válida, porque Spatial Extender sempre valida seus dados espaciais antes de aceitá-los. No entanto, outros fornecedores DBMS poderão não validar a entrada, mas sim solicitar que essa aplicação o faça.

Sintaxe

```
db2gse.ST_IsValid(g db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A tabela valid_test é criada com as colunas geotype e g1. A coluna geotype armazena o nome da subclasse de figura geométrica armazenada na coluna de figura geométrica g1.

```
CREATE TABLE valid_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

As instruções INSERT inserem uma subclasse de amostra na tabela valid_test.

```
INSERT INTO valid_test VALUES(
    'Point', db2gse.ST_PointFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Linestring',
    db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,11.92 25.64)',
db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Polygon', db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,
    25.02 34.15, 19.15 33.94,10.02 20.01))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multipoint', db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multilinestring', db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,
10.32 23.98,11.92 25.64),(9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multipolygon',
```



```
db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15,19.15 33.94,10.02 20.01)),((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))', db2gse.coordref()..srid(0))
)
```

A instrução SELECT relaciona o nome da subclasse armazenado na coluna geotype à dimensão desse geotipo.

```
SELECT geotype, db2gse.ST_IsValid(g1) Valid FROM valid_test
```

| GEOTYPE | Valid |
|--------------------|-------|
| ST_Point | 1 |
| ST_Linestring | 1 |
| ST_Polygon | 1 |
| ST_Multipoint | 1 |
| ST_Multilinestring | 1 |
| ST_Multipolygon | 1 |

6 record(s) selected.

ST_Length

ST_Length toma uma cadeia de linhas ou cadeia de linhas múltiplas e retorna seu comprimento.

Sintaxe

```
db2gse.ST_Length(c db2gse.ST_Curve)
db2gse.ST_Length(mc db2gse.ST_MultiCurve)
```

Tipo de retorno

Duplo

Exemplos

Um ecologista local está estudando os padrões migratórios da população de salmão nos cursos de água do município. O ecologista deseja obter o comprimento de todo o sistema fluvial que passa pelo município.

A seguinte instrução CREATE TABLE cria a tabela WATERWAYS. As colunas ID e NAME identificam cada sistema fluvial que está armazenado na tabela. A coluna ÁGUA é uma cadeia de linhas múltiplas pois os sistemas de rios e córregos são geralmente uma agregação de várias cadeias de linhas.

```
CREATE TABLE WATERWAYS (id integer, name varchar(128),
water db2gse.ST_MultiLineString);
```

A seguinte instrução SELECT utiliza a função ST_Length para retornar o nome e o comprimento de cada curso de água dentro do município.

```
SELECT name, db2gse.ST_Length(water) "Length"
FROM WATERWAYS;
```

a Figura 36 na página 277 exibe os sistemas fluviais que estão dentro dos limites do município.

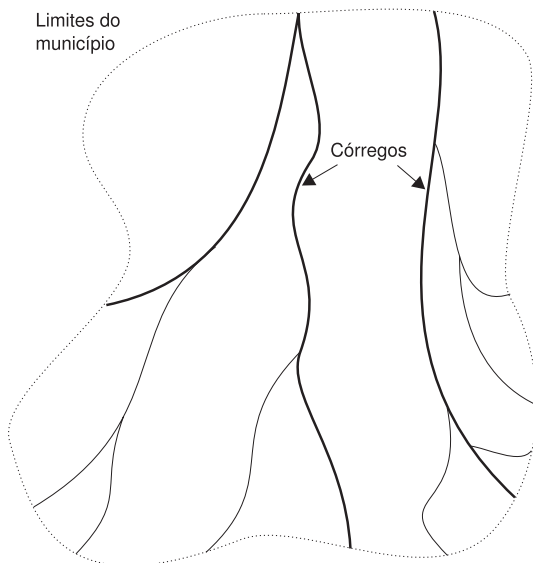


Figura 36. Utilizando ST_Length para determinar o comprimento total dos cursos de água num município

ST_LineFromText

ST_LineFromText recebe uma representação de texto convencional do tipo segmento de reta e um identificador do sistema de referências espaciais e retorna um segmento de reta.

Sintaxe

```
db2gse.ST_LineFromText(lineStringTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_LineString
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela LINESTRING_TEST, que tem uma coluna de cadeia de linhas simples LN1.

```
CREATE TABLE LINESTRING_TEST (ln1 db2gse.ST_LineString)
```

A seguinte instrução INSERT insere uma cadeia de linhas na coluna LN1 usando a função ST_LineFromText.

```
INSERT INTO LINESTRING_TEST  
VALUES (db2gse.ST_LineFromText('linestring(10.01 20.03,20.94 21.34,  
35.93 19.04)', db2gse.coordref()..srid(0)))
```

ST_LineFromWKB

ST_LineFromWKB recebe uma representação binária convencional do tipo segmento de reta e um identificador do sistema de referências espaciais e retorna um segmento de reta.

Sintaxe

```
db2gse.ST_LineFromWKB(WKBLineString Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_LineString
```

Exemplos

O seguinte fragmento de código preenche a tabela SEWERLINES com id, classe de tamanho e geometria exclusivos de cada linha sewer.

A tabela SEWERLINES foi criada com três colunas. A primeira coluna, SEWER_ID, identifica exclusivamente cada linha do sewer. A segunda coluna, CLASS, do tipo inteiro identifica o tipo de linha de sewer, que geralmente está associada à capacidade da linha. A terceira coluna, SEWER, do tipo cadeia de linhas armazena a figura geométrica da linha sewer.

```
CREATE TABLE SEWERLINES (sewer_id integer,
                        class integer,
                        sewer db2gse.ST_LineString);

/* Criar a instrução insert do SQL para preencher a id do sewer, o tamanho
   e a cadeia de linha sewer. Os pontos de interrogação são marcadores de
   parâmetros que indicam a id do sewer, a classe e os valores de figura
   geométrica do sewer que serão recuperados no tempo de execução. */
strcpy (wkb_sql,"insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.ST_LineFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Ligar o valor inteiro do id do sewer ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* Ligar o valor da classe de inteiro ao segundo parâmetro. */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);

/* Ligar o formato ao terceiro parâmetro. */
pcbvalue3 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, sewer_wkb, blob_len, &pcbvalue3);  
  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

ST_MLineFromText

ST_MLineFromText recebe uma representação de texto convencional do tipo segmento de reta múltiplo e um identificador do sistema de referência espacial e retorna um segmento de reta múltiplo.

Sintaxe

```
db2gse.ST_MLineFromText(multiLineStringTaggedText String, SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiLineString
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela MLINESTRING_TEST. MLINESTRING_TEST tem duas colunas: a coluna GID smallint, que identifica exclusivamente a linha e a coluna de cadeia de linhas múltiplas ML1.

```
CREATE TABLE ST_MLINESTRING_TEST (gid smallint, ml1 db2gse.ST_MultiLineString)
```

A seguinte instrução INSERT insere a cadeia de linhas múltiplas com a função ST_MLineFromText.

```
INSERT INTO MLINESTRING_TEST  
VALUES (1, db2gse.ST_MLineFromText('multilinestring((10.01 20.03,10.52 40.11,  
30.29 41.56,31.78 10.74),  
                                     (20.93 20.81, 21.52 40.10))',  
db2gse.coordref()..srid(0)))
```

ST_MLineFromWKB

ST_MLineFromWKB recebe uma representação binária convencional do tipo segmento de reta múltiplo e um identificador do sistema de referências espaciais e retorna um segmento de reta múltiplo.

Sintaxe

```
db2gse.ST_MLineFromWKB(WKBMultiLineString Blob(1M), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiLineString
```

Exemplos

O seguinte fragmento de código preenche a tabela WATERWAYS com uma id exclusiva, um nome e uma cadeia de linhas múltiplas de água.

A tabela WATERWAYS é criada com as colunas ID e NAME que identificam cada sistema de córregos e rios armazenados na tabela. A coluna ÁGUA é uma cadeia de linhas múltiplas pois os sistemas de rios e córregos são geralmente uma agregação de várias cadeias de linhas.

```
CREATE TABLE WATERWAYS (id      integer,  
                          name    varchar(128),  
                          water    db2gse.ST_MultiLineString);  
  
/* Criar a instrução insert do SQL para preencher a id, nome e  
   a cadeia de linhas múltiplas. Os pontos de interrogação são  
   marcadores de parâmetros que  
   indicam a id, o nome e os valores da água que serão recuperados no  
   tempo de execução.*/  
shp_sql="insert into WATERWAYS (id,name,water)  
values (?,?, db2gse.ST_MLineFromWKB (cast(? as blob(1m)),  
db2gse.coordref(..srid(0)))";  
  
/* Alocar memória para o tratamento da instrução SQL e associar o  
   tratamento da instrução ao tratamento da conexão. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Preparar as instruções SQL para execução. */  
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);  
  
/* Ligar o valor inteiro do id ao primeiro parâmetro.*/  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);  
  
/* Ligar o valor do nome varchar ao segundo parâmetro. */  
pcbvalue2 = name_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,  
SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);  
  
/* Ligar o formato ao terceiro parâmetro. */
```



```
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);
```

ST_MPointFromText

ST_MPointFromText recebe uma representação de texto convencional do tipo ponto múltiplo e um identificador do sistema de referências espaciais e retorna um ponto múltiplo.

Sintaxe

```
db2gse.ST_MPointFromText(multiPointTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPoint
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela MULTIPOINT_TEST com uma coluna de multiponto simples, MPT1.

```
CREATE TABLE MULTIPOINT_TEST (mpt1 db2gse.ST_MultiPoint)
```

A seguinte instrução INSERT insere um multiponto na coluna MPT1 utilizando a coluna ST_MPointFromText.

```
INSERT INTO MULTIPOINT_TEST  
VALUES (1, db2gse.ST_MPointFromText('multipoint(10.01 20.03,10.52 40.11,  
30.29 41.56,31.78 10.74)'),  
db2gse.coordref()..srid(0))
```

ST_MPointFromWKB

ST_MPointFromWKB recebe uma representação binária convencional do tipo ponto múltiplo e um identificador do sistema de referências espaciais para retornar um ponto múltiplo.

Sintaxe

```
db2gse.ST_MPointFromWKB(WKBMultiPoint Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPoint
```

Exemplos

O seguinte fragmento de código preencha a tabela SPECIES_SITINGS.

A tabela SPECIES_SITINGS foi criada com três colunas. As colunas SPECIES e GENUS identificam exclusivamente cada linha, os passos que o multiponto SITTINGS armazena as localizações dos sítios das espécies.

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                               genus varchar(32),
                               sitings db2gse.ST_MultiPoint);

/* Criar a instrução insert do SQL para preencher as espécies, gêneros e
   sítios. Os pontos de interrogação são marcadores de parâmetros que
   indicam as espécies, os gêneros e os valores do sítio que serão
   recuperados no tempo de execução.*/
strcpy (wkb_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.ST_MPointFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Ligar o valor de espécies varchar ao primeiro parâmetro. */
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, species_len, 0, &species, species_len, &pcbvalue1);
/* Ligar valor do gênero varchar ao segundo parâmetro. */
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, genus_len, 0, &name, genus_len, &pcbvalue2);

/* Ligar o formato ao terceiro parâmetro. */
pcbvalue3 = sitings_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, sitings_len, 0, sitings_wkb, sitings_len, &pcbvalue3);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);
```

ST_MPolyFromText

ST_MPolyFromText recebe uma representação de texto convencional do tipo polígono múltiplo e um identificador do sistema de referências espaciais e retorna um polígono múltiplo.

Essa função não pode receber como entrada um polígono múltiplo que contenha polígonos múltiplos com as mesmas coordenadas.

Sintaxe

```
db2gse.ST_MPolyFromText(multiPolygonTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPolygon
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela MULTIPOLYGON_TEST, que tem uma coluna de multipolígono simples, MPL1.

```
CREATE TABLE MULTIPOLYGON_TEST (mp11 db2gse.ST_MultiPolygon)
```

A seguinte instrução INSERT insere um multipolígono na coluna MPL1 usando a função ST_MPolyFromText.

```
INSERT INTO MULTIPOLYGON_TEST VALUES (  
db2gse.ST_MPolyFromText('multipolygon(((10.01 20.03,10.52 40.11,  
30.29 41.56,31.78 10.74,10.01 20.03),(21.23 15.74,21.34 35.21,28.94 35.35,  
29.02 16.83, 21.23 15.74)),((40.91 10.92,40.56 20.19,  
50.01 21.12,51.34 9.81, 40.91 10.92)))',  
db2gse.coordref()..srid(0)))
```

ST_MPolyFromWKB

ST_MPolyFromWKB recebe uma representação binária convencional do tipo polígono múltiplo e um identificador do sistema de referência espacial e retorna um polígono múltiplo.

Sintaxe

```
db2gse.ST_MPolyFromWKB(WKBMultiPolygon Blob(1M), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_MultiPolygon
```

Exemplos

O seguinte fragmento de código preenche a tabela LOTES.

A tabela LOTS armazena o LOT_ID que identifica exclusivamente cada lote e o multipolígono do LOT que contém a figura geométrica de linha do lote.

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );  
  
/* Criar a instrução insert do SQL para preencher a id do lote e o lote. Os  
pontos de interrogação são marcadores de parâmetro que indicam a id do  
lote e os valores do  
lote que serão recuperados no tempo de execução. */  
strcpy (wkb_sql,"insert into LOTS (lot_id,lot)  
values (?, db2gse.ST_MPolyFromWKB (cast(? as blob(1m)),  
db2gse.coordref().srid(0)))");  
  
/* Alocar memória para o tratamento da instrução SQL e associar o  
tratamento da instrução ao tratamento da conexão. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Preparar as instruções SQL para execução. */  
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);  
  
/* Ligar o valor inteiro do id do lote ao primeiro parâmetro.*/  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);  
  
/* Ligar o formato do lote ao segundo parâmetro. */  
pcbvalue2 = lot_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
SQL_BLOB, lot_len, 0, lot_wkb, lot_len, &pcbvalue2);  
  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

ST_NumGeometries

ST_NumGeometries toma uma coleção e retorna o número de figuras geométricas na coleção.

Sintaxe

```
db2gse.ST_NumGeometries(g db2gse.ST_GeomCollection)
```

Tipo de retorno

Inteiro

Exemplos

O engenheiro da cidade precisa saber o número real de construções do distrito associado a cada base da construção.

As bases da construção são armazenadas na tabela BUILDINGFOOTPRINTS que foi criada com a seguinte instrução CREATE TABLE.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                   lot_id      integer,
                                   footprint    db2gse.ST_MultiPolygon);
```

A seguinte instrução SELECT usa a função ST_NumGeometries para relacionar o BUILDING_ID que identifica exclusivamente cada construção e o número de construções em cada base.

```
SELECT building_id, db2gse.ST_NumGeometries (footprint) "Number of buildings"
FROM BUILDINGFOOTPRINTS;
```

ST_NumInteriorRing

ST_NumInteriorRing toma um polígono e retorna o número de seus anéis interiores.

Sintaxe

db2gse.ST_NumInteriorRing(p db2gse.ST_Polygon)

Tipo de retorno

Inteiro

Exemplos

Uma ornitologista que deseja estudar a população de pássaros em várias ilhas marinhas do sul, sabe que a zona de alimentação de uma determinada espécie restringe-se às ilhas que contém os lagos de água fresca. Portanto, ela deseja saber quais ilhas contêm um ou mais lagos.

A seguinte instrução CREATE TABLE cria a tabela ISLANDS. As colunas ID e NAME da tabela ISLANDS identificam cada ilha e a coluna polígono de LAND armazena a figura geométrica da ilha.

```
CREATE TABLE ISLANDS (id integer, name varchar(32), land db2gse.ST_Polygon);
```

Como os anéis interiores representam os lagos, a função ST_NumInteriorRing é usada para relacionar somente essas ilhas que têm pelo menos um anel interior.

```
SELECT name FROM ISLANDS WHERE db2gse.ST_NumInteriorRing(land) > 0;
```

ST_NumPoints

ST_NumPoints toma uma cadeia de linha e retorna seu número de pontos.

Sintaxe

```
db2gse.ST_NumPoints(l db2gse.ST_LineString)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela NUMPOINTS_TEST. A coluna GEOTYPE contém o tipo de figura geométrica armazenado na coluna de figura geométrica G1.

```
CREATE TABLE NUMPOINTS_TEST (geotype varchar(12), g1 db2gse.ST_Geometry)
```

A instrução INSERT abaixo insere uma cadeia de linhas.

```
INSERT INTO NUMPOINTS_TEST VALUES( linestring,  
db2gse.ST_LineFromText('linestring (10.02 20.01, 23.73 21.92)',  
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente relaciona o tipo de figura geométrica e o número de pontos contidos dentro de cada.

```
SELECT geotype, db2gse.ST_NumPoints(g1)  
FROM NUMPOINTS_TEST
```

```
GEOTYPE      Number of points  
-----  
ST_linestring      2  
1 record(s) selected.
```

ST_OrderingEquals

ST_OrderingEquals compara as duas figuras geométricas e retorna 1 (VERDADEIRO) se as figuras geométricas forem iguais e as coordenadas estiverem na mesma ordem; do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_OrderingEquals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A seguinte instrução CREATE TABLE cria a tabela LINESTRING_TEST, que tem duas colunas de cadeia de linhas, L1 e L2.

```
CREATE TABLE LINESTRING_TEST (l1 integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString);
```

A seguinte instrução INSERT insere duas cadeias de linha em L1 e L2 que são iguais e têm a mesma ordem de coordenadas.

```
INSERT INTO linestring_test VALUES (1,  
db2gse.LineFromText('linestring (10.01 20.02, 21.50 12.10)',  
db2gse.coordref()..srid(0)),  
db2gse.LineFromText('linestring (10.01 20.02, 21.50 12.10)',  
db2gse.coordref()..srid(0)));
```

A seguinte instrução INSERT insere duas cadeias de linha em L1 e L2 que são iguais mas não têm a mesma ordem de coordenadas.

```
INSERT INTO linestring_test VALUES (2,  
db2gse.LineFromText('linestring (10.01 20.02, 21.50 12.10)',  
db2gse.coordref()..srid(0)),  
db2gse.LineFromText('linestring (21.50 12.10,10.01 20.02)',  
db2gse.coordref()..srid(0)));
```

Conforme mostra a seguinte instrução SELECT e conjunto de resultados, a função ST_OrderingEquals:

- Retorna 1 (TRUE) quando as figuras geométricas que recebe como entrada são iguais e suas coordenadas estão na mesma ordem
- Retorna 0 (FALSE) quando as figuras geométricas que recebe não são iguais ou quando as coordenadas de uma das figuras geométricas segue uma ordem diferente das coordenadas da outra

```
SELECT l1, db2gse.ST_OrderingEquals(l1,l2) OrderingEquals FROM linestring_test  
l1 OrderingEquals  
--- -----  
1 1  
2 0
```

ST_Overlaps

ST_Overlaps toma dois objetos de figura geométrica e retorna 1 (VERDADEIRO) se a interseção dos objetos resultar num objeto de figura geométrica da mesma dimensão, mas não for igual a nenhum dos objetos de origem; do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Overlaps(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

O supervisor do município precisa de uma lista de áreas de lixo tóxico cujo raio de 8 km abranja áreas sensíveis.

A seguinte instrução CREATE TABLE cria a tabela SENSITIVE_AREAS. A tabela SENSITIVE_AREAS contém várias colunas que descrevem as instituições ameaçadas além da coluna ZONE, que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

A tabela HAZARDOUS_SITES armazena o identificador dos locais nas colunas SITE_ID e NAME, ao passo que a localização geográfica real de cada local é armazenada na coluna de ponto LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);
```

Na seguinte instrução SELECT, as tabelas SENSITIVES_AREAS E HAZARDOUS_SITES são unidas pela função ST_Overlaps. Ela retorna 1 (VERDADEIRO) para todas as linhas na tabela SENSITIVE_AREAS cujo polígono da zona abrange o raio de 8 km de buffer do ponto de localização HAZARDOUS_SITES.

```
SELECT hs.name
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa
WHERE db2gse.ST_Overlaps (buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

Na Figura 37 na página 293, o hospital e a escola estão no raio de 8 km das áreas de lixo tóxico do município, ao passo que a casa de repouso está fora.

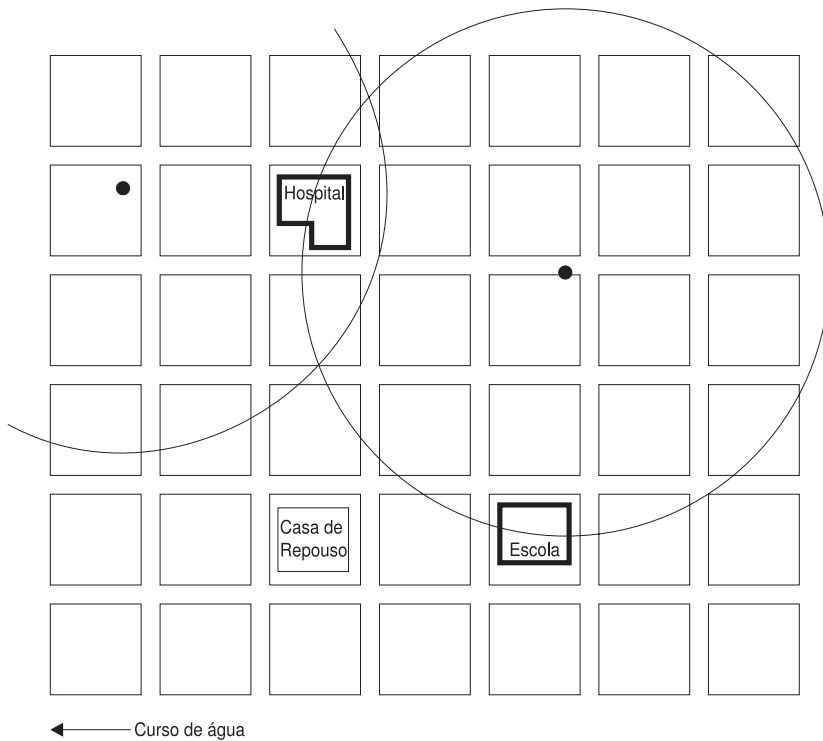


Figura 37. Usando ST_Overlaps para determinar as construções que pelo menos parcialmente estão dentro de uma área de lixo tóxico.

ST_Perimeter

ST_Perimeter retorna o perímetro de um polígono.

Sintaxe

```
db2gse.ST_Perimeter(p db2gse.ST_Polygon)
```

Tipo de retorno

Duplo

Exemplos

Um ecologista que estuda os pássaros da costa para determinar a linha costeira dos lagos dentro de uma determinada área. Os lagos estão armazenados como polígonos múltiplos na tabela WATERBODIES que foi criada com a seguinte instrução CREATE TABLE.

```
CREATE TABLE WATERBODIES (wbid integer,  
                             waterbody db2gse.ST_MultiPolygon);
```

Na seguinte instrução SELECT, a função ST_Perimeter retorna o perímetro ao redor de cada extensão de água, ao passo que a função SUM agrega os perímetros para retornar seu total.

```
SELECT SUM(db2gse.ST_Perimeter(waterbody))  
FROM waterbodies;
```

ST_Point

ST_Point retorna um ST_Point, dada uma coordenada x, coordenada y e referência espacial.

Sintaxe

```
db2gse.ST_Point(X Double, Y Double, SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela POINT_TEST, que tem uma coluna de ponto simples, PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

A função ST_Point converte as coordenadas de ponto em uma figura geométrica de ponto antes de a instrução INSERT inseri-la na coluna PT1.

```
INSERT INTO point_test VALUES(  
    db2gse.ST_Point(10.01,20.03, db2gse.coordref()..srid(0))  
)
```

ST_PointFromText

ST_PointFromText recebe uma representação de texto convencional do tipo ponto e um identificador do sistema de referências espaciais e retorna um ponto.

Sintaxe

```
db2gse.ST_PointFromText(pointTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela POINT_TEST, que tem uma coluna de ponto simples, PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

Antes de a instrução INSERT inserir o ponto na coluna PT1, a função ST_PointFromText converte as coordenadas de texto do ponto ao formato de ponto.

```
INSERT INTO POINT_TEST VALUES (  
    db2gse.ST_PointFromText ('point(10.01 20.03)',  
db2gse.coordref()..srid(0))
```

ST_PointFromWKB

ST_PointFromWKB recebe uma representação binária convencional do tipo ponto e um identificador do sistema de referências espaciais para retornar um ponto.

Sintaxe

```
db2gse.ST_PointFromWKB(WKBPoint Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

O seguinte fragmento de código preenche a tabela HAZARDOUS_SITES.

As áreas de risco são armazenadas na tabela HAZARDOUS_SITES criada com a seguinte instrução CREATE TABLE. A coluna LOCATION, definida como um ponto, armazena uma localização que é o centro geográfico de cada local arriscado.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);

/* Criar a instrução insert do SQL para preencher a id do local, nome e
   localização. Os pontos de interrogação são marcadores de parâmetros
   que indicam a
   id do local, nome e os valores de localização que serão recuperados no
   tempo de execução. */
strcpy (wkb_sql,"insert into HAZARDOUS_SITES (site_id, name, location)
values (?,?, db2gse.ST_PointFromWKB(cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);

/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Ligar o valor inteiro do id do local ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* Ligar o valor varchar do nome ao segundo parâmetro. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* Ligar o formato de localização ao terceiro parâmetro. */
pcbvalue3 = location_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, location_len, 0, location_wkb, location_len, &pcbvalue3);  
/* Executar a instrução insert . */  
rc = SQLExecute (hstmt);
```

ST_PointN

ST_PointN toma uma cadeia de linha e um índice de inteiros e retorna um ponto que é o vértice nth no caminho da cadeia de linhas.

Sintaxe

```
db2gse.ST_PointN(l db2gse.ST_Curve, n Integer)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

A seguinte tabela CREATE TABLE cria a tabela POINTN_TEST, que tem duas colunas: a coluna GID, que identifica exclusivamente cada linha, e a coluna da cadeia de linhas LN1.

```
CREATE TABLE POINTN_TEST (gid integer, ln1 db2gse.ST_LineString)
```

A seguinte instrução INSERT insere dois valores da cadeia de linhas. A primeira cadeia de linha não possui coordenadas Z ou medidas, ao passo que a segunda possui ambas.

```
INSERT INTO POINTN_TEST VALUES(1,
db2gse.ST_LineFromText('linestring (10.02 20.01,23.73 21.92,30.10 40.23)',
db2gse.coordref()..srid(0)))
INSERT INTO POINTN_TEST VALUES(2,
db2gse.ST_LineFromText('linestring zm (10.02 20.01 5.0 7.0,
23.73 21.92 6.5 7.1,
30.10 40.23 6.9 7.2)',
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente relacionam a coluna GID e o segundo vértice de cada cadeia de linha. A primeira linha resulta num ponto sem uma coordenada Z ou medida, ao passo que a segunda linha resulta num ponto com uma coordenada Z e uma medida. A função ST_PointN retorna pontos com uma coordenada Z ou uma medida, se existirem na cadeia de linhas de origem.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_PointN(ln1,2)) AS varchar(60))
"The 2nd vertice"
FROM POINTN_TEST
```

```
GID          The 2nd vertice
-----
1 POINT ( 23.73000000 21.92000000)
2 POINT ZM ( 23.73000000 21.92000000 7.00000000 7.10000000)
```

```
2 record(s) selected.
```

ST_PointOnSurface

O ST_PointOnSurface recebe tanto um polígono quanto um polígono múltiplo e retorna um ST_Point.

Sintaxe

```
db2gse.ST_PointOnSurface(s db2gse.ST_Surface)
db2gse.ST_PointOnSurface(ms db2gse.ST_MultiSurface)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

O engenheiro da cidade precisa criar um ponto de rótulo para cada base de construção.

As bases da construção são armazenadas na tabela BUILDINGFOOTPRINTS que foi criada com a seguinte instrução CREATE TABLE.

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

A função ST_PointOnSurface gera um ponto que tem garantia de estar na superfície das bases de construção.

```
SELECT db2gse.ST_PointOnSurface(footprint) FROM BUILDINGFOOTPRINTS;
```

ST_PolyFromText

ST_PolyFromText recebe uma representação de texto convencional do tipo polígono e um identificador do sistema de referências espaciais e retorna um polígono.

Sintaxe

```
db2gse.ST_PolyFromText(polygonTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Polygon
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela POLYGON_TEST com a coluna de polígono simples.

```
CREATE TABLE POLYGON_TEST (p11 db2gse.ST_Polygon)
```

A seguinte instrução INSERT insere um polígono na coluna de polígonos usando a função ST_PolyFromText.

```
INSERT INTO POLYGON_TEST VALUES (db2gse.ST_PolyFromText(  
'polygon((10.01 20.03,10.52 40.11,30.29  
41.56, 31.78 10.74,10.01 20.03))',  
  
db2gse.coordref()..srid(0)))
```

ST_PolyFromWKB

ST_PolyFromWKB recebe uma representação binária convencional do tipo polígono e um identificador do sistema de referências espaciais para retornar um polígono.

Sintaxe

```
db2gse.ST_PolyFromWKB(WKBPolygon Blob(1M), SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Polygon
```

Exemplos

O seguinte fragmento de código preenche a tabela SENSITIVE_AREAS.

A tabela SENSITIVE_AREAS contém várias colunas que descrevem as instituições ameaçadas além da coluna zona, que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name       varchar(128),
                                size       float,
                                type       varchar(10),
                                zone       db2gse.ST_Polygon);
```

```
/* Criar a instrução insert do SQL para preencher a id, nome, tamanho, tipo e
   zona. Os pontos de interrogação são marcadores de parâmetro que
   indicam a id, nome, tamanho, tipo e valores da zona que serão
   recuperados no tempo de execução. */
```

```
strcpy (shp_wkb,"insert into SENSITIVE_AREAS (id, name, size, type, zone)
values (?,?,,?, db2gse.ST_PolyFromWKB (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");
```

```
/* Alocar memória para o tratamento da instrução SQL e associar o
   tratamento da instrução ao tratamento da conexão. */
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Preparar as instruções SQL para execução. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);
```

```
/* Ligar o valor inteiro do id ao primeiro parâmetro.*/
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);
```

```
/* Ligar o valor varchar do nome ao segundo parâmetro. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);
```

```
/* Ligar a dimensão flutuante ao terceiro parâmetro. */
pcbvalue3 = 0;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
```

```

        SQL_REAL, 0, 0, &size, 0, &pcbvalue3);

/* Ligar o tipo varchar ao quarto parâmetro. */
pcbvalue4 = type_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_CHAR,
        SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);

/* Ligar o polígono de zona ao quinto parâmetro. */
pcbvalue5 = zone_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
        SQL_BLOB, zone_len, 0, zone_wkb, zone_len, &pcbvalue5);

/* Executar a instrução insert . */
rc = SQLExecute (hstmt);

```

ST_Polygon

ST_Polygon gera um ST_Polygon de um ST_LineString.

Sintaxe

```
db2gse.ST_Polygon(l db2gse.ST_LineString)
```

Tipo de retorno

```
db2gse.ST_Polygon
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela POLYGON_TEST, que tem uma coluna simples, P1.

```
CREATE TABLE POLYGON_TEST (p1 db2gse.ST_polygon)
```

A seguinte instrução INSERT converte um anel (uma cadeia de linhas que é fechada e simples) em um polígono e insere-o na coluna P1 usando a função ST_LineFromText dentro da função ST_Polygon.

```
INSERT INTO POLYGON_TEST VALUES (  
db2gse.ST_Polygon(db2gse.ST_LineFromText('linestring(10.01 20.03,20.94  
21.34,35.93 10.04,10.01 20.03)', db2gse.coordref()..srid(0)))  
)
```

ST_Relate

ST_Relate compara duas figuras geométricas e retorna 1 (VERDADEIRO) se as figuras geométricas atenderem às condições especificadas pela cadeia de matrizes do padrão DE-9IM; do contrário, 0 (FALSO) retornará. Para obter informações sobre as matrizes de padrão DE-9IM, consulte “Funções do predicado” na página 172.

Sintaxe

```
db2gse.ST_Relate(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry,  
patternMatrix CHAR(9))
```

Tipo de retorno

Inteiro

Exemplos

Uma matriz de padrão DE-9IM é um dispositivo para comparação de figuras geométricas. Há vários tipos de matrizes. Por exemplo, a matriz do padrão *é igual* informará se as duas figuras geométricas são iguais.

Neste exemplo, uma matriz de padrão igual, mostrada em Tabela 57, é lida da esquerda para a direita e de cima para baixo numa cadeia (“T*F**FFF”).

Tabela 57. Matriz de padrão igual

| | | b | | |
|----------|-----------------|-----------------|---------------|-----------------|
| | | Interior | Limite | Exterior |
| a | Interior | T | * | F |
| | Limite | * | * | F |
| | Exterior | F | F | * |

Em seguida, a tabela RELATE_TEST é criada com a seguinte instrução CREATE TABLE.

```
CREATE TABLE RELATE_TEST (rid integer, g1 db2gse.ST_Geometry,  
g2 db2gse.ST_Geometry, g3 db2gse.ST_Geometry);
```

As seguintes instruções INSERT inserem uma subclasse na tabela RELATE_TEST.

```
INSERT INTO RELATE_TEST VALUES(  
1,  
db2gse.ST_PointFromText('point (10.02 20.01)',  
db2gse.coordref()..srid(0),  
db2gse.ST_PointFromText('point (10.02 20.01)',  
db2gse.coordref()..srid(0),  
db2gse.ST_PointFromText('point (30.01 20.01)',  
db2gse.coordref()..srid(0)  
)
```

A seguinte instrução SELECT e o conjunto de resultados correspondente relaciona o nome da subclasse armazenado na coluna GEOTYPE à dimensão desse geotipo.

```
SELECT rid, relate(g1,g2, 'T**F**FFF*') equals FROM relate_test
```

```
RID      equals
-----  -----
1        1
```

1 record(s) selected.

ST_SRID

ST_SRID recebe um objeto de figura geométrica e retorna seu identificador do sistema de referências espaciais.

Sintaxe

```
db2gse.ST_SRID(g1 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

A função ST_SRID retorna o identificador do sistema de referências espaciais associado ao valor ST_Geometry.

Por exemplo, um tipo de figura geométrica é usado na instrução CREATE TABLE:

```
CREATE TABLE SRID_TEST(g1 db2gse.ST_Geometry)
```

Na seguinte instrução INSERT, é inserido uma figura geométrica do ponto localizado na coordenada 10.01,50.76 na coluna de figura geométrica G1. Quando a figura geométrica do ponto foi criada pela função ST_PointFromText, ela foi atribuída ao valor srid 1.

```
INSERT INTO SRID_TEST  
VALUES (db2gse.ST_PointFromText('point(10.01 50.76)',  
db2gse.coordref()..srid(0)))
```

A função ST_SRID retorna o identificador do sistema de referências espaciais da figura geométrica que acaba de ser fornecida, conforme ilustrado pela seguinte instrução SELECT e pelo conjunto de resultados correspondente.

```
SELECT db2gse.ST_SRID(g1) FROM SRID_TEST
```

```
g1  
-----  
0
```

ST_StartPoint

ST_StartPoint toma uma cadeia de linha e retorna um ponto que indica o primeiro ponto das cadeias de linhas.

Sintaxe

```
db2gse.ST_StartPoint(c db2gse.ST_Curve)
```

Tipo de retorno

```
db2gse.ST_Point
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela STARTPOINT_TEST. STARTPOINT_TEST tem duas colunas: a coluna de inteiros GID, que identifica exclusivamente as linhas da tabela e a coluna da cadeia de linhas LN1.

```
CREATE TABLE STARTPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

As seguintes instruções INSERT inserem as cadeias de linhas na coluna LN1. A primeira cadeia de linha não possui coordenadas Z ou medidas, ao passo que a segunda possui ambas.

```
INSERT INTO STARTPOINT_TEST VALUES(1,  
db2gse.ST_LineFromText('linestring (10.02 20.01,23.73  
21.92,30.10 40.23)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO STARTPOINT_TEST VALUES(2,  
db2gse.ST_LineFromText('linestring zm (10.02 20.01 5.0 7.0,  
23.73 21.92 6.5 7.1,30.10 40.23 6.9 7.2)',
```

```
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente mostram a função ST_StartPoint extraído o primeiro ponto de cada cadeia de linha. A função ST_AsText converte o ponto ao seu formato de texto. O primeiro ponto na lista não possui uma coordenada Z ou uma medida, enquanto o segundo ponto tem ambos, porque a cadeia de linhas de origem tinha.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_StartPoint (ln1)) as varchar(60))  
"Startpoint"  
FROM STARTPOINT_TEST
```

```
GID          Startpoint  
-----  
1 POINT ( 10.02000000 20.01000000)  
2 POINT ZM ( 10.02000000 20.01000000 5.00000000 7.00000000)
```

```
2 record(s) selected.
```

ST_SymmetricDiff

O `ST_SymmetricDiff` toma dois objetos de figura geométrica e retorna um objeto de figura geométrica que é a diferença simétrica dos objetos de origem.

A função `ST_SymmetricDiff` retorna a diferença simétrica (a lógica booleana XOR de espaço) de duas figuras geométricas que fazem interseção e estão na mesma dimensão. Se as figuras geométricas forem idênticas, `ST_SymmetricDiff` retornará uma figura geométrica vazia. Se não, uma parte de uma delas ou de ambas ficará fora da área de interseção. `ST_SymmetricDiff` retorna a parte ou as partes que não fazem interseção como uma coleção; por exemplo, como um polígono múltiplo.

Se `ST_SymmetricDiff` receber figuras geométricas de dimensões diferentes como entradas, retornará um valor nulo.

Sintaxe

```
db2gse.ST_SymmetricDiff(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

O supervisor do município deve determinar a área sensível e o raio de 8 km de áreas de risco que não têm interseção.

A seguinte instrução `CREATE TABLE` cria a tabela `SENSITIVE_AREAS`, que contém várias colunas que descrevem as instituições ameaçadas. A tabela `SENSITIVE_AREAS` também contém a coluna `ZONE`, que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name        varchar(128),
                                size        float,
                                type        varchar(10),
                                zone        db2gse.ST_Polygon);
```

A seguinte instrução `CREATE TABLE` cria a tabela `HAZARDOUS_SITES`, que armazena o identificador dos locais nas colunas `SITE_ID` e `NAME`, ao passo que a localização geográfica real de cada local é armazenada na coluna de ponto `LOCATION`.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                                name      varchar(128),
                                location  point);
```

A função `ST_Buffer` gera um buffer de 8 km ao redor das áreas do lixo tóxico. A função `ST_SymmetricDiff` gera polígonos a partir da interseção dos polígonos de locais com lixo tóxico em buffer e das áreas sensíveis. A função `ST_Area` retorna a área de polígono da interseção, de cada área de risco.

```

SELECT sa.name, hs.name,
       db2gse.ST_Area(db2gse.ST_SymmetricDiff (db2gse.ST_Buffer(hs.location,
(5 * 5280)),sa.zone))
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa

```

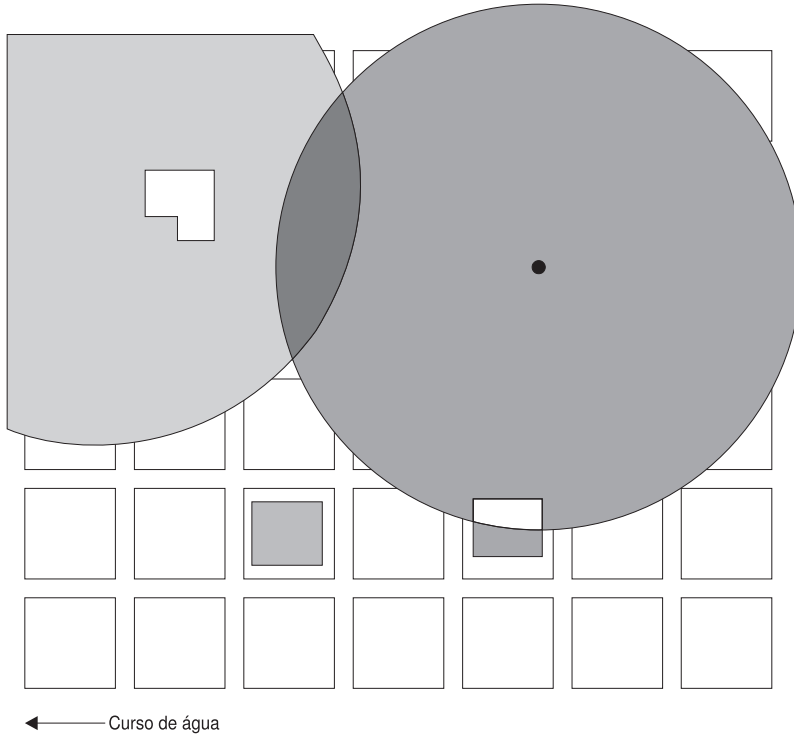


Figura 38. Utilizando o `ST_SymmetricDiff` para determinar as áreas de lixo tóxico que não contêm áreas sensíveis (construções desabitadas)

Na Figura 38, a diferença simétrica das áreas de lixo tóxico e das área sensíveis resulta na subtração das áreas de interseção.

ST_Touches

ST_Touches retorna 1 (VERDADEIRO) se nenhum dos pontos comuns a ambas as figuras geométricas fizerem interseção com os interiores das duas; do contrário, retornará 0 (FALSO). Pelo menos uma figura geométrica deve ser uma cadeia de linhas, polígono, cadeia de linhas múltiplas ou multipolígono.

Sintaxe

```
db2gse.ST_Touches(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

O técnico GIS precisa fornecer uma lista de todas as linhas de sewer cujas extremidades façam interseção com outra linha de sewer.

A seguinte instrução CREATE TABLE cria a tabela SEWERLINES, a qual possui três colunas. A primeira coluna, SEWER_ID, identifica exclusivamente cada linha do sewer. A segunda coluna, CLASS, do tipo inteiro identifica o tipo de linha de sewer, que geralmente está associada à capacidade da linha. A terceira coluna, SEWER, do tipo cadeia de linhas armazena a figura geométrica da linha sewer.

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer, sewer  
db2gse.ST_LineString);
```

A seguinte instrução SELECT retorna uma lista ordenada de SEWER_IDS que se encontram.

```
SELECT s1.sewer_id, s2.sewer_id  
FROM sewerlines s1, sewerlines s2  
WHERE db2gse.ST_Touches (s1.sewer, s2.sewer) = 1,  
ORDER BY 1,2;
```

ST_Transform

ST_Transform associa uma figura geométrica a um sistema de referências espaciais diferente do sistema de referências espaciais ao qual a figura geométrica está associada atualmente.

Sintaxe

```
db2gse.ST_Transform(g db2gse.ST_Geometry, SRID db2gse.coordref)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela TRANSFORM_TEST, que tem duas colunas de cadeia de linhas, L1 e L2.

```
CREATE TABLE TRANSFORM_TEST (tid integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString)
```

A seguinte instrução INSERT insere uma cadeia de linhas em l1 com SRID de 102.

```
INSERT INTO TRANSFORM_TEST VALUES (1, db2gse.ST_LineFromText('linestring  
(10.01 40.43, 92.32 29.89)',  
db2gse.coordref()..srid(102)),NULL)
```

A função ST_Transform converte a cadeia de linhas de L1 da referência de coordenada atribuída para SRID 102 à referência de coordenadas atribuída para SRID 105. A seguinte instrução UPDATE armazena a cadeia de linhas transformada na coluna l2.

```
UPDATE TRANSFORM_TEST SET l2 = db2gse.ST_Transform(l1,  
db2gse.coordref()..srid(105))
```

Se a figura geométrica atribuída ficar fora do sistema de coordenadas que serve de base ao novo sistema de referências espaciais, ST_Transform irá retornar a figura geométrica como um valor nulo.

Por exemplo, considere uma figura geométrica ST_Point com 10.01 como coordenada X e 20.02 como coordenada Y. Suponha que essa figura geométrica seja atribuída a um sistema de referências espaciais com os seguintes parâmetros:

| | |
|----------------|---|
| falsex | 0 |
| falsey | 0 |
| xyunits | 1 |

Em seguida, suponha que você invoque ST_Transform para alterar o sistema de referências espaciais da figura geométrica ST_Point para um dos seguintes parâmetros:

| | |
|----------------------|-----|
| <code>falsex</code> | 100 |
| <code>falsey</code> | 100 |
| <code>xyunits</code> | 1 |

`ST_Transform` retornará a figura geométrica como um nulo, porque as coordenadas (10.01, 20.02) ficam fora do sistema de coordenadas que serve de base ao novo sistema de referências espaciais.

ST_Union

ST_Union soma dois objetos de figura geométrica e retorna um objeto de figura geométrica que é a união dos objetos de origem.

Sintaxe

```
db2gse.ST_Union(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A seguinte instrução CREATE TABLE cria a tabela SENSITIVE_AREAS, que contém várias colunas que descrevem as instituições ameaçadas. A tabela SENSITIVE_AREAS também contém a coluna ZONE, que armazena a figura geométrica do polígono da instituição.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

A seguinte instrução CREATE TABLE cria a tabela HAZARDOUS_SITES que armazena o identificador dos locais nas colunas SITE_ID e NAME. A localização geográfica real de cada área está armazenada na coluna de ponto LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer, name varchar(128),
                               location db2gse.ST_Point);
```

A seguinte instrução SELECT usa a função ST_Buffer para gerar um buffer de 8 km ao redor das áreas de lixo tóxico. A função ST_Union gera polígonos a partir da união dos polígonos de locais de lixo tóxico em buffer e das áreas sensíveis. A função ST_Area retorna a união da área do polígono.

```
SELECT sa.name, hs.name,
       db2gse.ST_Area(db2gse.ST_Union(db2gse.ST_Buffer(hs.location,
(5 * 5280)),sa.zone))
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa;
```

ST_Within

ST_Within toma dois objetos de figura geométrica e retorna 1 (VERDADEIRO) se o primeiro objeto estiver completamente dentro do segundo; do contrário, retornará 0 (FALSO).

Sintaxe

```
db2gse.ST_Within(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo de retorno

Inteiro

Exemplos

No exemplo abaixo são criadas duas tabelas. A primeira, BUILDINGFOOTPRINTS, contém as bases de construção da cidade. A segunda tabela, LOTS, contém os lotes da cidade. O engenheiro da cidade deseja verificar se todas as bases de construção estão inteiramente dentro de seus lotes.

Nas duas tabelas, o tipo de dados de multipolígono armazena a figura geométrica das bases e lotes de construção. O projetista selecionou os multipolígonos para os dois recursos, porque os lotes podem ser desmembrados por recursos naturais, como um rio e que as bases de construção podem ser geralmente constituídas de vários edifícios.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                   lot_id      integer,
                                   footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (  lot_id integer, lot db2gse.ST_MultiPolygon );
```

Utilizando a seguinte instrução SELECT, o engenheiro da cidade primeiro seleciona as construções que não estão completamente dentro de um lote.

```
SELECT building_id
   FROM BUILDINGFOOTPRINTS, LOTS
  WHERE db2gse.ST_Within(footprint,lot) ≠ 1;
```

Embora a primeira consulta forneça uma lista de todos os BUILDINGS_IDS que têm bases foram de um polígono do lote, ela não determinará se o restante possui a id de lote atribuído a eles. Esta segunda consulta executa uma verificação de integridade dos dados na coluna LOT_ID da tabela BUILDINGFOOTPRINTS.

```
SELECT bf.building_id "building id",
       bf.lot_id      "buildings lot_id",
       LOTS.lot_id    "LOTS lot_id"
  FROM BUILDINGFOOTPRINTS bf, LOTS
  WHERE db2gse.ST_Within(footprint,lot) = 1 AND
        LOTS.lot_id <> bf.lot_id;
```

ST_WKBTToSQL

ST_WKBTToSQL constrói um valor de ST_Geometry dada a sua representação do modo binário reconhecida. Um valor 0 de SRID é usado automaticamente.

Sintaxe

```
db2gse.ST_WKBTToSQL(WKBGeometry Blob(1M))
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A instrução CREATE TABLE a seguir cria a tabela LOTS, que possui duas colunas: a coluna LOT_ID, que identifica exclusivamente cada lote, e a coluna de multipolígono LOT, que contém a figura geométrica de cada lote.

```
CREATE TABLE lots (lot_id integer,  
                   lot      db2gse.ST_MultiPolygon);
```

O seguinte fragmento de código C contém funções ODBC incorporadas às funções SQL do Spatial Extender que inserem dados na tabela LOTS.

A função ST_WKBTToSQL converte representações de WKB em figura geométrica do Spatial Extender. A instrução INSERT inteira é copiada em uma cadeia wkb_sql char. A instrução INSERT contém marcadores do parâmetro para aceitar os dados do ID_do LOT e os dados do LOTE dinamicamente.

```
/* Criar a instrução insert do SQL para preencher o id do lote e o  
   multipolígono do lote. Os pontos de interrogação são marcadores  
   do parâmetro que  
   indicam a id do lote e os valores do lote  
   que serão recuperados no tempo de execução.*/
```

```
strcpy (wkb_sql,"insert into lots (lot_id, lot)  
        values(?, db2gse.ST_WKBTToSQL(cast(? as blob(1m))))");
```

```
/* Alocar memória para o tratamento da instrução SQL e associar o  
   tratamento da instrução ao tratamento da conexão. */
```

```
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Preparar as instruções SQL para execução. */
```

```
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);
```

```
/* Ligar o valor inteiro da chave ao primeiro parâmetro.*/
```

```
pcbvalue1 = 0;
```

```
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
                      SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Ligar o formato ao segundo parâmetro. */
```

```
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);

/* Executar a instrução insert . */

rc = SQLExecute (hstmt);
```

ST_WKTTToSQL

ST_WKTTToSQL constrói um valor de ST_Geometry dada a sua representação textual reconhecida. Um valor 0 de SRID é usado automaticamente.

Sintaxe

```
db2gse.ST_WKTTToSQL(geometryTaggedText Varchar(4000))
```

Tipo de retorno

```
db2gse.ST_Geometry
```

Exemplos

A seguinte tabela CREATE TABLE cria a tabela GEOMETRY_TEST, que contém duas colunas: a coluna GID do tipo inteiro, que identifica exclusivamente cada linha, e a coluna G1, que armazena a figura geométrica.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

As seguintes instruções INSERT inserem os dados nas colunas GID e G1 da tabela GEOMETRY_TEST. A função ST_WKTTToSQL converte a representação de texto de cada figura geométrica na subclasse de instâncias do Spatial Extender correspondente.

```
INSERT INTO GEOMETRY_TEST VALUES(
1, db2gse.ST_WKTTToSQL ('point (10.02 20.01)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
2, db2gse.ST_WKTTToSQL('linestring (10.01 20.01, 10.01 30.01, 10.01 40.01)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
3, db2gse.ST_WKTTToSQL('polygon ((10.02 20.01, 11.92 35.64, 25.02 34.15,
19.15 33.94, 10.02 20.01))')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
4, db2gse.ST_WKTTToSQL('multipoint (10.02 20.01,10.32 23.98,11.92 25.64)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
5, db2gse.ST_WKTTToSQL('multilinestring ((10.02 20.01,      10.32 23.98,
11.92 25.64),(9.55 23.75,15.36 30.11))')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
6, db2gse.ST_WKTTToSQL('multipolygon (((10.02 20.01, 11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
((51.71 21.73, 73.36 27.04, 71.52 32.87, 52.43 31.90, 51.71 21.73))))')
)
```

ST_X

ST_X toma um ponto e retorna sua coordenada X.

Sintaxe

```
ST_X(p ST_Point)
```

Tipo de retorno

Duplo

Exemplos

A seguinte tabela CREATE TABLE cria a tabela X_TEST, que tem duas colunas: a coluna GID, que identifica exclusivamente a linha, e a coluna de ponto PT1.

```
CREATE TABLE X_TEST (gid integer, pt1 db2gse.ST_Point)
```

A seguinte instrução INSERT insere duas linhas. Uma é um ponto sem uma coordenada Z ou uma medida. A outra coluna possui uma coordenada Z e uma medida.

```
INSERT INTO X_TEST VALUES(1,  
db2gse.ST_PointFromText('point (10.02 20.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO X_TEST VALUES(2,  
db2gse.ST_PointFromText('point zm (10.02 20.01 5.0 7.0)',  
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondentes relacionam a coluna GID e a coordenada Double X dos pontos.

```
SELECT gid, db2gse.ST_X(pt1) "The X coordinate" FROM X_TEST
```

```
GID          The X coordinate  
-----  
1          +1.002000000000000E+001  
2          +1.002000000000000E+001
```

2 record(s) selected.

ST_Y

ST_Y toma um ponto e retorna sua coordenada Y.

Sintaxe

```
db2gse.ST_Y(p db2gse.ST_Point)
```

Tipo de retorno

Duplo

Exemplos

A seguinte tabela CREATE TABLE cria a tabela Y_TEST, que tem duas colunas: a coluna GID, que identifica exclusivamente a linha, e a coluna de ponto PT1.

```
CREATE TABLE Y_TEST (gid integer, pt1 db2gse.ST_Point)
```

A instrução INSERT insere duas linhas. Uma é um ponto sem uma coordenada Z ou uma medida. A outra coluna possui uma coordenada Z e uma medida.

```
INSERT INTO Y_TEST VALUES(1,  
db2gse.ST_PointFromText('point (10.02 20.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO Y_TEST VALUES(2,  
db2gse.ST_PointFromText('point zm (10.02 20.01 5.0 7.0)',  
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente relacionam a coluna GID e a coordenada Double Y dos pontos.

```
SELECT gid, db2gse.ST_Y(pt1) "The Y coordinate" FROM Y_TEST
```

```
GID          The Y coordinate  
-----  
1          +2.001000000000000E+001  
2          +2.001000000000000E+001
```

2 record(s) selected.

Z

Z toma um ponto e retorna sua coordenada Z.

Sintaxe

```
Z(p db2gse.ST_Point)
```

Tipo de retorno

Duplo

Exemplos

A seguinte tabela CREATE TABLE cria a tabela Z_TEST, que tem duas colunas: a coluna GID, que identifica exclusivamente a linha, e a coluna de ponto PT1.

```
CREATE TABLE Z_TEST (gid integer, pt1 db2gse.ST_Point)
```

A seguinte instrução INSERT insere duas linhas. Uma é um ponto sem uma coordenada Z ou uma medida. A outra coluna possui uma coordenada Z e uma medida.

```
INSERT INTO Z_TEST VALUES(1,  
db2gse.ST_PointFromText('point (10.02 20.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO Z_TEST VALUES(2,  
db2gse.ST_PointFromText('point zm (10.02 20.01 5.0 7.0)',  
db2gse.coordref()..srid(0)))
```

A seguinte instrução SELECT e o conjunto de resultados correspondente relacionam a coluna GID e a coordenada Double Z dos pontos. A primeira linha é NULA porque o ponto não tem uma coordenada Z.

```
SELECT gid, db2gse.Z(pt1) "The Z coordinate" FROM Z_TEST
```

```
GID          The Z coordinate  
-----  
1            -  
2    +5.000000000000000E+000
```

2 record(s) selected.

Capítulo 15. Sistemas de coordenadas

Este capítulo fornece informações de referência sobre o sistema de referências espaciais (SRS) e os valores de coordenadas utilizados para interpretar dados espaciais.

- “Visão geral dos sistemas de coordenadas”
- “Unidades lineares suportadas” na página 325
- “Unidades angulares suportadas” na página 326
- “Esferóides suportados” na página 326
- “Dados geodéticos suportados” na página 328
- “Meridianos principais suportados” na página 330
- “Projeções do mapa suportadas” na página 330
- “Projeções cônicas suportadas” na página 331
- “Projeções azimutais ou do circuito impresso suportadas” na página 331
- “Parâmetros de projeção do mapa suportados” na página 332

Visão geral dos sistemas de coordenadas

A representação de texto reconhecida dos sistemas de referência espacial fornece uma representação textual padrão para informações sobre o sistema de referência. As definições da representação de texto convencionais são modeladas conforme o modelo de dados do sistema de coordenadas do Petrotechnical Open Software Corporation/European Professional Surveyors Group (POSC/EPSC).

Um sistema de referência espacial é uma (latitude-longitude) geográfica, um (X,Y) projetado ou um sistema de coordenadas (X,Y,Z) geocêntrico. O sistema de coordenadas é composto de vários objetos. Cada objeto tem uma palavra-chave em maiúsculas (por exemplo, DATUM ou UNIT) seguido dos parâmetros de definição, delimitados por vírgulas, do objeto entre colchetes. Alguns objetos são compostos de outros objetos, portanto, o resultado é uma estrutura aninhada.

Nota: As implementações estão livres para substituir os colchetes padrão () por chaves [] e estar preparadas para ler os dois formatos.

A definição EBNF (Extended Backus Naur Form) para a representação de cadeia de um sistema de coordenadas que usa colchetes é como segue (consulte a nota acima sobre o uso de colchetes):

```

<sistema de coordenadas> = <cs projetado> | <cs geográfico> | <cs geocêntrico>
<cs projetado> = PROJCS["<nome>", <cs geográfico>, <projeção>, {<parâmetro>,*
    <unidade linear>}]
<projeção> = PROJECTION["<nome>"]
<parâmetro> = PARAMETER["<nome>", <valor>]
<valor> = <número>

```

Um sistema de coordenadas do arquivo será identificado pela palavra-chave PROJCS se os dados estiverem em coordenadas projetadas (por GEOGCS se em coordenadas geográficas ou por GEOCCS se em coordenadas geocêntricas). A palavra-chave PROJCS é seguida de todas as partes" que definem o sistema de coordenadas projetadas. A primeira parte de qualquer objeto é sempre o nome. Vários objetos seguem o nome do sistema de coordenadas projetadas: o sistema de coordenadas geográficas, a projeção de mapas, um ou mais parâmetros e a unidade linear de medida. Todos os sistemas de coordenadas projetadas são baseados num sistema de coordenadas geográficas, portanto, esta sessão descreve primeiro as partes específicas de um sistema de coordenadas projetadas. Por exemplo, a zona UTM 10N nos dados NAD83 está definida:

```

PROJCS["NAD_1983_UTM_Zone_10N",
<cs geográfico>,
PROJECTION["Transverse_Mercator"],
PARAMETER["Falso_Leste".500000,0],
PARAMETER["Falso_Norte".0,0],
PARAMETER["Central_Meridiano",-123,0],
PARAMETER["Escala_Fator".0,9996],
PARAMETER["Latitude_de_Origem".0,0],
UNIT["Metro",1.0]]

```

O nome e vários objetos definem o objeto do sistema de coordenadas geográficas em turnos: o dado, o meridiano principal e a unidade angular de medida.

```

<cs geográfico> = GEOGCS["<nome>", <dado>, <meridiano principal>, <unidade
    angular>]
<dado> = DATUM["<nome>", <esferóide>]
<esferóide> = SPHEROID["<nome>", <eixo semi-principal>, <condensação inversa>]
<eixo semi-principal> = <número>
    (o eixo semiprincipal é medido em metros e deve ser > 0.)
<condensação inversa> = <número>
<meridiano principal> = PRIMEM["<nome>", <longitude>]
<longitude> = <número>

```

A cadeia do sistema de coordenadas geográficas para a zona UTM 10 em NAD83:

```

GEOGCS["GCS_Norte_americano_1983",
DATUM["D_Norte_americano_1983",
SPHEROID["GRS_1980".6378137.298,257222101]],
PRIMEM["Greenwich",0],
UNIT["Grau".0,0174532925199433]]

```

O objeto UNIT pode representar unidade angular ou linear de medidas:

```
<unidade angular> = <unidade>
<unidade linear> = <unidade>
<unidade> = UNIT["<nome>", <fator de conversão>]
<fator de conversão> = <número>
```

O fator de conversão especifica o número de metros (para uma unidade linear) ou o número de radianos (para uma unidade angular) por unidade e deve ser maior que zero.

Assim, a representação completa da cadeia da Zona UTM 10N é a seguinte:

```
PROJCS["NAD_1983_UTM_Zone_10N",
GEOGCS["GCS_Norte_americano_1983",
DATUM["D_Norte_americano_1983",SPHEROID["GRS_1980",6378137,298.257222101]],
PRIMEM["Greenwich",0],UNIT["Grau",0.0174532925199433]],
PROJECTION["Transverse_Mercator"],PARAMETER["Falso_Leste",500000.0],
PARAMETER["Falso_Norte",0.0],PARAMETER["Meridiano_Central",-123.0],
PARAMETER["Escala_Fator",0.9996],PARAMETER["Latitude_de_Origem",0.0],
UNIT["Metro",1.0]]
```

Um sistema de coordenadas geométricas é semelhante a um sistema de coordenadas geográficas:

```
<cs geocêntrico> = GEOCCS["<nome>", <dado>, <meridiano principal>,
<unidade linear>]
```

Unidades lineares suportadas

Tabela 58. Unidades lineares suportadas

| Unidade | Fator de conversão |
|-------------------------|--------------------|
| Metro | 1,0 |
| Pé (Internacional) | 0,3048 |
| Pé americano | 12/39,37 |
| Pé americano modificado | 12,0004584/39,37 |
| Pé de Clarke | 12/39,370432 |
| Pé indiano | 12/39,370141 |
| Ligação | 7.92/39.370432 |
| Ligação (Benoit) | 7.92/39.370113 |
| Ligação (Sears) | 7.92/39.370147 |
| Cadeia (Benoit) | 792/39.370113 |
| Cadeia (Sears) | 792/39.370147 |
| Jarda (Indian) | 36/39.370141 |

Tabela 58. Unidades lineares suportadas (continuação)

| Unidade | Fator de conversão |
|---------------|--------------------|
| Jarda (Sears) | 36/39.370147 |
| Braça | 1.8288 |
| Milha náutica | 1852.0 |

Unidades angulares suportadas

Tabela 59. Unidades angulares suportadas

| Unidade | Fator de conversão |
|-----------------|--------------------|
| Radiano | 1,0 |
| Grau Decimal | $\pi/180$ |
| Minuto Decimal | $(\pi/180)/60$ |
| Segundo Decimal | $(\pi/180)/36000$ |
| Gon | $\pi/200$ |
| Grade | $\pi/200$ |

Esferóides suportados

Tabela 60. Esferóides suportados

| Nome | Eixo semi-principal | Condensação inversa |
|------------------------|---------------------|---------------------|
| Airy | 6377563.396 | 299.3249646 |
| Airy modificado | 6377340.189 | 299.3249646 |
| Australiano | 6378160 | 298.25 |
| Bessel | 6377397.155 | 299.1528128 |
| Bessel modificado | 6377492.018 | 299.1528128 |
| Bessel (Namíbia) | 6377483.865 | 299.1528128 |
| Clarke 1866 | 6378206.4 | 294.9786982 |
| Clarke 1866 (Michigan) | 6378693.704 | 294.978684677 |
| Clarke 1880 | 6378249.145 | 293.465 |
| Clarke 1880 (Arc) | 6378249.145 | 293.466307656 |
| Clarke 1880 (Benoit) | 6378300.79 | 293.466234571 |
| Clarke 1880 (IGN) | 6378249.2 | 293.46602 |
| Clarke 1880 (RGS) | 6378249.145 | 293.465 |

Tabela 60. Esferóides suportados (continuação)

| Nome | Eixo semi-principal | Condensação inversa |
|---------------------------|----------------------------|----------------------------|
| Clarke 1880 (SGA) | 6378249.2 | 293.46598 |
| Everest 1830 | 6377276.345 | 300.8017 |
| Everest 1975 | 6377301.243 | 300.8017 |
| Everest (Sarawak e Sabah) | 6377298.556 | 300.8017 |
| Everest modificado 1948 | 6377304.063 | 300.8017 |
| Fischer 1960 | 6378166 | 298.3 |
| Fischer 1968 | 6378150 | 298.3 |
| Fischer modificado (1960) | 6378155 | 298.3 |
| GEM10C | 6378137 | 298.257222101 |
| GRS 1980 | 6378137 | 298.257222101 |
| Hayford 1909 | 6378388 | 297.0 |
| Helmert 1906 | 6378200 | 298.3 |
| Hough | 6378270 | 297.0 |
| Internacional 1909 | 6378388 | 297.0 |
| Internacional 1924 | 6378388 | 297.0 |
| Novo Internacional 1967 | 6378157.5 | 298.2496 |
| Krasovsky | 6378245 | 298.3 |
| Mercury 1960 | 6378166 | 298.3 |
| Mercury modificado 1968 | 6378150 | 298.3 |
| NWL9D | 6378145 | 298.25 |
| OSU_86F | 6378136.2 | 298.25722 |
| OSU_91A | 6378136.3 | 298.25722 |
| Plessis 1817 | 6376523 | 308.64 |
| América do Sul 1969 | 6378160 | 298.25 |
| Ásia do Sul | 6378155 | 298.3 |
| Esfera (raio = 1.0) | 1 | 0 |
| Esfera (raio = 6371000 m) | 6371000 | 0 |
| Esfera (raio = 6370997 m) | 6370997 | 0 |
| Struve 1860 | 6378297 | 294.73 |
| Walbeck | 6376896 | 302.78 |
| War Office | 6378300.583 | 296 |
| WGS 1960 | 6378165 | 298.3 |

Tabela 60. Esferóides suportados (continuação)

| Nome | Eixo semi-principal | Condensação inversa |
|----------|---------------------|---------------------|
| WGS 1966 | 6378145 | 298.25 |
| WGS 1972 | 6378135 | 298.26 |
| WGS 1984 | 6378137 | 298.257223563 |

Dados geodéticos suportados

Tabela 61. Dados geodéticos suportados

| Dados geodéticos suportados | |
|------------------------------------|---------------------------------|
| Adindan | Lisboa |
| Afgooye | Loma Quintana |
| Agadez | Lome |
| Dados geodéticos australianos 1966 | Luzon 1911 |
| Dados geodéticos australianos 1984 | Mahe 1971 |
| Ain el Abd 1970 | Makassar |
| Amersfoort | Malongo 1987 |
| Aratu | Manoca |
| Arc 1950 | Massawa |
| Arc 1960 | Merchich |
| Ancienne Triangulation Francaise | Militar-Geographische Institute |
| Barbados | Mhast |
| Batavia | Minna |
| Beduaram | Monte Mario |
| Beijing 1954 | M'poraloko |
| Reseau National Belge 1950 | NAD Michigan |
| Reseau National Belge 1972 | North American Datum 1927 |
| Bermuda 1957 | North American Datum 1983 |
| Bern 1898 | Nahrwan 1967 |
| Bern 1938 | Naparima 1972 |
| Bogota | Nord de Guerre |
| Bukit Rimpah | NGO 1948 |
| Camacupa | Nord Sahara 1959 |
| Campo Inchauspe | NSWC 9Z-2 |

Tabela 61. Dados geodéticos suportados (continuação)

| Dados geodéticos suportados | |
|------------------------------------|---------------------------------------|
| Cape | Nouvelle Triangulation Francaise |
| Carthage | New Zealand Geodetic Datum 1949 |
| Chua | OS (SN) 1980 |
| Conakry 1905 | OSGB 1936 |
| Corrego Alegre | OSGB 1970 (SN) |
| Cote d'Ivoire | Padang 1884 |
| Datum 73 | Palestina 1923 |
| Deir ez Zor | Pointe Noire |
| Deutsche Hauptdreiecksnetz | Provisional South American Datum 1956 |
| Douala | Pulkovo 1942 |
| European Datum 1950 | Qatar |
| European Datum 1987 | Qatar 1948 |
| Egito 1907 | Qornoq |
| European Reference System 1989 | RT38 |
| Fahud | South American Datum 1969 |
| Gandajika 1970 | Sapper Hill 1943 |
| Garoua | Schwarzeck |
| Geocentric Datum of Australia 1994 | Segora |
| Guiana Francesa | Serindung |
| Herat North | Estocolmo 1938 |
| Hito XVIII 1963 | Sudão |
| Hu Tzu | Shan Tananarive 1925 |
| Hungarian Datum 1972 | Timbalai 1948 |
| Indian 1954 | TM65 |
| Indian 1975 | TM75 |
| Indonesian Datum 1974 | Tóquio |
| Jamaica 1875 | Trinidad 1903 |
| Jamaica 1969 | Trucial Coast 1948 |
| Kalianpur | Voirol 1875 |
| Kandawala | Voirol Unifie 1960 |
| Kertau | WGS 1972 |
| Kuwait Oil Company | WGS 1972 Transit Broadcast Ephemeris |

Tabela 61. Dados geodéticos suportados (continuação)

| Dados geodéticos suportados | |
|-----------------------------|----------|
| La Canoa | WGS 1984 |
| Lake | Yacare |
| Leigon | Yoff |
| Libéria 1964 | Zanderij |

Meridianos principais suportados

Tabela 62. Meridianos principais suportados

| Localização | Coordenadas |
|-------------|-------------------|
| Greenwich | 0° 0' 0" |
| Bern | 7° 26' 22.5" E |
| Bogota | 74° 4' 51.3" W |
| Bruxelas | 4° 22' 4.71" E |
| Ferro | 17° 40' 0" W |
| Jakarta | 106° 48' 27.79" E |
| Lisboa | 9° 7' 54.862" W |
| Madri | 3° 41' 16.58" W |
| Paris | 2° 20' 14.025" E |
| Roma | 12° 27' 8.4" E |
| Estocolmo | 18° 3' 29" E |

Projeções do mapa suportadas

Tabela 63. Projeções do mapa suportadas

| Projeções cilíndricas | Projeções pseudocilíndricas |
|------------------------|-----------------------------|
| Behrmann | Craster parabolic |
| Cassini | Eckert I |
| Área igual cilíndrica | Eckert II |
| Equiretangular | Eckert III |
| Estereográfico de Gall | Eckert IV |
| Gauss-Kruger | Eckert V |
| Mercator | Eckert VI |

Tabela 63. Projeções do mapa suportadas (continuação)

| Projeções cilíndricas | Projeções pseudocilíndricas |
|-----------------------|-----------------------------------|
| Miller cilíndrico | McBryde-Thomas flat polar quartic |
| Oblíquo | Mercator (Hotine) Mollweide |
| Plate-Carée | Robinson |
| Times | Sinusoidal (Sansom-Flamsteed) |
| Transverse Mercator | Winkel I |

Projeções cônicas suportadas

Tabela 64. Projeções cônicas suportadas

| Projeções cônicas suportadas | |
|----------------------------------|--|
| Área igual cônica de Albers | Chamberlin trimétrico |
| Cônico conformal oblíquo bipolar | Dois pontos equidistantes |
| Bonne | Área igual Hammer-Aitoff |
| Cônico equidistante | Van der Grinten I |
| Cônico conformal de Lambert | Diversos |
| Policônico | Alaska série E |
| Cônico simples | Grade Alaska (Esterográfico modificado por Snyder) |

Projeções azimutais ou do circuito impresso suportadas

- Azimutal equidistante
- Perspectiva geral do lado direito vertical
- Gnomônico
- Área igual de Lambert Azimutal
- Ortográfico
- Polar-Esterográfico
- Esterográfico

Parâmetros de projeção do mapa suportados

Tabela 65. Parâmetros de projeção do mapa suportados

| Parâmetro | Descrição |
|----------------------|--|
| meridiano_central | A linha da longitude escolhida como a origem das coordenadas x. |
| fator_de_escala | Utilizado geralmente para reduzir a quantidade de distorção na projeção de um mapa. |
| padrão_paralelo_1 | Uma linha de latitude que geralmente não apresenta distorção. Usada também para "latitude de escala verdadeira." |
| padrão_paralelo_2 | Uma linha de longitude que geralmente não apresenta distorção. |
| longitude_do_centro | A longitude que define o ponto central da projeção do mapa. |
| latitude_do_centro | A latitude que define o ponto central da projeção do mapa. |
| longitude_da_origem | A longitude escolhida como a origem das coordenadas x. |
| latitude_da_origem | A latitude escolhida como a origem das coordenadas y. |
| leste_falso | Incluído em coordenadas x. Usado para fornecer valores positivos. |
| norte_falso | Incluído em coordenadas y. Usado para fornecer valores positivos. |
| azimute | O ângulo leste do norte que define a linha central de uma projeção oblíqua. |
| longitude_de_ponto_1 | A longitude do primeiro ponto necessário para uma projeção de mapa. |
| latitude_de_ponto_1 | A latitude do primeiro ponto necessário para uma projeção de mapa. |
| longitude_de_ponto_2 | A longitude do segundo ponto necessário para uma projeção de mapa. |
| latitude_de_ponto_2 | A latitude do segundo ponto necessário para uma projeção de mapa. |
| longitude_de_ponto_3 | A longitude do terceiro ponto necessário para uma projeção de mapa. |
| latitude_de_ponto_3 | A latitude do terceiro ponto necessário para uma projeção de mapa. |
| número_de_landsat | O número de um satélite Landsat. |

Tabela 65. Parâmetros de projeção do mapa suportados (continuação)

| Parâmetro | Descrição |
|------------------------------|--|
| número_do caminho | O número de caminho orbital de um determinado satélite. |
| peso_de_ponto de_perspectiva | O peso acima da terra do ponto de perspectiva da projeção do mapa. |
| fipszone | Número de zona do Sistema de Coordenadas Planas do Estado. |
| zona | Número de zona UTM. |

Capítulo 16. Formatos de arquivos para dados espaciais

Este capítulo documenta as representações reconhecidas do Spatial Extender. As representações estão descritas como *convencionais* pois foram definidas pelo Open GIS Consortium (OGC) e não são específicas do Spatial Extender. Três tipos de valores espaciais são importantes para entender a importação e exportação de dados espaciais:

- As representações de texto convencionais (WKT) de OGC
- As representações (WKB) do modo binário reconhecidas de OGC
- As representações de formato ESRI

As representações de texto reconhecidas de OGC

O Spatial Extender possui várias funções que geram figuras geométricas a partir de descrições de textos:

ST_GeomFromText

Cria uma figura geométrica a partir da representação de texto de qualquer tipo de figura geométrica.

ST_PointFromText

Cria um ponto a partir de uma representação de texto de ponto.

ST_LineFromText

Cria uma cadeia de linhas a partir de uma representação de texto da cadeia de linhas.

ST_PolyFromText

Cria um polígono a partir de uma representação do texto do polígono.

ST_MPointFromText

Cria um multiponto a partir de representações de textos multipontos.

ST_MLineFromText

Cria uma cadeia de linhas múltiplas a partir de uma representação de texto de cadeia de linhas múltiplas.

ST_MPolyFromText

Cria um multipolígono a partir de uma representação de texto de multipolígono.

A representação de texto é uma cadeia de formato de texto ASCII que permite que a figura geométrica seja trocada no formato de texto ASCII. Você pode usar estas funções num programa (3GL ou 4GL) de linguagem de terceira ou

quarta geração mas elas não exigem as definições de estruturas de programas especiais. A função ST_AsText converte uma figura geométrica existente numa representação de texto.

Cada tipo de figura geométrica possui uma representação de texto reconhecida, que pode ser usada para construir novas instâncias do tipo e converter instâncias existentes na forma textual para exibição alfanumérica.

A representação de texto reconhecida de uma figura geométrica é definida da seguinte forma: a notação {}* denota 0 ou mais repetições dos tokens dentro das chaves; as chaves não aparecem na lista de tokens de saída.

```
<Texto Marcado de figura geométrica> :=
| <Texto Marcado de Ponto>
| <Texto Marcado de Cadeia de Linha>
| <Texto Marcado de Polígono>
| <Texto Marcado de MultiPonto>
| <Texto Marcado de Cadeia de Linhas Múltiplas>
| <Texto Marcado de MultiPolígono>

<Texto Marcado de Ponto> :=
POINT (<Texto de Ponto>)

<Texto Marcado de Cadeia de Linha> :=
LINESTRING (<Texto de Cadeia de Linha>)

<Texto Marcado de Polígono> :=
POLYGON (<Texto de Polígono>)

<Texto Marcado de MultiPonto> :=
MULTIPOINT (<MultiPoint Text>)

<Texto Marcado de Cadeia de Linhas Múltiplas> :=
MULTILINESTRING (<Texto de Cadeia de Linhas Múltiplas>)

<Texto Marcado de MultiPolígono> :=
MULTIPOLYGON (<Texto de MultiPolígono>)

<Texto de Ponto> := EMPTY
| <Ponto>
| Z (<PontoZ>)
| M (<PontoM>)
| ZM (<PontoZM>)

<Ponto> := <x> <y>
<x> := literal de precisão dupla
<y> := literal de precisão dupla
<PontoZ> := <x> <y> <z>
<x> := literal de precisão dupla
<y> := literal de precisão dupla
<z> := literal de precisão dupla
<PontoM> := <x> <y> <m>
<x> := literal de precisão dupla
<y> := literal de precisão dupla
```

```

<m> := literal de precisão dupla
<PontoZM> := <x> <y> <z> <m>
<x> := literal de precisão dupla
<y> := literal de precisão dupla
<z> := literal de precisão dupla
<m> := literal de precisão dupla

```

```

<Texto de Cadeia de Linha> := EMPTY
| ( <Texto de Ponto > {, <Texto de Ponto > }* )
| Z ( <Texto de Ponto Z > {, <Texto de Ponto Z> }* )
| M ( <Texto de Ponto M > {, <Texto de Ponto M> }* )
| ZM ( <Texto de Ponto ZM > {, <Texto de Ponto ZM> }* )

```

```

<Texto de Polígono> := EMPTY
| ( <Texto de Cadeia de Linha > {, <Texto de Cadeia de Linha > }* )

```

```

<MultiPoint Text> := EMPTY
| ( <Texto de Ponto > {, <Texto de Ponto > }* )

```

```

<Texto de Cadeia de Linhas Múltiplas> := EMPTY
| ( <Texto de Cadeia de Linha > {, <Texto de Cadeia de Linha > }* )

```

```

<Texto de MultiPolígono> := EMPTY
| ( <Texto de Polígono > {, <Texto de Polígono > }* )

```

A sintaxe da função básica é:

função (<descrição do texto>,<SRID db2gse.coordref>)

O SRID, o identificador de referência espacial e a chave primária da tabela SPATIAL_REFERENCES, identifica o sistema de referência espacial que está armazenado na tabela SPATIAL_REFERENCES. Antes de uma figura geométrica ser inserida numa coluna espacial, seu SRID deve corresponder ao SRID da coluna espacial.

A descrição do texto é composta de três componentes básicos que são colocados entre aspas, por exemplo:

```
<'tipo de figura geométrica'> ['tipo de coordenada'] ['lista de coordenada']
```

em que:

tipo de figura geométrica

É um dos seguintes: ponto, cadeia de linha, polígono, multiponto, cadeia de linhas múltiplas ou multipolígono.

tipo de coordenada

Especifica se a figura geométrica terá coordenadas Z ou medidas. Deixe este argumento em branco se nenhum deles existir na figura geométrica. Caso contrário, defina o tipo de coordenada em Z para figuras geométricas que contêm coordenadas Z, M para figuras geométricas com medidas e ZM para figuras geométricas que possuam ambas.

lista de coordenadas

Define os vértices da figura geométrica. As listas de coordenadas são delimitadas por vírgulas e colocadas entre parênteses. As figuras geométricas com vários componentes exigem conjuntos de parênteses abrindo e fechando cada parte do componente. Se a figura geométrica estiver vazia, a palavra-chave EMPTY substituirá a coordenada.

A Tabela 66 mostra uma lista completa de exemplos de todas as representações de texto possíveis.

Tabela 66. Tipos de figura geométrica e suas representações de texto

| Tipo de figura geométrica | Descrição do texto | Comentário |
|----------------------------------|---|---|
| ponto | ponto vazio | ponto vazio |
| ponto | ponto z vazio | ponto vazio com coordenada z |
| ponto | ponto m vazio | ponto vazio com medida |
| ponto | ponto zm vazio | ponto vazio com coordenada z e medida |
| ponto | ponto (10.05 10.28) | ponto |
| ponto | ponto z (10.05 10.28 2.51) | ponto com coordenada z |
| ponto | ponto m (10.05 10.28 4.72) | ponto com medida |
| ponto | ponto zm (10.05 10.28 2.51 4.72) | ponto com coordenada z e medida |
| cadeia de linha | cadeia de linha vazia | cadeia de linha vazia |
| cadeia de linha | cadeia de linha z vazia | cadeia de linha vazia com coordenadas z |
| cadeia de linha | cadeia de linha m vazia | cadeia de linha vazia com medidas |
| cadeia de linha | cadeia de linha zm vazia | cadeia de linha vazia com coordenadas z e medidas |
| cadeia de linha | cadeia de linha(10.05 10.28 , 20.95 20.89) | cadeia de linha |
| cadeia de linha | cadeia de linha z (10.05 10.28 3.09, 20.95 31.98 4.72, 21.98 29.80 3.51) | cadeia de linha com coordenadas z |
| cadeia de linha | cadeia de linha m (10.05 10.28 5.84, 20.95 31.98 9.01, 21.98 29.80 12.84) | cadeia de linha com medidas |
| cadeia de linha | cadeia de linha zm () | cadeia de linha com coordenadas z e medidas |
| polígono | polígono vazio | polígono vazio |

Tabela 66. Tipos de figura geométrica e suas representações de texto (continuação)

| Tipo de figura geométrica | Descrição do texto | Comentário |
|----------------------------------|---|--|
| polígono | polígono z vazio | polígono vazio com coordenadas z |
| polígono | polígono m vazio | polígono vazio com medidas |
| polígono | polígono zm vazio | polígono vazio com coordenadas z e medidas |
| polígono | polígono ((10 10, 10 20, 20 20, 20 15, 10 10)) | polígono |
| polígono | polígono z (()) | polígono com coordenadas z |
| polígono | polígono m (()) | polígono com medidas |
| polígono | polígono zm (()) | polígono com coordenadas z e medidas |
| multiponto | multiponto vazio | multiponto vazio |
| multiponto | multiponto z vazio | multiponto vazio com coordenadas z |
| multiponto | multiponto m vazio | multiponto vazio com medidas |
| multiponto | multiponto zm vazio | multiponto vazio com coordenadas z com medidas |
| multiponto | multiponto vazio | multiponto vazio |
| multiponto | multiponto (10 10, 20 20) | multiponto com dois pontos |
| multiponto | multiponto z (10 10 2, 20 20 3) | multiponto com coordenadas z |
| multiponto | multiponto m (10 10 4, 20 20 5) | multiponto com medidas |
| multiponto | multiponto zm (10 10 2 4, 20 20 3 5) | multiponto com coordenadas z e medidas |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas vazia | cadeia de linhas múltiplas vazia |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas z vazia | cadeia de linhas múltiplas vazia com coordenadas z |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas m vazia | cadeia de linhas múltiplas vazia com medidas |

Tabela 66. Tipos de figura geométrica e suas representações de texto (continuação)

| Tipo de figura geométrica | Descrição do texto | Comentário |
|----------------------------|---|--|
| cadeia de linhas múltiplas | cadeia de linhas múltiplas zm vazia | cadeia de linhas múltiplas vazia com coordenadas z e medidas |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas (()) | cadeia de linhas múltiplas |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas z (()) | cadeia de linhas múltiplas com coordenadas z |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas m (()) | cadeia de linhas múltiplas com medidas |
| cadeia de linhas múltiplas | cadeia de linhas múltiplas zm (()) | cadeia de linhas múltiplas com coordenadas z e medidas |
| multipolígono | multipolígono vazio | multipolígono vazio |
| multipolígono | multipolígono z vazio | multipolígono vazio com coordenadas z |
| multipolígono | multipolígono m vazio | multipolígono vazio com medidas |
| multipolígono | multipolígono z | multipolígono vazio com coordenadas z e medidas |
| multipolígono | multipolígono ((())) | multipolígono |
| multipolígono | multipolígono z ((())) | multipolígono com coordenadas z |
| multipolígono | multipolígono m (((10 10 2, 10 20 3, 20 20 4, 20 15 5, 10 10 2), (50 40 7, 50 50 3, 60 50 4, 60 40 5, 50 40 7))) | multipolígono com medidas |
| multipolígono | multipolígono zm ((())) | multipolígono com coordenadas z e medidas |

As representações (WKB) do modo binário reconhecidas

Spatial Extender possui várias funções que geram figuras geométricas a partir de representações do modo binário:

ST_GeomFromWKB

Cria uma figura geométrica a partir de uma representação WKB de qualquer tipo de figura geométrica.

ST_PointFromWKB

Cria um ponto a partir de uma representação de WKB.

ST_LineFromWKB

Cria uma cadeia de linhas a partir de uma representação de WKB.

ST_PolyFromWKB

Cria um polígono a partir de uma representação de WKB do polígono.

ST_MPointFromWKB

Cria um multiponto a partir de uma representação de WKB.

ST_MLineFromWKB

Cria uma cadeia de linhas múltiplas a partir de uma representação de WKB de cadeia de linhas múltiplas.

ST_MPolyFromWKB

Cria um multipolígono a partir de uma representação de WKB de multipolígono.

A representação do modo binário reconhecida é um fluxo contíguo de bytes. Ela permite que a figura geométrica sejam trocada entre um cliente ODBC e um banco de dados SQL no modo binário. Como estas funções de figura geométrica exigem a definição das estruturas da linguagem de programação para mapear a representação do modo binário, elas destinam-se ao uso dentro de um programa da linguagem de terceira geração (3GL). Elas não se adaptam ao ambiente da linguagem de quarta geração (4GL). A função `ST_AsBinary` converte um valor de figura geométrica existente numa representação do modo binário reconhecida.

A representação do modo binário reconhecido para figura geométrica é obtida pela serialização de uma instância da figura geométrica como uma seqüência de tipos numéricos. Estes tipos são retirados do conjunto (inteiro não-assinado, duplo) e, cada tipo numérico é serializado como uma seqüência de bytes. Os tipos são serializados através de uma das representações padrão, do modo binário, bem-definidas para tipos numéricos (NDR, XDR). Uma tag de um byte que precede os bytes serializados descreve a codificação binária específica (NDR ou XDR) usada para um fluxo de bytes de figura geométrica. A única diferença entre as duas codificações da figura geométrica é uma da ordem de bytes: A codificação XDR é Big Endian; a codificação NDR é Little Endian.

Definições do tipo numérico

Um *inteiro não assinado* é um tipo de dados de 32 bits (4 bytes) que codifica um inteiro não negativo no intervalo [1, 4294967295].

Um *duplo* é um tipo de dados de precisão dupla de (8 bytes) e 64 bits que codifica um número de precisão dupla através do formato de precisão dupla 754 do IEEE.

Estas definições são comuns para XDR e NDR.

XDR (Big Endian) codificação de tipos numéricos

A representação XDR de um inteiro não-assinado é Big Endian (byte mais significativo primeiro).

A representação de XDR de um duplo é Big Endian (bit de sinal é o primeiro bit).

NDR (Little Endian) codificação de tipos numéricos

A representação NDR de um inteiro não-assinado é Little Endian (byte menos significativo primeiro).

A representação NDR de um duplo é Little Endian (bit de sinal é o último byte).

Conversão entre NDR e XDR

A conversão entre os tipos de dados NDR e XDR para inteiros não-assinados e duplos é uma operação simples. Ela envolve a inversão da ordem de bytes dentro de cada inteiro não-assinado ou duplo no fluxo de bytes.

Descrição dos fluxos de bytes WKGeometry

Esta seção descreve a representação do modo binário reconhecida para figura geométrica. O bloco de construção básica é o fluxo de bytes para um ponto, que consiste em dois duplos. Os fluxos de bytes para outras figuras geométricas são construídos através de fluxos de bytes para figuras geométricas que já estejam definidas.

```
// Definições do tipo básico
// byte : 1 byte
// uint32 : inteiro não-assinado de 32 bits (4 bytes)
// double : número de precisão dupla (8 bytes)

// Construindo Blocos : Ponto, Anel Linear

Point {
    double x;
    double y;
};
LinearRing {
    uint32 numPoints;
    Point points[numPoints];
};
enum wkbGeometryType {
    wkbPoint = 1,
    wkbLineString = 2,
    wkbPolygon = 3,
    wkbMultiPoint = 4,
    wkbMultiLineString = 5,
    wkbMultiPolygon = 6
};
enum wkbByteOrder {
    wkbXDR = 0,
    // Big Endian
```

```

    wkbNDR = 1 // Little Endian
};
WKBPoint {
    byte            byteOrder;
    uint32  wkbType; // 1
    Ponto    point;
};
WKBLineString {
    byte            byteOrder;
    uint32  wkbType; // 2
    uint32  numPoints;
    Point   points[numPoints];
};

WKBPolygon {
    byte            byteOrder;
    uint32  wkbType; // 3
    uint32  numRings;
    LinearRing rings[numRings];
};
WKBMultiPoint {
    byte            byteOrder;
    uint32  wkbType; // 4
    uint32  num_wkbPoints;
    WKBPoint WKBPoints[num_wkbPoints];
};
WKBMultiLineString {
    byte            byteOrder;
    uint32  wkbType; // 5
    uint32  num_wkbLineStrings;
    WKBLineString WKBLineStrings[num_wkbLineStrings];
};

wkbMultiPolygon {
    byte            byteOrder;
    uint32  wkbType; // 6
    uint32  num_wkbPolygons;
    WKBPolygon wkbPolygons[num_wkbPolygons];
};

;WKGeometry {
    union {
        WKBPoint            point;
        WKBLineString      linestring;
        WKBPolygon          polygon;
        WKBMultiPoint      mpoint;
        WKBMultiLineString mlinestring;
        WKBMultiPolygon     mpolygon;
    }
};

```

A figura a seguir mostra uma representação NDR.

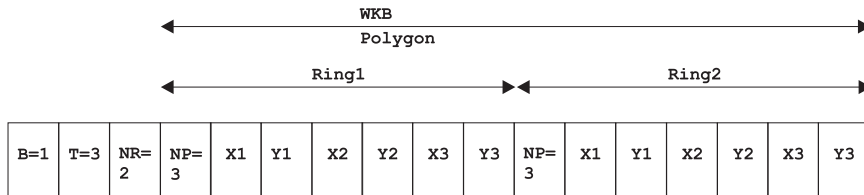


Figura 39. Representação no formato NDR. (B=1) do tipo polígono (T=3) com 2 lineares (NR=2), cada anel tendo 3 pontos (NP=3).

Assertivas para a representação WKB

A representação do modo binário reconhecida para figura geométrica foi projetada para representar as instâncias dos tipos de figura geométrica descritos no Modelo do Objeto de figura geométrica e Especificação do Abstrato da OpenGIS.

Estas assertiva implicam o seguinte para anéis, polígonos e multipolígonos:

Anéis lineares

Os anéis são simples e fechados, significando que os anéis lineares não podem fazer interseção entre si.

Polígonos

Dois anéis lineares no limite de um polígono não podem se cruzarem. Os anéis lineares no limite de um polígono podem fazer interseção no máximo em um único ponto, mas somente como tangente.

Multipolígonos

Os interiores de dois polígonos que são elementos de um multipolígono não podem fazer interseção. Os limites de qualquer polígono que seja elemento de um multipolígono podem tocar somente um número finito de pontos.

As representações de formato ESRI

O Spatial Extender possui várias funções que geram figuras geométricas a partir de representações de formato ESRI. Além da representação bidimensional suportada pela representação do modo binário de GIS aberto, a representação de formato ESRI também suporta as coordenadas Z opcionais e medidas. As seguintes funções geram figura geométrica a partir de um formato ESRI:

GeometryFromShape

Cria uma figura geométrica a partir de uma representação de formato de qualquer tipo de figura geométrica.

PointFromShape

Cria um ponto a partir de uma representação de formato.

LineFromShape

Cria uma cadeia de linhas a partir de uma representação de formato.

PolyFromShape

Cria um polígono a partir de uma representação de formato do polígono.

MPointFromShape

Cria um multiponto a partir de uma representação do formato.

MLineFromShape

Cria uma cadeia de linhas múltiplas a partir de uma representação de formato da cadeia de linhas múltiplas.

MPolyFromShape

Cria um multipolígono a partir de uma representação de formato de multipolígono.

A sintaxe geral destas funções é a mesma. O primeiro argumento é a representação de formato fornecida como um tipo de dados do objeto grande do modo binário (BLOB). O segundo argumento é o identificador de referência espacial para atribuir a figura geométrica. A função `GeometryFromShape` tem a seguinte sintaxe:

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), cr db2gse.coordref)
```

Como estas funções de formato exigem a definição das estruturas da linguagem de programação C para mapear a representação binária, elas destinam-se ao uso dentro de um programa 3GL e não se adequam a um ambiente 4GL. A função `AsShape` converte o valor da função em uma representação de formato ESRI.

Um tipo de formato 0 indica um formato nulo, sem dados geométricos para o formato.

| Valor | Tipo de Shape |
|-------|---------------|
| 0 | Shape Nulo |
| 1 | Ponto |
| 3* | PoliLinha |
| 5 | Polígono |
| 8 | MultiPonto |
| 11 | PontoZ |
| 13 | PoliLinhaZ |

| Valor | Tipo de Shape |
|-------|---------------|
| 15 | PolígonoZ |
| 18 | MultiPontoZ |
| 21 | PontoM |
| 23 | PoliLinhaM |
| 25 | PolígonoM |
| 28 | MultiPontoM |

Nota: * Os tipos de formatos não especificados acima (2, 4, 6 etc.) são reservados para uso futuro.

Tipos de formatos no espaço XY

Ponto

Um ponto consiste num par de coordenadas de precisão dupla na ordem X, Y.

Tabela 67. Conteúdo do fluxo de bytes do ponto

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-------|---------|--------|-----------|
| Byte 0 | Tipo de Shape | 1 | Inteiro | 1 | Little |
| Byte 4 | X | X | Duplo | 1 | Little |
| Byte 12 | Y | Y | Duplo | 1 | Little |

MultiPonto

Um MultiPonto consiste numa coleção de pontos. O quadro delimitador é armazenado na ordem Xmin, Ymin, Xmax, Ymax.

Tabela 68. Conteúdo do fluxo de bytes MultiPonto

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 8 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 40 | Pontos | Pontos | Ponto | NumPoints | Little |

Polilinha

Uma Polilinha é um conjunto ordenado de vértices que consiste em uma ou mais partes. Uma parte é uma seqüência conectada de dois ou mais pontos. Os pontos podem ou não estar conectados entre si. As partes podem ou não fazer interseção entre si.

Como esta especificação não proíbe pontos consecutivos com coordenadas idênticas, os leitores do arquivo de formato deverão manipular tais casos. Por outro lado, as partes degeneradas, de comprimento zero que poderão resultar não são permitidas.

Os campos para uma Polilinha são:

Quadro

O quadro delimitador da Polilinha armazenada na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de partes na Polilinha.

NumPoints

O número total de pontos para todas as partes.

Partes Uma matriz do comprimento NumParts. Cada Polilinha armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada parte na Polilinha são armazenados de ponto a ponto. Os pontos para a parte 2 seguem os pontos da parte 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada parte. Não existe delimitador na matriz de pontos entre as partes.

Tabela 69. Conteúdo do fluxo de bytes Polilinha

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 3 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |

Nota: $X = 44 + 4 * \text{NumParts}$.

Polígono

Um polígono consiste num ou mais anéis. Um anel é uma seqüência conectada de quatro ou mais pontos que forma um loop fechado, sem auto-interseção. Um polígono pode conter vários anéis externos. A ordem dos vértices ou orientação de um anel indica qual lado do anel é o interior do polígono. A área à direita do observador que caminha pelo anel na ordem do

vértice é a área dentro do polígono. Os vértices de anéis que definem os orifícios nos polígonos estão no sentido anti-horário. Os vértices de um único polígono de anel, portanto, estão sempre no sentido horário. Os anéis de um polígono são denominados partes.

Como esta especificação não proíbe pontos consecutivos com coordenadas idênticas, os leitores do arquivo de formato deverão manipular tais casos. Por outro lado, as partes degeneradas, de comprimento zero ou de área zero que poderão resultar não são permitidas.

Os campos para um polígono são:

Quadro

O quadro delimitador do polígono armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de anéis no polígono.

NumPoints

O número total de pontos para todos os anéis.

Partes Uma matriz do comprimento NumParts. Para cada anel, armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada anel no polígono são armazenados de ponto a ponto. Os pontos do Anel 2 seguem os pontos do Anel 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada anel. Não existe delimitador na matriz de pontos entre anéis.

Avisos importantes sobre os formatos do polígono:

- Os anéis estão fechados (o primeiro e o último vértice de um anel DEVEM ser o mesmo).
- A ordem dos anéis na matriz de pontos não é significativa.
- Os polígonos armazenados num arquivo de formato devem estar limpos. Um polígono limpo é um que:
 - Não possui auto-interseções. Isto significa que um segmento pertencente a um anel não pode fazer interseção com um segmento que pertença a outro anel. Os anéis de um polígono podem se tocar nos vértices mas não nos segmentos. Os segmentos colineares são considerados interseção.
 - Tem o interior de um polígono no lado "correto" da linha que o define. A área à direita do observador que passa pelo anel na ordem do vértice é o interior do polígono. Os vértices de um

único polígono de anel, portanto, estão sempre no sentido horário. Os anéis que definem orifícios nestes polígonos têm um sentido anti-horário.

Polígonos "sujos" ocorrem quando os anéis que definem os orifícios no polígono também estão no sentido horário, o que causa a sobreposição dos interiores.

Um Exemplo de Instância de Polígono:

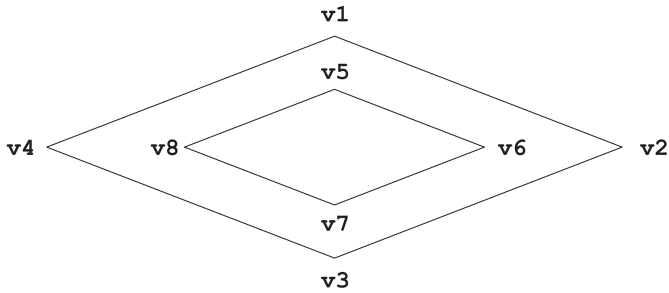


Figura 40. Um polígono com um orifício e oito vértices

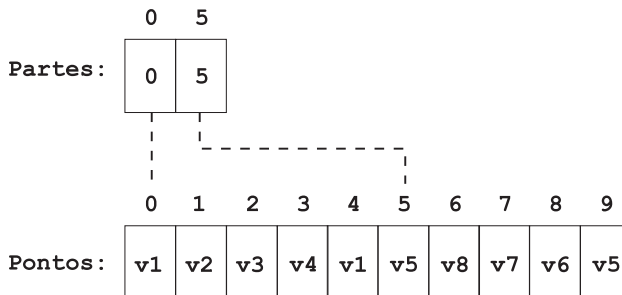


Figura 41. Conteúdo do fluxo de bytes do polígono. NumParts igual a 2 e NumPoints igual a 10. Observe que a ordem dos pontos para o polígono (orifício) do donut é inversa.

Tabela 70. Conteúdo do fluxo de dados do polígono

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-----------|---------|----------|-----------|
| Byte 0 | Tipo de Shape | 5 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |

Tabela 70. Conteúdo do fluxo de dados do polígono (continuação)

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|--------|--------|-------|-----------|-----------|
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |

Nota: $X = 44 + 4 * \text{NumParts}$.

Tipos de formatos medidos no espaço XY

PointM

Um PointM consiste num par de coordenadas de precisão dupla na ordem X, Y, mais uma medida M.

Tabela 71. Conteúdo do fluxo de bytes do pontoM

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-------|---------|--------|-----------|
| Byte 0 | Tipo de Shape | 21 | Inteiro | 1 | Little |
| Byte 4 | X | X | Duplo | 1 | Little |
| Byte 12 | Y | Y | Duplo | 1 | Little |
| Byte 20 | M | M | Duplo | 1 | Little |

MultiPontoM

Os campos para um MultiPontoM são:

Quadro

O quadro delimitador do MultipontoM armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumPoints

O número de pontos.

Pontos

Uma matriz de Pontos do comprimento NumPoints.

NumMs

O número de Medidas que seguem. NumMs pode ter somente dois valores zero se nenhuma Medida seguir este campo; ou igual a NumPoints se Medidas estiverem presentes.

Faixa M

As medidas mínimas e máximas para o MultipontoM armazenado na ordem Mmin, Mmax.

Matriz M

Uma matriz de Medidas do comprimento NumPoints.

Tabela 72. Conteúdo do fluxo de bytes do MultipontoM

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 28 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 40 | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte X | NumMs | NumMs | Inteiro | 1 | Little |
| Byte X+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte X+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte X+20* | Marray | Marray | Duplo | NumPoints | Little |

Notas:

1. $X = 40 + (16 * \text{NumPoints})$
2. * opcional

PolilinhaM

Um arquivo de formato PolilinhaM consiste em uma ou mais partes. Uma parte é uma seqüência conectada de dois ou mais pontos. As partes podem ou não estar conectadas entre si. As partes podem ou não fazer interseção entre si.

Os campos para uma PolilinhaM são:

Quadro

O quadro delimitador da PolilinhaM armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de partes na PolilinhaM.

NumPoints

O número total de pontos para todas as partes.

Partes Uma matriz do comprimento NumParts. Para cada parte, armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada parte na PolilinhaM são armazenados de ponto a ponto. Os pontos para a parte 2 seguem os pontos da parte 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada parte. Não existe delimitador na matriz de pontos entre as partes.

NumMs

O número de Medidas que seguem. NumMs pode ter somente dois valores zero se nenhuma Medida seguir este campo; ou igual a NumPoints se Medidas estiverem presentes.

Faixa M

As medidas mínimas e máximas para a PolilinhaM armazenada na ordem Mmin, Mmax.

Matriz M

Uma matriz do comprimento NumPoints. As medidas de cada parte na PolilinhaM são armazenadas de ponto a ponto. As medidas para a parte 2 seguem as medidas da parte 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada parte. Não existe delimitador na matriz de medida entre as partes.

Tabela 73. Conteúdo do fluxo de bytes da PolilinhaM

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 13 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte Y | NumMs | NumMs | Inteiro | 1 | Little |
| Byte Y+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte Y+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte Y+20* | Marray | Marray | Duplo | NumPoints | Little |

Notas:

1. $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$.
2. * opcional

PolígonoM

Um PolígonoM consiste num número de anéis. Um anel é um loop fechado, sem auto-interseção. Observe que interseções são calculadas no espaço XY, não no espaço XYM. Um PolígonoM pode conter vários anéis externos. Os anéis de um PolígonoM são denominados partes.

Os campos para um PolígonoM são:

Quadro

O quadro delimitador do PolígonoM armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de anéis no PolígonoM.

NumPoints

O número total de pontos para todos os anéis.

Partes Uma matriz do comprimento NumParts. Para cada anel, armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada anel no PolígonoM são armazenados de ponto a ponto. Os pontos do Anel 2 seguem os pontos do Anel 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada anel. Não existe delimitador na matriz de pontos entre anéis.

NumMs

O número de Medidas que seguem. NumMs pode ter somente dois valores zero se nenhuma Medida seguir este campo, ou igual a NumPoints se Medidas estiverem presentes.

Faixa M

As medidas mínimas e máximas para o PolígonoM armazenado na ordem Mmin, Mmax.

Matriz M

Uma matriz do comprimento NumPoints. As medidas de cada anel no PolígonoM são armazenados de ponto a ponto. As medidas do Anel 2 seguem os pontos do Anel 1 etc. A matriz das partes mantém o índice matricial da medida inicial para cada anel. Não existe delimitador na matriz de medida entre os anéis.

Avisos importantes sobre os formatos do PolígonoM:

- Os anéis estão fechados (o primeiro e o último vértice de um anel devem ser o mesmo).
- A ordem dos anéis na matriz de pontos não é significativa.

Tabela 74. Conteúdo do fluxo de bytes do PolígonoM

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|----------|---------|--------|-----------|
| Byte 0 | Tipo de Shape | 15 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |

Tabela 74. Conteúdo do fluxo de bytes do PolígonoM (continuação)

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|-----------|-----------|---------|-----------|-----------|
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte Y | NumMs | NumMs | Inteiro | 1 | Little |
| Byte Y+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte Y+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte Y+20* | Marray | Marray | Duplo | NumPoints | Little |

Notas:

1. $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$.
2. * opcional

Tipos de formatos no espaço XYZ

PontoZ

Um PontoZ consiste em coordenadas triplas e duplas na ordem de X, Y, Z mais uma medida.

Tabela 75. Conteúdo do fluxo de bytes do PontoZ

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-------|---------|--------|-----------|
| Byte 0 | Tipo de Shape | 11 | Inteiro | 1 | Little |
| Byte 4 | X | X | Duplo | 1 | Little |
| Byte 12 | Y | Y | Duplo | 1 | Little |
| Byte 20 | Z | Z | Duplo | 1 | Little |
| Byte 28 | Medida | M | Duplo | 1 | Little |

MultipontoZ

Um MultipontoZ representa um conjunto de PontosZ, da seguinte forma:

- O quadro delimitador é armazenado na ordem Xmin, Ymin, Xmax, Ymax.
- O intervalo delimitador de Z é armazenado na ordem Zmin, Zmax. O intervalo delimitador de M é armazenado na ordem Mmin, Mmax.

Tabela 76. Conteúdo do fluxo de bytes do MultipontoZ

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|---------|---------------|-------|---------|--------|-----------|
| Byte 0 | Tipo de Shape | 18 | Inteiro | 1 | Little |

Tabela 76. Conteúdo do fluxo de bytes do MultipontoZ (continuação)

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|-----------|-----------|---------|-----------|-----------|
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 40 | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte X | Zmin | Zmin | Duplo | 1 | Little |
| Byte X+8 | Zmax | Zmax | Duplo | 1 | Little |
| Byte X+16 | Zarray | Zarray | Duplo | NumPoints | Little |
| Byte Y | NumMs | NumMs | Inteiro | 1 | Little |
| Byte Y+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte Y+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte Y+20* | Marray | Marray | Duplo | NumPoints | Little |

Notas:

1. $X = 40 + (16 * \text{NumPoints})$; $Y = X + 16 + (8 * \text{NumPoints})$
2. * opcional

PolilinhaZ

Uma PolilinhaZ consiste em uma ou mais partes. Uma parte é uma seqüência conectada de dois ou mais pontos. As partes podem ou não estar conectadas entre si. As partes podem ou não fazer interseção entre si.

Os campos para uma PolilinhaZ são:

Quadro

O quadro delimitador da PolilinhaZ armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de partes na PolilinhaZ.

NumPoints

O número total de pontos para todas as partes.

Partes Uma matriz do comprimento NumParts. Para cada parte, armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada parte na PolilinhaZ são armazenados de ponto a ponto. Os pontos para a parte 2 seguem os pontos da parte 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada parte. Não existe delimitador na matriz de pontos entre as partes.

Faixa Z

Os valores mínimos e máximo de Z para a PolilinhaZ armazenada na ordem Zmin, Zmax.

Matriz Z

Uma matriz do comprimento NumPoints. Os valores de Z para cada parte na PolilinhaZ são armazenados de ponto a ponto. Os valores de Z para a parte 2 seguem os valores de Z para a parte 1 e assim por diante. A matriz das partes mantém o índice matricial do ponto inicial para cada parte. Não existe delimitador na matriz Z entre as partes.

NumMs

O número de Medidas que seguem. NumMs pode ter somente dois valores zero se nenhuma Medida seguir este campo; ou igual a NumPoints se Medidas estiverem presentes.

Faixa M

As medidas mínimas e máximas para a PolilinhaZ armazenada na ordem Mmin, Mmax.

Matriz M

Uma matriz do comprimento NumPoints. As medidas para cada parte na PolilinhaZ são armazenados de ponto a ponto. As medidas para a parte 2 seguem as medidas da parte 1 etc. A matriz das partes mantém o índice matricial da medida inicial para cada parte. Não existe delimitador na matriz de medida entre as partes.

Tabela 77. Conteúdo do fluxo de bytes da PolilinhaZ

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 13 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte Y | Zmin | Zmin | Duplo | 1 | Little |
| Byte Y+8 | Zmax | Zmax | Duplo | 1 | Little |
| Byte Y+16 | Zarray | Zarray | Duplo | NumPoints | Little |
| Byte Z | NumMs | NumMs | Inteiro | 1 | Little |
| Byte Z+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte Z+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte Z+20* | Marray | Marray | Duplo | NumPoints | Little |

Notas:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$, $Z = Y + 16 + (8 * \text{NumPoints})$
2. * opcional

PolígonoZ

Um PolígonoZ consiste num número de anéis. Um anel é um loop fechado, sem auto-interseção. Um PolígonoZ pode conter vários anéis externos. Os anéis de um PolígonoZ são denominados partes.

Os campos para um PolígonoZ são:

Quadro

O quadro delimitador do PolígonoM armazenado na ordem Xmin, Ymin, Xmax, Ymax.

NumParts

O número de anéis no PolígonoZ.

NumPoints

O número total de pontos para todos os anéis.

Partes Uma matriz do comprimento NumParts. Para cada anel, armazena o índice de seu primeiro ponto na matriz de pontos. Os índices matriciais são relativos a 0.

Pontos

Uma matriz do comprimento NumPoints. Os pontos de cada anel no PolígonoZ são armazenados de ponto a ponto. Os pontos do Anel 2 seguem os pontos do Anel 1 etc. A matriz das partes mantém o índice matricial do ponto inicial para cada anel. Não existe delimitador na matriz de pontos entre anéis.

Faixa Z

Os valores mínimos e máximo de Z para o arco armazenado na ordem Zmin, Zmax.

Matriz Z

Uma matriz do comprimento NumPoints. Os valores de Z para cada anel no PolígonoZ são armazenados de ponto a ponto. Os valores de Z para o Anel 2 seguem os valores de Z para o Anel 1 e assim por diante. A matriz das partes mantém o índice matricial do valor Z inicial para cada anel. Não existe delimitador na matriz do valor Z entre os anéis.

NumMs

O número de Medidas que seguem. NumMs pode ter somente dois valores zero se nenhuma Medida seguir este campo; ou igual a NumPoints se Medidas estiverem presentes.

Faixa M

As medidas mínimas e máximas para o PolígonoZ armazenado na ordem Mmin, Mmax.

Matriz M

Uma matriz do comprimento NumPoints. As medidas para cada anel no PolígonoZ são armazenados de ponto a ponto. As medidas do Anel 2 seguem os pontos do Anel 1 etc. A matriz das partes mantém o índice matricial da medida inicial para cada anel. Não existe delimitador na matriz de medida entre os anéis.

Avisos importantes sobre os formatos do PolígonoZ:

- Os anéis estão fechados (o primeiro e o último vértice de um anel DEVEM ser o mesmo).
- A ordem dos anéis na matriz de pontos não é significativa.

Tabela 78. Conteúdo do fluxo de bytes do PolígonoZ

| Posição | Campo | Valor | Tipo | Número | Solicitar |
|------------|---------------|-----------|---------|-----------|-----------|
| Byte 0 | Tipo de Shape | 15 | Inteiro | 1 | Little |
| Byte 4 | Caixa | Caixa | Duplo | 4 | Little |
| Byte 36 | NumParts | NumParts | Inteiro | 1 | Little |
| Byte 40 | NumPoints | NumPoints | Inteiro | 1 | Little |
| Byte 44 | Partes | Partes | Inteiro | NumParts | Little |
| Byte X | Pontos | Pontos | Ponto | NumPoints | Little |
| Byte Y | Zmin | Zmin | Duplo | 1 | Little |
| Byte Y+8 | Zmax | Zmax | Duplo | 1 | Little |
| Byte Y+16 | Zarray | Zarray | Duplo | NumPoints | Little |
| Byte Z | NumMs | NumMs | Inteiro | 1 | Little |
| Byte Z+4* | Mmin | Mmin | Duplo | 1 | Little |
| Byte Z+12* | Mmax | Mmax | Duplo | 1 | Little |
| Byte Z+20* | Marray | Marray | Duplo | NumPoints | Little |

Parte 3. Apêndices

Avisos

A IBM pode não oferecer os produtos, serviços ou recursos abordados neste documento em todos os países. Consulte o representante local da IBM para obter informações sobre os produtos e serviços atualmente disponíveis em sua região. Qualquer referência a um produto, programa ou serviço IBM não significa que apenas os produtos, programas ou serviços IBM possam ser utilizados. Qualquer produto, programa ou serviço funcionalmente equivalente, que não infrinja nenhum direito de propriedade intelectual da IBM ou quaisquer outros direitos poderá ser utilizado. Contudo, é de inteira responsabilidade do Cliente a avaliação e verificação da operação de qualquer produto, programa ou serviço não-IBM.

A IBM pode ter patentes ou solicitações de patentes relativas a assuntos tratados nesta publicação. O fornecimento deste documento não lhe concede nenhuma licença sobre essas patentes. Pedidos de licença devem ser enviados, por escrito, à:

Gerência de Relações
Comerciais e Industriais da IBM Brasil
Av. Pasteur, nº 138-146
Botafogo - Rio de Janeiro - RJ
CEP 22290-240
Brasil

O parágrafo a seguir não se aplica ao Reino Unido e a nenhum país em que tais disposições não estejam de acordo com a legislação local: A INTERNATIONAL BUSINESS MACHINES CORPORATION FORNECE ESTA PUBLICAÇÃO "COMO ESTÁ" SEM GARANTIA DE NENHUM TIPO, SEJA EXPRESSA OU IMPLÍCITA, INCLUINDO, MAS NÃO LIMITANDO AS GARANTIAS IMPLÍCITAS DE NÃO-INFRAÇÃO, COMERCIALIZAÇÃO OU ADEQUAÇÃO A UM DETERMINADO OBJETIVO. Alguns países não permitem a exclusão de garantias expressas ou implícitas em certas transações, portanto, esta disposição pode não se aplicar ao Cliente.

Estas informações podem incluir imprecisões técnicas ou erros tipográficos. Periodicamente são feitas alterações nas informações aqui contidas; tais alterações serão incorporadas em futuras edições desta publicação. A IBM pode, a qualquer momento, aperfeiçoar e/ou alterar o(s) produtos e/ou programa(s) descrito(s) nesta publicação.

Todas as referências nestas informações a sites não-IBM na Web são fornecidas apenas por conveniência e, de forma alguma, servem como endosso desses

sites na Web. Os materiais nesses sites na Web não fazem parte dos materiais deste produto IBM e o uso desses sites na Web é de responsabilidade do Cliente.

A IBM pode utilizar ou distribuir todas os comentários fornecidos pelo Cliente da maneira como achar conveniente, sem que isso implique em qualquer compromisso para com o Cliente.

Portadores de licenças deste programa que pretendam ter informações sobre o mesmo com o objetivo de permitir: (1) a troca de informações entre programas criados independentemente e outros programas (incluindo este), e (2) o uso mútuo de informações trocadas, devem entrar em contato com a:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Estas informações podem estar disponíveis, observadas as condições e os termos apropriados, incluindo, em alguns casos, o pagamento de uma taxa.

O programa licenciado descrito neste documento e todo o material licenciado disponível são fornecidos pela IBM sob os termos do Contrato com o Cliente IBM, Contrato de Licença de Programa Internacional IBM ou qualquer outro contrato equivalente entre as partes.

Todos os dados de desempenho aqui contidos foram determinados em um ambiente controlado. Portanto, os resultados obtidos em outros ambientes operacionais podem variar de maneira significativa. Algumas medidas podem ter sido feitas em sistemas em nível de desenvolvimento e não há garantia de que tais medidas serão as mesmas em sistemas geralmente disponíveis. Além disso, algumas medidas podem ter sido estimadas por extrapolação. Os resultados reais podem variar. Os usuários deste documento devem verificar os dados aplicáveis em seus ambientes específicos.

As informações referentes a produtos não-IBM foram obtidas dos fornecedores destes produtos, de seus anúncios publicados e de outras fontes publicamente disponíveis. A IBM não testou estes produtos e não pode confirmar a precisão do desempenho, compatibilidade ou quaisquer outras reivindicações relacionadas aos produtos não-IBM. As questões sobre as capacidades de produtos não-IBM devem ser endereçadas aos fornecedores destes produtos.

Todas as afirmações referentes à direção ou intenção futura da IBM estão sujeitas à alteração ou retratação sem aviso prévio e representam apenas metas e objetivos.

Estas informações podem conter exemplos de dados e relatórios utilizados em operações comerciais diárias. Para ilustrá-las da forma mais completa possível, os exemplos incluem nomes de pessoas, empresas, marcas e produtos. Todos os nomes são fictícios e qualquer semelhança com nomes e endereços utilizados por empresas reais será mera coincidência.

LICENÇA DE DIREITOS AUTORAIS:

Estas informações podem conter programas aplicativos de amostra na linguagem fonte, que ilustram as técnicas de programação em várias plataformas operacionais. O Cliente pode copiar, modificar e distribuir tais programas de amostra em qualquer formato sem pagamento à IBM, para fins de desenvolver, usar, comercializar ou distribuir programas aplicativos que estejam de acordo com a interface de programação de aplicativos da plataforma operacional para a qual os programas de amostra são escritos. Estes exemplos não foram testados completamente em todas as condições. Portanto, a IBM não pode garantir ou confirmar a excelência em confiabilidade, desempenho ou função de tais programas.

Cada cópia ou qualquer parte destes programas de amostra ou qualquer trabalho derivativo deve incluir um aviso de copyright, como segue:

© (o nome de sua empresa) (ano). Partes destes códigos são derivados da IBM Corp. Sample Programs. © Copyright IBM Corp. _insira o ano ou anos_.
Todos os direitos reservados.

Marcas

Os termos a seguir, que podem estar indicados por um asterisco(*), são marcas da International Business Machines Corporation nos Estados Unidos e/ou em outros países.

| | |
|---|------------------|
| ACF/VTAM | IBM |
| AISPO | IMS |
| AIX | IMS/ESA |
| AIX/6000 | LAN DistanceMVS |
| AIXwindows | MVS/ESA |
| AnyNet | MVS/XA |
| APPN | Net.DataOS/2 |
| AS/400 | OS/390 |
| BookManager | OS/400 |
| CICS | PowerPC |
| C Set++ | QBIC |
| C/370 | QMF |
| DATABASE 2 | RACF |
| DataHub | RISC System/6000 |
| DataJoiner | RS/6000 |
| DataPropagator | S/370 |
| DataRefresher | SP |
| DB2 | SQL/DS |
| DB2 Connect | SQL/400 |
| DB2 Extenders | System/370 |
| DB2 OLAP Server | System/390 |
| DB2 Universal Database | SystemView |
| Distributed Relational Database Architecture | VisualAge |
| DRDA | VM/ESA |
| eNetwork | VSE/ESA |
| Extended Services | VTAM |
| FFST | WebExplorer |
| First Failure Support Technology | WIN-OS/2 |

Os termos a seguir são marcas ou marcas registradas de outras empresas:

Microsoft, Windows e Windows NT são marcas ou marcas registradas da Microsoft Corporation.

Java ou todas as marcas e logotipos baseados em Java e Solaris são marcas da Sun Microsystems, Inc. nos Estados Unidos e/ou em outros países.

Tivoli e NetView são marcas da Tivoli Systems Inc. nos Estados Unidos e/ou em outros países.

UNIX é uma marca registrada nos Estados Unidos, em outros países ou em ambos, licenciada exclusivamente através da X/Open Company Limited.

Outros nomes de empresas, produtos ou serviços que podem estar denotados por um asterisco duplo (**) podem ser marcas ou marcas de serviço de terceiros.

Índice Remissivo

A

AIX

- onde os dados de referência estão armazenados 33
- onde são armazenadas as definições de macro para constantes 83

anéis lineares 344

aplicações

- diretrizes para gravação 73
- procedimentos armazenados 83

ArcExplorer

- usando como interface 9, 69

ArcExplorer Java Versão 3.0

- download 29

AsShape 161, 193, 198

ativando bancos de dados para

- operações espaciais
- descrição 10
- discussão 34
- opções de menu do DB2 Control Center 34

B

banco de dados

- ativando operações espaciais
- discussão 34
- opções de menu do DB2 Control Center 34
- programa de amostra 74
- ativando para operações espaciais
- db2gse.gse_enable_db 94
- desativando o suporte para operações espaciais
- db2gse.gse_disable_db 89
- programa de amostra 74

C

cadeias de linhas 159, 166

cadeias de linhas múltiplas 159, 169

camadas

- descrição 12
- exibição do catálogo
- DB2GSE.GEOMETRY_COLUMNS 144
- registrando colunas da exibição como
- janela Criar Camada Espacial 50

camadas (continuação)

- registrando colunas da exibição como (continuação)
- programa de amostra 79
- registrando colunas da view como
- db2gse.gse_register_layer 109
- registrando colunas de tabelas como
- db2gse.gse_register_layer 109
- janela Criar Camada Espacial 48
- programa de amostra 76
- usando db2gse.gse_unregister_layer para cancelar o registro 119

cenário de tarefas 12

Centro de Controle do DB2

- Janela Criar Referências Espaciais 41
- Janela Criar Sistema de Referências Espaciais 40
- Janela Executar Geocoder 57, 58
- janela Exportar Dados Espaciais 64
- Janela Importar Dados Espaciais 60, 62, 63

classe 160

codificação NDR 341, 342

codificação XDR 341, 342

colunas espaciais 53

consultas

- explorando índices espaciais 71
- interfaces para submissão 9, 69
- programa de amostra 79
- tipos de funções espaciais para uso 69
- utilizando funções e predicados espaciais 71

Conteúdo do fluxo de bytes da

PolilinhaM 352

Conteúdo do fluxo de bytes da

PolilinhaZ 356

Conteúdo do fluxo de bytes do

MultipontoM 350, 351

Conteúdo do fluxo de bytes do

MultipontoZ 354

Conteúdo do fluxo de bytes do

PolígonoM 353

Conteúdo do fluxo de bytes do

PolígonoZ 358

Conteúdo do fluxo de bytes do

ponto 346

Conteúdo do fluxo de bytes do

pontoM 350

Conteúdo do fluxo de bytes do

PontoZ 354

Conteúdo do fluxo de bytes

MultiPonto 346

Conteúdo do fluxo de bytes

Polilinha 347

Conteúdo do fluxo de dados do

polígono 349

coordenadas

coordenadas X

descrição 35

propriedades das figuras

geométricas 160

coordenadas Y

descrição 35

propriedades das figuras

geométricas 160

Coordenadas Z

descrição 36

propriedades das figuras

geométricas 161

descrição 6

coordenadas X

descrição 35

propriedade das figuras

geométricas 160

coordenadas Y

descrição 35

propriedade das figuras

geométricas 160

Coordenadas Z

descrição 36

propriedade das figuras

geométricas 161

D

dados de referência 33

dados do atributo 5

dados espaciais

derivados de dados do

atributo 7

derivados de outros dados

espaciais

discussão 7

- dados espaciais (*continuação*)
 - funções espaciais que derivam os dados 184
 - exportando
 - db2gse.gse_export_shape 100
 - discussão 59
 - janela Exportar Dados Espaciais 64
 - programa de amostra 80
 - formatos de arquivo
 - representações de formato ESRI 192, 344
 - representações WKB (binário convencional) 340
 - representações WKB (well-known binary) 191
 - Representações WKT (texto convencional) 335
 - Representações WKT (well-known text) 190
 - importando
 - db2gse.gse_import_sde 102
 - db2gse.gse_import_shape 104
 - discussão 8, 59
 - Janela Importar Dados Espaciais 59, 62
 - programa de amostra 76
 - natureza dos 6
 - dados fonte 5
 - dados geodéticos 328
 - DB2 Control Center
 - janela Criar Camada Espacial para registrar uma coluna da exibição como uma camada 50
 - para registrar uma coluna da tabela como uma camada 48
 - Janela Criar Índice Espacial solicitando o Spatial Extender a partir de 32
 - DB2GSE.COORD_REF_SYS 143
 - DB2GSE.GEOMETRY_COLUMNS 144
 - db2gse.gse_disable_autogc 86
 - db2gse.gse_disable_db 89
 - db2gse.gse_disable_sref 90
 - db2gse.gse_enable_autogc 91
 - db2gse.gse_enable_db 94
 - db2gse.gse_enable_idx 95
 - db2gse.gse_enable_sref 97
 - db2gse.gse_export_shape 100
 - db2gse.gse_import_sde 102
 - db2gse.gse_import_shape 104
 - db2gse.gse_register_gc 107
 - db2gse.gse_register_layer 109
 - db2gse.gse_run_gc 116
 - db2gse.gse_unregist_gc 118
 - db2gse.gse_unregist_layer 119
 - DB2GSE.SPATIAL_GEOCODER 144
 - DB2GSE.SPATIAL_REF_SYS 145
 - Dicas de detecção de problemas
 - programa de amostra 27
 - utilização do runGseDemo 27
 - dimensão 163
 - disparadores
 - ativando a geocodificação automática
 - db2gse.gse_enable_autogc 91
 - desativando a geocodificação automática
 - db2gse.gse_disable_autogc 86
 - usando para chamar o geocoder 45, 54
- ## E
- EBNF (Extended Backus Naur) 323
 - envolpe 149
 - EnvelopesIntersect 177, 199
 - esferóides 326
 - exibições do catálogo
 - DB2GSE.COORD_REF_SYS 143
 - DB2GSE.GEOMETRY_COLUMNS 144
 - DB2GSE.SPATIAL_GEOCODER 144
 - DB2GSE.SPATIAL_REF_SYS 145
 - exterior 158, 162
- ## F
- fatores de deslocamento
 - especificando 38, 41
 - fatores de escala
 - especificando 39, 42
 - figuras geométricas
 - cadeias de linhas 159, 166
 - cadeias de linhas múltiplas 159, 169
 - correspondência a tipos de dados espaciais 159
 - discussão 157
 - grades de índice espacial 149
 - multipolígonos 159, 171
 - multipontos 159, 169
 - polígonos 159, 167
 - pontos 159, 165
 - propriedades
 - classe 160
 - coordenadas X 160
 - coordenadas Y 160
 - Coordenadas Z 161
 - dimensão 163
 - envolpe 149
 - figuras geométricas (*continuação*)
 - propriedades (*continuação*)
 - exterior 158, 162
 - interior 158, 162
 - limite 158, 162
 - medidas 161
 - fluxos de bytes WKBBGeometry 342
 - formatos
 - no espaço XY 346
 - no espaço XYZ 354
 - funções espaciais
 - .ST_Area 222
 - AsShape 193, 198
 - categorizado pelas operações executadas 69
 - EnvelopesIntersect 177, 199
 - GeometryFromShape 192
 - Is3d 161, 203
 - IsMeasured 161, 204
 - LineFromShape 193, 205
 - LocateAlong 188, 207
 - LocateBetween 189, 209
 - M 166, 211
 - MLine FromShape 212
 - MLineFromShape 193
 - MPointFromShape 193, 214
 - MPolyFromShape 193, 215
 - PointFromShape 165, 193, 216
 - PolyFromShape 193, 218
 - predicados 71
 - ShapeToSQL 192, 220
 - ST_Area 168, 171
 - ST_AsBinary 192, 224
 - ST_AsText 191, 225
 - ST_Boundary 162, 226
 - ST_Buffer 187, 228
 - ST_Centroid 168, 171, 230
 - ST_Contains 183, 231
 - ST_Convexhull 233
 - ST_ConvexHull 189
 - ST_CoordDim 162, 235
 - ST_Crosses 180, 237
 - ST_Difference 185, 239
 - ST_Dimension 164, 240
 - ST_Disjoint 175, 242
 - ST_Distance 184, 244
 - ST_Endpoint 166, 245
 - ST_Envelope 163, 246
 - ST_Equals 174, 249
 - ST_ExteriorRing 168, 250
 - ST_GeometryN 169, 252
 - ST_GeometryType 160, 253
 - ST_GeomFromText 190, 255, 335
 - ST_GeomFromWKB 191, 257, 340

funções espaciais (*continuação*)

- ST_InteriorRingN 168, 259
- ST_Intersection 184, 264
- ST_Intersects 176, 266
- ST_IsClosed 167, 170, 267
- ST_IsEmpty 163, 269
- ST_IsRing 167, 271
- ST_IsSimple 163, 273
- ST_IsValid 160, 274
- ST_Length 166, 169, 276
- ST_LineFromText 190, 278, 335
- ST_LineFromWKB 191, 279, 341
- ST_MLineFromText 190, 281, 335
- ST_MLineFromWKB 192, 282, 341
- ST_MPointFromText 190, 284, 335
- ST_MPointFromWKB 192, 285, 341
- ST_MPolyFromText 191, 286, 335
- ST_MPolyFromWKB 192, 287, 341
- ST_NumGeometries 169, 288
- ST_NumInteriorRing 168, 289
- ST_NumPoints 167, 290
- ST_OrderingEquals 175, 291
- ST_Overlaps 179, 292
- ST_Perimeter 169, 294
- ST_Point 165, 295
- ST_PointFromText 165, 296, 335
- ST_PointFromWKB 165, 191, 297, 340
- ST_PointN 166, 299
- ST_PointOnSurface 168, 300
- ST_PolyFromText 190, 301, 335
- ST_PolyFromWKB 191, 302, 341
- ST_Polygon 190, 304
- ST_Relate 184, 305
- ST_SRID 307
- ST_StartPoint 166, 308
- ST_SymmetricDiff 187, 309
- ST_Touches 178, 311
- ST_Transform 312
- ST_Union 186, 314
- ST_Within 181, 315
- ST_WKBToSQL 191, 316
- ST_WKTTToSQL 190, 318
- ST_X 165, 319
- ST_Y 165, 320

tipos

- associadas à figuras geométricas
- instanciáveis 164

funções espaciais (*continuação*)

tipos (*continuação*)

- associadas a propriedades de figuras geométricas 159
- funções do predicado 172
- funções que comparam figuras geométricas 172
- Funções que geram figuras geométricas 184
- funções que mostram relações entre figuras geométricas 172
- troca de dados 190
- usando para explorar índices espaciais 71
- Z 165

G

geocoder padrão 53

geocoders

- ativando a geocodificação automática
 - db2gse.gse_enable_autogc 91
 - discussão 45, 54
 - janela Criar Camada Espacial 48
 - programa de amostra 77
- desativando a geocodificação automática
 - db2gse.gse_disable_autogc 86
 - Janela Executar Geocoder 58
 - programa de amostra 78
- executando em modo batch
 - discussão 54
 - Janela Executar Geocoder 57
 - programa de amostra 76, 78
- executando no modo batch
 - db2gse.gse_run_gc 116
- geocoder padrão 53
- geocoders não-padrão
 - discussão 53
 - usando db2gse.gse_register_gc para registrar 107
 - usando db2gse.gse_unregister_gc para cancelar registro 118
- views do catálogo
 - DB2GSE.SPATIAL_GEOCODER 144
- geocodificação automática 54
- geocodificação batch 54
- geocodificação incremental 54
- geocoding
 - batch 54
 - descrição 7
 - discussão 53

geocoding (*continuação*)

- incremental 54
- precisão 15
- GeometryFromShape 192, 201

I

índices B tree 148

índices de grade 67

índices espaciais 147

- como eles são gerados 149
- criando
 - db2gse.gse_enable_idx 95
 - determinando o tamanho da grade 68, 154
 - Janela Criar Índice Espacial 67
 - programa de amostra 77
- explorando 71
- índices de grade 67
- utilizando 153

informação espacial

- descrição 3
- recuperando e analisando explorando índices espaciais 71
- interfaces para uso 9, 69
- programa de amostra 79
- tipos de funções espaciais para uso 69
- utilizando funções e predicados espaciais 71

instalando o Spatial Extender

- requisitos de hardware e software 18
- verificação 26

interfaces para o Spatial Extender 9

- interior 158, 162

Is3d 161, 203

IsMeasured 161, 204

itens de dados 6

J

janela Criar Camada Espacial

- para registrar uma coluna da exibição como uma camada 50
- para registrar uma coluna da tabela como uma camada 48

Janela Criar Índice Espacial 67

Janela Criar Referências Espaciais 41

Janela Criar Sistema de Referências Espaciais 40

Janela Executar Geocoder 57, 58

janela Exportar Dados Espaciais 64

janela Importar Dados Espaciais 60, 62

- Janela Importar Dados Espaciais 63
- Java 2 Runtime Environment (JRE)
 - v1.2.2 29
- L**
- limite 158, 162
- LineFromShape 193, 205
- LocateAlong 188, 207
- LocateBetween 189, 209
- M**
- M 166, 211
- M falso
 - especificando 39, 43
- matrizes padrão 173
- medidas
 - descrição 36, 161
 - propriedades das figuras geométricas 161
- mensagens 123
- mensagens de aviso 123
- mensagens de erro 123
- mensagens informativas 123
- meridianos principais 330
- MLine FromShape 212
- MLineFromShape 193
- modelo de sistema de coordenadas do POSC/EPSB 323
- mosaico 189
- MPointFromShape 193, 214
- MPolyFromShape 193, 215
- multipolígonos 159, 171
- multipontos 159, 169
- P**
- palavra-chave GEOGCS 324
- palavra-chave PROJCS 324
- palavra-chave UNIT 325
- parâmetros de projeção do mapa 332
- PointFromShape 165, 193, 216
- polígonos 159, 167
- PolyFromShape 193, 218
- pontos 159, 165
- precisão
 - geocodificação 15
 - geocoding 55
 - preservando para sistemas de referência espacial 37
- procedimentos armazenados
 - db2gse.gse_disable_autogc 86
 - db2gse.gse_disable_db 89
 - db2gse.gse_disable_sref 90
 - db2gse.gse_enable_autogc 91
 - db2gse.gse_enable_db 94
 - db2gse.gse_enable_idx 95
- procedimentos armazenados (*continuação*)
 - db2gse.gse_enable_sref 97
 - db2gse.gse_export_shape 100
 - db2gse.gse_import_sde 102
 - db2gse.gse_import_shape 104
 - db2gse.gse_register_gc 107
 - db2gse.gse_register_layer 109
 - db2gse.gse_run_gc 116
 - db2gse.gse_unregister_gc 118
 - db2gse.gse_unregister_layer 119
- programa de amostra
 - descrição 73
- projeções
 - azimutal 331
 - circuito impresso 331
 - cônicas 331
 - mapa
 - parâmetros 332
 - tipos 330
 - projeções azimutais 331
 - projeções cônicas 331
 - projeções do circuito impresso 331
 - projeções do mapa 330
- R**
- recursos geográficos
 - descrição 3
 - representados por dados 4
 - tipos de dados associados 46
- representações de formato ESRI
 - discussão 344
 - funções espaciais associadas 192
- representações WKB (binário convencional)
 - discussão 340
- representações WKB (well-known binary)
 - funções espaciais associadas 191
- Representações WKT (texto convencional)
 - discussão 335
- Representações WKT (well-known text)
 - funções espaciais associadas 190
- requisitos de espaço em disco 19
- S**
- ShapeToSQL 192, 220
- sistema de coordenadas geométricas 325
- sistema de informações geográficas (GIS)
 - criando 10
 - descrição 3
 - utilizando 11
- sistemas de coordenadas 323
 - derivando sistemas de referência espacial de 36
 - descrição 6, 35
 - exibições do catálogo
 - DB2GSE.COORD_REF_SYS 143
- sistemas de referência espaciais
 - criando
 - db2gse.gse_enable_sref 97
 - discussão 35
 - Janela Criar Sistema de Referências Espaciais 40
 - programa de amostra 74
 - descrição 10
 - eliminando
 - db2gse.gse_disable_sref 90
 - programa de amostra 74
 - especificando parâmetros
 - fatores de deslocamento 38, 41
 - fatores de escala 39, 42
 - M falso 39, 43
 - unidades M 40, 43
 - unidades XY 39, 42
 - unidades Z 40, 42
 - X falso 38, 41
 - Y falso 39, 42
 - Z falso 39, 42
 - exibição do catálogo
 - DB2GSE.SPATIAL_REF_SYS 145
- Spatial Extender
 - aplicações
 - diretrizes para gravação 73
 - procedimentos armazenados 83
 - configuração 17
 - exibições do catálogo 143
 - finalidade 3
 - funções espaciais 195
 - instalando
 - requisitos de hardware e software 18
 - interfaces para o 9
 - mensagens de erro, de aviso e informativas 123
 - procedimentos armazenados 83
 - programa de amostra
 - descrição 73
 - recursos
 - dados de referência 33
 - para operações espaciais 34
 - resumo 33
 - solicitando a partir do Centro de Controle do DB2 32

Spatial Extender (*continuação*)
tarefas, resumo de
cenário 12
programa de amostra 73
realizado por procedimentos
armazenados 84
Visão Geral 9
SRID (identificador do sistema de
referência espacial) 337
ST_Area 168, 171, 222
ST_AsBinary 192, 224
ST_AsText 191, 225
ST_Boundary 162, 226
ST_Buffer 187, 228
ST_Centroid 168, 171, 230
ST_Contains 183, 231
ST_Convexhull 233
ST_ConvexHull 189
ST_CoordDim 162, 235
ST_Crosses 180, 237
ST_Difference 185, 239
ST_Dimension 164, 240
ST_Disjoint 175, 242
ST_Distance 184, 244
ST_Endpoint 166, 245
ST_Envelope 163, 246
ST_Equals 174, 249
ST_ExteriorRing 168, 250
ST_GeometryN 169, 252
ST_GeometryType 160, 253
ST_GeomFromText 190, 255, 335
ST_GeomFromWKB 191, 257, 340
ST_InteriorRingN 168, 259
ST_Intersection 184, 264
ST_Intersects 176, 266
ST_IsClosed 167, 170, 267
ST_IsEmpty 163, 269
ST_IsRing 167, 271
ST_IsSimple 163, 273
ST_IsValid 160, 274
ST_Length 166, 169, 276
ST_LineFromText 190, 278, 335
ST_LineFromWKB 191, 279, 341
ST_MLineFromText 190, 281, 335
ST_MLineFromWKB 192, 282, 341
ST_MPointFromText 190, 284, 335
ST_MPointFromWKB 192, 285, 341
ST_MPolyFromText 191, 286, 335
ST_MPolyFromWKB 192, 287, 341
ST_NumGeometries 169, 288
ST_NumInteriorRing 168, 289
ST_NumPoints 167, 290
ST_OrderingEquals 175, 291
ST_Overlaps 179, 292
ST_Perimeter 169, 294

ST_Point 165, 295
ST_PointFromText 165, 296, 335
ST_PointFromWKB 165, 191, 297,
340
ST_PointN 166, 299
ST_PointOnSurface 168, 300
ST_PolyFromText 190, 301, 335
ST_PolyFromWKB 191, 302, 341
ST_Polygon 190, 304
ST_Relate 184, 305
ST_SRID 307
ST_StartPoint 166, 308
ST_SymmetricDiff 187, 309
ST_Touches 178, 311
ST_Transform 312
ST_Union 186, 314
ST_Within 181, 315
ST_WKBToSQL 191, 316
ST_WKTToSQL 190, 318
ST_X 165, 319
ST_Y 165, 320

T

tipos de dados espaciais 45
correspondência à figuras
geométricas 159
descrição 45
Tipos de formatos medidos nos
espaços XY 350

U

unidades angulares 326
unidades lineares 325
unidades M
especificando 40, 43
unidades XY
especificando 39, 42
unidades Z
especificando 40, 42

W

Windows NT
onde os dados de referência estão
armazenados 33
onde são armazenadas as
definições de macro para
constantes 83
WKBGeometry 342

X

X falso
especificando 38, 41

Y

Y falso
especificando 39, 42

Z

Z 165
Z falso
especificando 39, 42

Entre em Contate com a IBM

Se você tiver algum problema técnico, reveja e execute as ações sugeridas pelo *Manual de Detecção de Problemas* antes de entrar em contato com o Suporte ao Cliente do DB2. Este manual sugere informações que você pode reunir para ajudar ao Suporte ao Cliente do DB2 a atendê-lo melhor.

Para obter informações ou solicitar quaisquer produtos do DB2 Universal Database, entre em contato com o representante IBM em uma filial local ou entre em contato com qualquer revendedor de software IBM autorizado.

Se você mora nos Estados Unidos, poderá ligar para um dos seguintes números:

- 1-800-237-5511 para suporte ao cliente
- 1-888-426-4343 para conhecer as opções de serviço disponíveis

Informações Sobre o Produto

Se você mora nos Estados Unidos, poderá ligar para um dos seguintes números:

- 1-800-IBM-CALL (1-800-426-2255) ou 1-800-3IBM-OS2 (1-800-342-6672) para solicitar produtos ou obter informações gerais.
- 1-800-879-2755 para solicitar publicações.

<http://www.ibm.com/software/data/>

As páginas do DB2 na World Wide Web fornecem informações atuais do DB2 sobre notícias, descrições de produtos, programas educativos e muito mais.

<http://www.ibm.com/software/data/db2/library/>

A Biblioteca Técnica de Serviço e o Produto do DB2 (DB2 Product and Service Technical Library) fornecem acesso às perguntas frequentes, correções, manuais e informações técnicas atualizadas do DB2.

Nota: Estas informações podem estar apenas em Inglês.

<http://www.elink.ibm.com/pbl/pbl/>

O site na Web para pedidos de Publicações Internacionais (International Publications) fornece informações sobre como pedir manuais.

<http://www.ibm.com/education/certify/>

O Programa de Certificação Profissional (Professional Certification

Program) do site da IBM na Web fornece informações sobre o teste de certificação para uma variedade de produtos IBM, incluindo o DB2.

ftp.software.ibm.com

Inicie sessão como anônimo. No diretório /ps/products/db2, você pode encontrar demonstrações, correções, informações e ferramentas relacionadas ao DB2 e muitos outros produtos.

comp.databases.ibm-db2, bit.listserv.db2-l

Estes grupos de notícias da Internet estão disponíveis para que os usuários possam discutir suas experiências com os produtos DB2.

On Compuserve: GO IBMDB2

Digite esse comando para acessar os fóruns do IBM DB2 Family. Todos os produtos DB2 são suportados nesses fóruns.

Para obter mais informações sobre como entrar em contato com a IBM fora dos Estados Unidos, consulte o Apêndice A do *Manual de Suporte do Software IBM (IBM Software Support Handbook)*. Para acessar este documento, vá para a seguinte página na Web: <http://www.ibm.com/support/>, e selecione o link IBM Software Support Handbook próximo à parte inferior da página.

Nota: Em alguns países, os revendedores autorizados pela IBM devem entrar em contato com a estrutura de suporte ao revendedor em vez do Centro de Suporte IBM.



Número da Peça: CT7M2NA

Impresso em Brazil

S517-6993-01



(1P) P/N: CT7M2NA



Spine information:



IBM® DB2® Spatial Extender

DB2 Spatial Extender Referência e Manual do
Usuário

Versão 7