

IBM DB2 Spatial Extender



Benutzer- und Referenzhandbuch

Version 7

IBM DB2 Spatial Extender



Benutzer- und Referenzhandbuch

Version 7

Hinweis

Vor Verwendung dieses Handbuchs und des darin beschriebenen Produkts sollten die allgemeinen Informationen unter „Bemerkungen“ auf Seite 361 gelesen werden.

- Die IBM Homepage finden Sie im Internet unter: **ibm.com**
- IBM und das IBM Logo sind eingetragene Marken der International Business Machines Corporation.
- Das e-business Symbol ist eine Marke der International Business Machines Corporation
- Infoprint ist eine eingetragene Marke der IBM.
- ActionMedia, LANDesk, MMX, Pentium und ProShare sind Marken der Intel Corporation in den USA und/oder anderen Ländern.
- C-bus ist eine Marke der Corollary, Inc. in den USA und/oder anderen Ländern.
- Java und alle Java-basierenden Marken und Logos sind Marken der Sun Microsystems, Inc. in den USA und/oder anderen Ländern.
- Microsoft Windows, Windows NT und das Windows-Logo sind Marken der Microsoft Corporation in den USA und/oder anderen Ländern.
- PC Direct ist eine Marke der Ziff Communications Company in den USA und/oder anderen Ländern.
- SET und das SET-Logo sind Marken der SET Secure Electronic Transaction LLC.
- UNIX ist eine eingetragene Marke der Open Group in den USA und/oder anderen Ländern.
- Marken anderer Unternehmen/Hersteller werden anerkannt.

Änderungen in der IBM Terminologie

Die ständige Weiterentwicklung der deutschen Sprache nimmt auch Einfluß auf die IBM Terminologie. Durch die daraus resultierende Umstellung der IBM Terminologie, kann es u. U. vorkommen, dass in diesem Handbuch sowohl alte als auch neue Termini gleichbedeutend verwendet werden. Dies ist der Fall, wenn auf ältere existierende Handbuchauschnitte und/oder Programmteile zurückgegriffen wird.

Zweite Ausgabe (Juni 2001)

Diese Veröffentlichung ist eine Übersetzung des Handbuchs
IBM DB2 Spatial Extender User's Guide and Reference Version 7,
IBM Form SC27-0701-01,

herausgegeben von International Business Machines Corporation, USA

© Copyright International Business Machines Corporation 1998, 2001

© Copyright IBM Deutschland GmbH 1998, 2001

Informationen, die nur für bestimmte Länder Gültigkeit haben und für Deutschland, Österreich und die Schweiz nicht zutreffen, wurden in dieser Veröffentlichung im Originaltext übernommen.

Möglicherweise sind nicht alle in dieser Übersetzung aufgeführten Produkte in Deutschland angekündigt und verfügbar; vor Entscheidungen empfiehlt sich der Kontakt mit der zuständigen IBM Geschäftsstelle.

Änderung des Textes bleibt vorbehalten.

Herausgegeben von:
SW TSC Germany
Kst. 2877
Juni 2001

Inhaltsverzeichnis

Abbildungsverzeichnis	vii
Tabellen	ix
Zu diesem Handbuch	xi
Zielgruppe	xi
Konventionen	xi
Anmerkungen einsenden	xii
<hr/>	
Teil 1. Spatial Extender verwenden	1
Kapitel 1. Informationen zum Spatial Extender	3
Der Zweck des Spatial Extender.	3
Daten, die geografische Merkmale darstellen	4
Wie Daten geografische Merkmale darstellen	4
Das Wesen der räumlichen Daten	6
Woher räumliche Daten stammen	6
Ein Spatial Extender-GIS erstellen und verwenden	8
Schnittstellen zu Spatial Extender und ihre Funktionalität	9
Aufgaben zum Erstellen und Verwenden eines Spatial Extender-GIS	10
Szenario: Eine Versicherungsgesellschaft aktualisiert ihr GIS.	13
Kapitel 2. Spatial Extender installieren	17
Konfiguration von DB2 Spatial Extender	17
Systemvoraussetzungen	18
Unterstützte Betriebssysteme	18
Erforderliche Datenbanksoftware	18
Erforderlicher Plattenspeicherplatz	19
DB2 Spatial Extender für Windows NT und Windows 2000 installieren	20
DB2 Spatial Extender für AIX installieren	22
CD-ROM anhängen	22
SMIT oder den Befehl installp verwenden	25
Exemplarumgebung des DB2 Spatial Extender einrichten	26
Installation überprüfen	26
Hinweise zur Fehlerbehebung für das Beispielprogramm	28

Überlegungen nach Abschluss der Installation	29
Herunterladen von ArcExplorer	30
CD-ROMs mit Bezugsdaten, Daten und Karten für den DB2 Spatial Extender-Geocodierer verwenden	30
Spatial Extender aufrufen	32

Kapitel 3. Ressourcen einrichten	33
Bestand der Ressourcen	33
Bezugsdaten	33
Ressourcen, die eine Datenbank für räumliche Operationen aktivieren	34
Eine Datenbank für räumliche Operationen aktivieren	35
Ein räumliches Bezugssystem erstellen	35
Informationen zu Koordinaten und räumlichen Bezugssystemen	35
Über die Steuerzentrale ein räumliches Bezugssystem erstellen	40

Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren	45
Informationen zu Typen von räumlichen Daten	45
Datentypen für Einheitenmerkmale	46
Datentypen für Merkmale mit mehreren Einheiten	47
Ein Datentyp für alle Merkmale	48
Eine räumliche Spalte für eine Tabelle definieren, diese Spalte als Schicht registrieren und einen Geocodierer zum Verwalten aktivieren	48
Einschränkungen	51
Eine Sichtspalte als Schicht registrieren	51

Kapitel 5. Räumliche Spalten ausfüllen	53
Geocodierer verwenden	53
Informationen zur Geocodierung	53
Den Geocodierer im Stapelbetrieb ausführen	57
Daten importieren und exportieren	59
Informationen zum Importieren und Exportieren	59

Daten von der Datenbankebene in eine neue oder eine vorhandene Tabelle importieren	60	Von räumlichen Funktionen zurückgegebene Nachrichten.	134
Daten von der Tabellenebene in eine vorhandene Tabelle importieren	63	Kapitel 11. Katalogsichten	143
Daten in eine Formdatei exportieren	64	DB2GSE.COORD_REF_SYS	143
Kapitel 6. Räumliche Indizes erstellen	67	DB2GSE.GEOMETRY_COLUMNS	144
Über die Steuerzentrale einen räumlichen Index erstellen	67	DB2GSE.SPATIAL_GEOCODER	145
Größe der Gitterzellen festlegen	68	DB2GSE.SPATIAL_REF_SYS	145
Kapitel 7. Räumliche Informationen abrufen und analysieren	69	Kapitel 12. Räumliche Indizes	147
Methoden zum Ausführen einer räumlichen Analyse	69	Fragment eines Beispielprogramms	147
Räumliche Abfrage erstellen	69	B-Baumstruktur-Indizes	148
Räumliche Funktionen und SQL	69	Möglichkeiten zum Erstellen eines räumlichen Index	149
Räumliche Prädikate und SQL	71	Wie ein räumlicher Index erstellt wird	149
Kapitel 8. Anwendungen für Spatial Extender schreiben	73	Richtlinien zur Verwendung eines räumlichen Index	153
Das Beispielprogramm verwenden	73	Gitterzellengröße auswählen	154
Die Schritte des Beispielprogramms	73	Anzahl der Stufen auswählen	155
Teil 2. Referenzmaterial	81	Kapitel 13. Geometrien und zugeordnete räumliche Funktionen	157
Kapitel 9. Gespeicherte Prozeduren	83	Informationen zu Geometrien	157
db2gse.gse_disable_autogc	86	Merkmale und zugeordnete Funktionen	160
db2gse.gse_disable_db	89	Klasse.	160
db2gse.gse_disable_sref	90	Koordinaten und Maße	161
db2gse.gse_enable_autogc	91	X- und Y-Koordinaten	161
db2gse.gse_enable_db	95	Z-Koordinaten und Maße	161
db2gse.gse_enable_idx	96	Die Funktion ST_CoordDim	162
db2gse.gse_enable_sref	98	Innenbereich, Begrenzung und Außenbereich	163
db2gse.gse_export_shape	101	Einfach oder nicht einfach	163
db2gse.gse_import_sde	103	Leer oder nicht leer	163
db2gse.gse_import_shape	105	Umschlag	163
db2gse.gse_register_gc	108	Dimension	164
db2gse.gse_register_layer	110	Kennung des räumlichen Bezugssystems	165
db2gse.gse_run_gc	117	Exemplarfähige Geometrien und zugeordnete Funktionen	165
db2gse.gse_unregist_gc	119	Punkte	165
db2gse.gse_unregist_layer	120	Linienfolgen	166
Kapitel 10. Nachrichten	123	Polygone.	168
Von der Steuerzentrale zurückgegebene Nachrichten.	123	Mehrpunktangaben	169
Von gespeicherten Prozeduren zurückgegebene Nachrichten	123	Mehrlinienfolgen	170
		Multipolygone	171
		Funktionen, die Beziehungen und Vergleiche zeigen, Geometrien generieren und Werteformate umsetzen.	172
		Funktionen, die Beziehungen oder Vergleiche zwischen geografischen Merkmalen zeigen	173

Funktionen, die neue Geometrien aus vorhandenen generieren.	186	ST_Intersection.	264
Funktionen, die das Format der Werte einer Geometrie umsetzen	192	ST_Intersects	266
Kapitel 14. Räumliche Funktionen für SQL-Abfragen	197	ST_IsClosed.	267
Verschachtelung	198	ST_IsEmpty	269
ST_Geometry in einen Subtyp umwandeln	198	ST_IsRing	271
Eine Gruppe in eine Basisgeometrie umwandeln.	199	ST_IsSimple.	273
AsShape	200	ST_IsValid	274
EnvelopesIntersect	201	ST_Length	276
GeometryFromShape.	203	ST_LineFromText	278
Is3d	205	ST_LineFromWKB	279
IsMeasured	206	ST_MLineFromText	281
LineFromShape	207	ST_MLineFromWKB	282
LocateAlong	209	ST_MPointFromText	284
LocateBetween	211	ST_MPointFromWKB	285
M	213	ST_MPolyFromText	286
MLine FromShape	214	ST_MPolyFromWKB	287
MPointFromShape	216	ST_NumGeometries	288
MPolyFromShape.	217	ST_NumInteriorRing.	289
PointFromShape	218	ST_NumPoints.	290
PolyFromShape	219	ST_OrderingEquals	291
ShapeToSQL	221	ST_Overlaps	292
ST_Area	223	ST_Perimeter	294
ST_AsBinary	225	ST_Point	295
ST_AsText	226	ST_PointFromText.	296
ST_Boundary	227	ST_PointFromWKB	297
ST_Buffer	229	ST_PointN	299
ST_Centroid	231	ST_PointOnSurface	300
ST_Contains	232	ST_PolyFromText	301
ST_ConvexHull	234	ST_PolyFromWKB	302
ST_CoordDim	236	ST_Polygon	304
ST_Crosses	238	ST_Relate	305
ST_Difference	240	ST_SRID	307
ST_Dimension	241	ST_StartPoint	308
ST_Disjoint	243	ST_SymmetricDiff.	309
ST_Distance.	245	ST_Touches	311
ST_Endpoint	246	ST_Transform	312
ST_Envelope	247	ST_Union	314
ST_Equals	249	ST_Within	315
ST_ExteriorRing	250	ST_WKBToSQL	316
ST_GeometryN	252	ST_WKToSQL	318
ST_GeometryType	253	ST_X	319
ST_GeomFromText	255	ST_Y	320
ST_GeomFromWKB	257	Z	321
ST_InteriorRingN.	259	Kapitel 15. Koordinatensysteme	323
		Übersicht über die Koordinatensysteme	323
		Unterstützte lineare Einheiten.	326
		Unterstützte Winkleinheiten	326
		Unterstützte Spheroide	327
		Unterstützte geodätische Fakten	329

Unterstützte primäre Längengrade	331
Unterstützte Kartenprojektionen	331
Unterstützte konische Projektionen	332
Unterstützte azimutale oder planare Projektionen	332
Unterstützte Kartenprojektionsparameter	332

Kapitel 16. Dateiformate für räumliche

Daten	335
Die bekannten OGC-Textdarstellungen	335
Die bekannten OGC-BinärDarstellungen (WKB)	340
Definition numerischer Typen.	341
XDR-Codierung (Big Endian) numerischer Typen.	342
NDR-Codierung (Little Endian) numerischer Typen.	342
Umsetzung zwischen NDR und XDR	342

Beschreibung der WKBBGeometry-Byteströme	342
Aussagen zur WKB-Darstellung	344
Die ESRI-Formdarstellungen	344
Formtypen im XY-Raum	346
Maß-Formtypen in XY-Räumen	350
Formtypen im XYZ-Raum	354

Teil 3. Schlussteil 359

Bemerkungen	361
Marken	364

Index	367
------------------------	------------

Kontaktaufnahme mit IBM	373
Produktinformationen	373

Abbildungsverzeichnis

1. Tabellenzeile, die ein geografisches Merkmal darstellt; Tabellenzeile, deren Adressdaten ein geografisches Merkmal darstellt	5
2. Tabellen mit hinzugefügten Spalten für räumliche Daten	5
3. Tabellen mit räumlichen Daten, die aus Quellendaten abgeleitet wurden	7
4. Tabelle, die neue räumliche Daten enthält, die aus vorhandenen räumlichen Daten abgeleitet wurden	8
5. Client/Server-Konfiguration	18
6. Hierarchie der Typen räumlicher Daten	46
7. Anwendung einer 10.0e0-Gitterstufe	150
8. Auswirkung beim Hinzufügen der Gitterstufen 30.0e0 und 60.0e0	152
9. Hierarchie der vom Spatial Extender unterstützten Geometrien	159
10. Linienfolgeobjekte	168
11. Polygone	168
12. Mehrlinienfolgen	171
13. Multipolygone	172
14. ST_Equals	175
15. ST_Disjoint	177
16. ST_Touches	179
17. ST_Overlaps	180
18. ST_Within	183
19. ST_Contains	185
20. Mindestentfernung zwischen zwei Städten	186
21. ST_Intersection	187
22. ST_Difference	188
23. ST_Union	188
24. ST_Buffer	189
25. LocateAlong	190
26. LocateBetween	191
27. ST_ConvexHull	191
28. Über die Fläche eine Gebäudegrundfläche finden	224
29. Ein Puffer mit einem Fünf-Meilen-Radius wird auf einen Punkt angewendet	230
30. Mit ST_Contains sicherstellen, dass alle Gebäude vollständig auf dem jeweiligen Grundstück liegen	233
31. Mit ST_Crosses die Wasserwege ermitteln, die durch den Lagerungsbereich für gefährliche Abfälle fließen	239
32. Mit ST_Disjoint die Gebäude ermitteln, die nicht innerhalb (in der Schnittmenge) eines Gefahrengiets liegen	244
33. Mit ST_ExteriorRing die Länge der Küstenlinie einer Insel ermitteln	251
34. Mit ST_InteriorRingN die Länge der Seeufer auf allen Inseln ermitteln	259
35. Mit ST_Intersection feststellen, wie groß der gefährdete Bereich in den Gebäuden ist	265
36. Mit ST_Length die Gesamtlänge der Wasserwege im Landkreis ermitteln	277
37. Mit ST_Overlaps die Gebäude ermitteln, die zumindest teilweise im Gefährdungsbereich der Abfall-Lagerstätten liegen	293
38. Mit ST_SymmetricDiff die Gefahrenbereiche ermitteln, die keine sensiblen Bereiche (bewohnte Gebäude) enthalten	310
39. Darstellung im NDR-Format	344
40. Ein Polygon mit einem Loch und acht Scheitelpunkten	349
41. Inhalt des Polygon-Bytestroms	349

Tabellen

1. Erforderlicher Plattenspeicherplatz	19	26. Eingabeparameter für die gespeicherte	
2. CD-ROMs mit Daten und Karten	30	Prozedur db2gse.gse_register_layer.	111
3. Räumliche Funktionen und Operationen	69	27. Ausgabeparameter für die gespeicherte	
4. Regeln zur Nutzung des Index	72	Prozedur db2gse.gse_register_layer.	116
5. Spatial Extender-Beispielprogramm	74	28. Eingabeparameter für die gespeicherte	
6. Eingabeparameter für die gespeicherte		Prozedur db2gse.gse_run_gc.	117
Prozedur db2gse.gse_disable_autogc. . .	87	29. Ausgabeparameter für die gespeicherte	
7. Ausgabeparameter für die gespeicherte		Prozedur db2gse.gse_run_gc.	118
Prozedur db2gse.gse_disable_autogc. . .	88	30. Eingabeparameter für die gespeicherte	
8. Ausgabeparameter für die gespeicherte		Prozedur db2gse.gse_unregist_gc. . .	119
Prozedur db2gse.gse_disable_db.	89	31. Ausgabeparameter für die gespeicherte	
9. Eingabeparameter für die gespeicherte		Prozedur db2gse.gse_unregist_gc. . .	119
Prozedur db2gse.gse_disable_sref. . . .	90	32. Eingabeparameter für die gespeicherte	
10. Ausgabeparameter für die gespeicherte		Prozedur db2gse.gse_unregist_layer..	121
Prozedur db2gse.gse_disable_sref. . . .	90	33. Ausgabeparameter für die gespeicherte	
11. Eingabeparameter für die gespeicherte		Prozedur db2gse.gse_unregist_layer..	122
Prozedur db2gse.gse_enable_autogc. . .	92	34. SQLSTATE-Werte und SQLCODE-	
12. Ausgabeparameter für die gespeicherte		Werte der Nachrichten, die von räumli-	
Prozedur db2gse.gse_enable_autogc. . .	94	chen Funktionen zurückgegeben wer-	
13. Ausgabeparameter für die gespeicherte		den.	140
Prozedur db2gse.gse_enable_db.	95	35. Spalten in der Katalogsicht	
14. Eingabeparameter für die gespeicherte		DB2GSE.COORD_REF_SYS.	143
Prozedur db2gse.gse_enable_idx.	96	36. Spalten in der Katalogsicht	
15. Ausgabeparameter für die gespeicherte		DB2GSE.GEOMETRY_COLUMNS	144
Prozedur db2gse.gse_enable_idx.	97	37. Spalten in der Katalogsicht	
16. Eingabeparameter für die gespeicherte		DB2GSE.SPATIAL_GEOCODER	145
Prozedur db2gse.gse_enable_sref.	98	38. Spalten in der Katalogsicht	
17. Ausgabeparameter für die gespeicherte		DB2GSE.SPATIAL_REF_SYS	145
Prozedur db2gse.gse_enable_sref.. . . .	100	39. Die 10.0e0-Gitterzelleneinträge für die	
18. Eingabeparameter für die gespeicherte		Beispielgeometrien	151
Prozedur db2gse.gse_export_shape. . . .	101	40. Die Schnittpunkte der Geometrien im	
19. Ausgabeparameter für die gespeicherte		dreistufigen Index	153
Prozedur db2gse.gse_export_shape. . . .	102	41. Matrix für ST_Within.	174
20. Eingabeparameter für die gespeicherte		42. Matrix für Gleichheit	176
Prozedur db2gse.gse_import_sde.. . . .	104	43. Matrix für ST_Disjoint	177
21. Ausgabeparameter für die gespeicherte		44. Matrix für ST_Intersects (1).	178
Prozedur db2gse.gse_import_sde.. . . .	104	45. Matrix für ST_Intersects (2).	178
22. Eingabeparameter für die gespeicherte		46. Matrix für ST_Intersects (3).	178
Prozedur db2gse.gse_import_shape. . . .	106	47. Matrix für ST_Intersects (4).	178
23. Ausgabeparameter für die gespeicherte		48. Matrix für ST_Touches (1).	179
Prozedur db2gse.gse_import_shape. . . .	107	49. Matrix für ST_Touches (2)	180
24. Eingabeparameter für die gespeicherte		50. Matrix für ST_Touches (3)	180
Prozedur db2gse.gse_register_gc.	108	51. Matrix für ST_Overlaps (1)	180
25. Ausgabeparameter für die gespeicherte		52. Matrix für ST_Overlaps (2)	181
Prozedur db2gse.gse_register_gc.	109	53. Matrix für ST_Crosses (1)	182

54.	Matrix für ST_Crosses (2)	182	67.	Punkt-Bytestrom, Inhalt	346
55.	Matrix für ST_Within.	184	68.	Mehrpunkt-Bytestrom, Inhalt	346
56.	Matrix für ST_Contains	185	69.	Mehrfachlinien-Bytestrom, Inhalt	347
57.	Equals-Mustermatrix	305	70.	Polygon-Bytestrom, Inhalt	349
58.	Unterstützte lineare Einheiten	326	71.	PunktM-Bytestrom, Inhalt	350
59.	Unterstützte Winkeleinheiten	326	72.	MehrpunktangabeM-Bytestrom, Inhalt	350
60.	Unterstützte Spheroide	327	73.	MehrfachlinieM-Bytestrom, Inhalt	352
61.	Unterstützte geodätische Fakten	329	74.	PolygonM-Bytestrom, Inhalt	353
62.	Unterstützte primäre Längengrade	331	75.	PunktZ-Bytestrom, Inhalt	354
63.	Unterstützte Kartenprojektionen	331	76.	MehrpunktangabeZ-Bytestrom, Inhalt	354
64.	Unterstützte konische Projektionen	332	77.	MehrfachlinieZ-Bytestrom, Inhalt	356
65.	Unterstützte Kartenprojektions- parameter	332	78.	PolygonZ-Bytestrom, Inhalt.	358
66.	Geometrietypen und ihre Text- darstellung	338			

Zu diesem Handbuch

Dieses Buch ist in zwei Teile gegliedert. Der erste Teil enthält Informationen zum Konzept des DB2 Spatial Extender und erläutert die Installation, Konfiguration, Administration und Programmierung des Spatial Extender auf Windows NT- und AIX-Systemen. Der zweite Teil umfasst Referenzinformationen zu gespeicherten Prozeduren, Geometrien, Funktionen, Nachrichten und Katalogsichten, die Sie mit Spatial Extender verwenden.

Technische Änderungen des Textes sind durch eine vertikale Linie links der Änderung gekennzeichnet.

Zielgruppe

Dieses Buch wendet sich an Administratoren, die die räumliche Umgebung einrichten, und an Anwendungsprogrammierer, die Anwendungen mit räumlichen Daten entwickeln.

Konventionen

In diesem Buch werden die folgenden Hervorhebungsconventionen verwendet:

Fettdruck

Kennzeichnet Befehle und Steuerzeichen der grafischen Benutzerschnittstelle (GUI), z. B. Feldnamen, Ordernamen oder Menüauswahlen.

Monospace-Schrift

Kennzeichnet Beispiele zur Codierung von eingegebenem Text.

Kursivschrift

Kennzeichnet Variablen, die Sie durch einen Wert ersetzen müssen. Kursivschrift dient außerdem der Kennzeichnung von Handbuchtiteln und zum Hervorheben von Wörtern.

GROSSBUCHSTABEN

Kennzeichnet SQL-Schlüsselwörter und Namen von Objekten (z. B. Tabellen, Sichten und Server).

Anmerkungen einsenden

Ihre Rückmeldungen helfen IBM, Informationen noch besser bereitzustellen. Senden Sie Ihre Anmerkungen zu diesem Handbuch oder anderen Komponenten der DB2-Dokumentation an uns ein. Sie können Ihre Kommentare auf zwei Arten einsenden:

- Senden Sie Ihre Kommentare über das Web. Sie können hierbei das Online-Kommentarformular IBM Data Management unter <http://www.ibm.com/software/data/rcf> verwenden.
- Senden Sie Ihre Kommentare per E-Mail an comments@vnet.ibm.com. Geben Sie dabei ggf. den Namen des Produkts, die Versionsnummer des Produkts sowie den Namen und die Teilenummer des Buchs an. Wenn Sie Kommentare zu bestimmten Textstellen haben, geben Sie außerdem die Position des entsprechenden Texts an (z. B. Kapitel und Abschnittsnamen, Tabellenummer, Seitennummer oder den Namen eines Hilfeabschnitts) an.

Teil 1. Spatial Extender verwenden

Kapitel 1. Informationen zum Spatial Extender

Dieses Kapitel stellt den Spatial Extender vor. Es beschreibt die Aufgabe des Spatial Extender sowie die von ihm verarbeiteten Daten und zeigt, wie der Spatial Extender verwendet wird. Das Kapitel schließt mit einem kurzen Überblick über den Rest des Handbuchs.

Der Zweck des Spatial Extender

Mit dem Spatial Extender können Sie ein *geografisches Informationssystem* (GIS) erstellen: einen Komplex aus Objekten, Daten und Anwendungen, der das Generieren und Analysieren räumlicher Informationen zu geografischen Merkmalen ermöglicht. *Geografische Merkmale* umfassen die Objekte, die die Oberfläche der Erde bilden oder sich darauf befinden. Diese Objekte bilden sowohl die natürliche Umgebung (beispielsweise Flüsse, Wälder, Berge oder Wüsten) als auch Kulturland (Städte, Wohngebiete, Gewerbegebiete und Begrenzungen bzw. Grenzen).

Räumliche Informationen umfassen beispielsweise folgende Faktoren:

- Standort geografischer Merkmale in Relation zu ihrer Umgebung (z. B. Standorte von Krankenhäusern in Städten oder die Nähe von Wohngebieten zu Erdbebengebieten.)
- Die Beziehung zwischen verschiedenen geografischen Merkmalen (beispielsweise Informationen darüber, dass sich ein Fluss-System in einer bestimmten Region befindet, oder dass bestimmte Brücken in dieser Region über die Zuflüsse dieses Fluss-Systems führen)
- Maße, die für ein oder mehrere geografische Merkmale gelten (beispielsweise die Entfernung zwischen einem Bürogebäude und dem Rand des Grundstücks oder der Umkreis eines Vogelschutzgebiets)

Räumliche Informationen können, für sich alleine oder in Verbindung mit den Ausgabedaten eines traditionellen relationalen Datenbankverwaltungssystems (Relational Database Management System, RDBMS), die Konzeption von Projekten unterstützen und Unternehmens- und strategische Entscheidungen erleichtern. Der Leiter einer Sozialbehörde muss beispielsweise überprüfen, welche Antragsteller und Empfänger von Unterstützungsleistungen tatsächlich im Zuständigkeitsbereich seiner Behörde wohnen. Spatial Extender kann diese Informationen aus der Angabe des Zuständigkeitsbereichs und den Adressen der Personen ableiten.

Oder der Besitzer einer Restaurantkette möchte in anderen Orten der Umgebung weitere Filialen eröffnen. Zum Ermitteln geeigneter Standorte muss er sich Fragen wie die folgenden stellen: Wo in diesen Städten sind die typischen Gäste für meine Restaurants konzentriert? Wo sind die wichtigen Zufahrtsstraßen? Wo ist die Kriminalität am niedrigsten? Wo sind die Restaurants der Konkurrenz? Spatial Extender kann räumliche Informationen in visuellen Anzeigen erstellen, die diese Fragen beantworten. Das zugrundeliegende RDBMS kann Kennzeichnungen und erläuternde Texte für diese Anzeigen bereitstellen.

Dieses Handbuch enthält noch weitere Beispiele zur Verwendung von Spatial Extender, insbesondere in „Kapitel 7. Räumliche Informationen abrufen und analysieren“ auf Seite 69, „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73 und „Kapitel 14. Räumliche Funktionen für SQL-Abfragen“ auf Seite 197.

Daten, die geografische Merkmale darstellen

Dieser Abschnitt bietet einen Überblick über die Daten, die Sie generieren, speichern und bearbeiten, um räumliche Informationen zu erhalten.

Folgende Themen werden hierbei behandelt:

- Wie Daten geografische Merkmale darstellen
- Das Wesen der räumlichen Daten
- Möglichkeiten zum Erstellen räumlicher Daten

Wie Daten geografische Merkmale darstellen

In Spatial Extender kann ein geografisches Merkmal als Zeile in einer Tabelle dargestellt werden oder als Teil einer solchen Zeile. Betrachten Sie beispielsweise zwei der in „Der Zweck des Spatial Extender“ auf Seite 3 erwähnten geografischen Merkmale, Gewerbegebiete und Wohngebiete. In Abb. 1 auf Seite 5 steht jede Zeile der Tabelle BRANCHES für eine Zweigstelle einer Bank. Als Variation dazu steht jede Zeile der Tabelle CUSTOMERS in Abb. 1 auf Seite 5 als Ganzes für einen Kunden der Bank. Ein Teil jeder Zeile — insbesondere die Zellen mit einer Kundenadresse — kann jedoch als Wohnort des Kunden betrachtet werden.

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

Abbildung 1. Tabellenzeile, die ein geografisches Merkmal darstellt; Tabellenzeile, deren Adressdaten ein geografisches Merkmal darstellt. Die Datenzeile in der Tabelle BRANCHES steht für eine Zweigstelle einer Bank. Die Zellen für Adressdaten in der Tabelle CUSTOMERS stehen für den Wohnort des Kunden. Die Namen und Adressen in beiden Tabellen sind frei erfunden.

Die Tabellen in Abb. 1 enthalten Daten, die die Zweigstellen und Kunden der Bank kennzeichnen. Solche Daten werden als *attributive Daten* bezeichnet.

Eine Untermenge der attributiven Daten — die Werte, die die Adresse der Zweigstelle und des Kunden angeben — können in Werte umgesetzt werden, die räumliche Daten liefern. Das Beispiel in Abb. 1 zeigt die Zweigstellenadresse 92467 Airzone Blvd., San Jose CA 95141; eine Kundenadresse lautet 9 Concourt Circle, San Jose CA 95141. Spatial Extender kann diese Adressen in Werte umsetzen, die angeben, wo sich die Zweigstelle und der Wohnort des Kunden in Relation zu der Umgebung befinden. Abb. 2 zeigt die Tabellen BRANCHES und CUSTOMERS mit den neuen Spalten, die solche Werte aufnehmen können.

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

Abbildung 2. Tabellen mit hinzugefügten Spalten für räumliche Daten. In jeder Tabelle wird die Spalte LOCATION die Koordinaten zu der jeweiligen Adresse enthalten.

Wenn Adressen und ähnliche Kennungen als Ausgangspunkt für räumliche Informationen dienen, werden sie als *Quellendaten* bezeichnet. Da die abgeleiteten Werte räumliche Informationen liefern, werden sie als *räumliche Daten*

bezeichnet. Der nächste Abschnitt beschreibt die räumlichen Daten und stellt die ihnen zugeordneten Datentypen vor.

Das Wesen der räumlichen Daten

Viele räumliche Daten bestehen aus Koordinaten. Eine *Koordinate* ist eine Zahl, die eine Position in Relation zu einem Bezugspunkt angibt. Breitengrade sind beispielsweise Koordinaten, die die Position relativ zum Äquator angeben. Längengrade sind Koordinaten, die die Position relativ zum Greenwich-Längengrad angeben. Die Position des Yellowstone Nationalparks ist somit durch seinen Breitengrad (44,45 Grad nördlich vom Äquator) und seinen Längengrad (110,40 Grad westlich des Greenwich-Längengrads) festgelegt.

Breiten- und Längengrade, ihre Bezugspunkte und andere zugeordnete Parameter werden zusammen als *Koordinatensystem* bezeichnet. Es gibt auch Koordinatensysteme, die auf anderen Werten basieren als auf Längen- und Breitengraden. Diese Koordinatensysteme haben eigene Maßeinheiten zur Position, eigene Bezugspunkte und weitere eindeutige Parameter.

Das einfachste räumliche Datenelement besteht aus zwei Koordinaten, die die Position eines einzelnen geografischen Merkmals definieren. (Ein *Datenelement* ist der Wert bzw. sind die Werte in einer Zelle einer relationalen Tabelle.) Ein umfangreicheres räumliches Datenelement besteht aus mehreren Koordinaten, die einen linearen Pfad wie beispielsweise eine Straße oder einen Fluss definiert. Eine dritte Art besteht aus Koordinaten, die den Umriss eines Gebiets definieren, beispielsweise den Rand eines Grundstücks oder eines Überschwemmungsgebiets. Diese und andere Arten von Spatial Extender unterstützte räumlichen Datenelemente werden in „Kapitel 13. Geometrien und zugeordnete räumliche Funktionen“ auf Seite 157 ausführlich beschrieben.

Jedes räumliche Datenelement ist ein Exemplar eines Typ von räumlichen Daten. Der Datentyp für zwei Koordinaten, die die Position eines Standorts kennzeichnen, ist `ST_Point`; der Datentyp für Koordinaten, die lineare Pfade definieren, ist `ST_LineString`; und der Datentyp für Koordinaten, die einen Umfang definieren, ist `ST_Polygon`. Diese Typen sowie andere Datentypen für räumliche Daten sind strukturierte Typen, die zu einer einzigen Hierarchie gehören. Einen Überblick über die Hierarchie finden Sie im Abschnitt „Informationen zu Typen von räumlichen Daten“ auf Seite 45.

Woher räumliche Daten stammen

Sie können räumliche Daten auf folgende Arten erhalten:

- Durch Ableiten aus attributiven Daten
- Durch Ableiten aus anderen räumlichen Daten
- Durch Importieren

Attributive Daten als Quelldaten verwenden

Spatial Extender kann räumliche Daten aus attributiven Daten ableiten, beispielsweise aus Adressen (wie beispielsweise in „Wie Daten geografische Merkmale darstellen“ auf Seite 4). Dieser Prozess wird als *Geocodierung* bezeichnet. Zum Anzeigen der betreffenden Sequenz betrachten Sie Abb. 2 auf Seite 5 als „Vorher“ und Abb. 3 als „Nachher“. Abb. 2 auf Seite 5 zeigt, dass die beiden Tabellen BRANCHES und CUSTOMERS jeweils eine Spalte nur mit Nullwerten enthalten, die die räumlichen Daten aufnehmen kann. Angenommen, der Spatial Extender „geocodiert“ die Adressen in diesen Tabellen zum Abrufen von Koordinaten dieser Adressen und platziert diese Koordinaten in den Spalten. Abb. 3 verdeutlicht dieses Ergebnis.

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

Abbildung 3. Tabellen mit räumlichen Daten, die aus Quelldaten abgeleitet wurden. Die Spalte LOCATION in der Tabelle CUSTOMERS enthält Koordinaten, die ein Geocodierer aus der Adresse in den Spalten ADDRESS, CITY, STATE und ZIP abgeleitet hat. Ebenso enthält die Spalte LOCATION in der Tabelle BRANCHES Koordinaten, die der Geocodierer aus der Adresse in den Spalten ADDRESS, CITY, STATE und ZIP dieser Tabelle abgeleitet hat. Dieses Beispiel ist erfunden; es werden simulierte Koordinaten und keine realen Koordinaten dargestellt.

Spatial Extender verwendet eine als *Geocodierer* bezeichnete Funktion, um attributive Daten in räumliche Daten umzusetzen und diese räumlichen Daten in Tabellenspalten einzutragen. Weitere Informationen zu Geocodierern finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53.

Andere räumliche Daten als Quelldaten verwenden

Räumliche Daten können nicht nur aus attributiven Daten generiert werden, sondern auch aus anderen räumlichen Daten. Die Bank, deren Zweigstellen in der Tabelle BRANCHES definiert sind, möchte beispielsweise wissen, wie viele Kunden ihren Wohnort innerhalb eines Radius von fünf Meilen um die einzelnen Zweigstellen haben. Bevor die Bank diese Informationen aus der Datenbank abrufen kann, muss in der Datenbank die Definition der Zone eingegeben werden, die einen Radius von fünf Meilen um die einzelnen Zweigstellen definiert. Die Funktion ST_Buffer des Spatial Extender kann eine solche Definition erstellen. Mit den Koordinaten der einzelnen Zweigstellen als Eingabe kann ST_Buffer die Koordinaten generieren, die den Umriss der

gewünschten Zonen festlegen. Abb. 4 zeigt die Tabelle BRANCHES mit den von ST_Buffer bereitgestellten Informationen.

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

Abbildung 4. Tabelle, die neue räumliche Daten enthält, die aus vorhandenen räumlichen Daten abgeleitet wurden. Die Koordinaten in der Spalte SALES_AREA wurden mit der Funktion ST_Buffer aus den Koordinaten in der Spalte LOCATION abgeleitet. Wie die Koordinaten in der Spalte LOCATION sind auch diejenigen in der Spalte SALES_AREA nur simuliert und stellen keine realen Koordinaten dar.

Zusätzlich zu ST_Buffer bietet Spatial Extender verschiedene andere Funktionen, mit denen räumliche Daten aus bereits vorhandenen räumlichen Daten abgeleitet werden können. Beschreibungen von ST_Buffer und diesen weiteren Funktionen finden Sie im Abschnitt „Funktionen, die neue Geometrien aus vorhandenen generieren“ auf Seite 186.

Räumliche Daten importieren

Eine dritte Möglichkeit, räumliche Daten zu erhalten, ist das Importieren aus Dateien, die in einem von Spatial Extender unterstützten Format vorliegen. Eine Beschreibung dieser Formate finden Sie in „Kapitel 16. Dateiformate für räumliche Daten“ auf Seite 335. Diese Dateien enthalten Daten, die normalerweise Karten zugeordnet werden: Volkszählungsdaten, Überschwemmungsgebiete, Erdbebenzonen etc. Durch die Verwendung solcher Daten in Verbindung mit den generierten räumlichen Daten können Sie die verfügbaren geografischen Daten erweitern. Wenn eine Katastrophenschutzbehörde beispielsweise ermitteln will, welchen Risiken ein Wohngebiet ausgesetzt ist, könnte mit ST_Buffer eine Zone um das Gebiet herum definiert werden. Anschließend könnten dann Daten zu Überschwemmungsgebieten und Erdbebenzonen importiert werden, um festzustellen, welche dieser Risiken in der Zone vorliegen.

Ein Spatial Extender-GIS erstellen und verwenden

Sie erstellen ein Spatial Extender-GIS durch Einrichten des Spatial Extender und die Entwicklung von GIS-Projekten in den kombinierten Umgebungen von Spatial Extender und seiner zugrundeliegenden DB2 RDBMS. Sie verwenden dieses GIS durch die Implementierung dieser Projekte, d. h. durch Generieren und Analysieren der — räumlichen und herkömmlichen — Informationen, für die diese Projekte konzipiert wurden. Hierzu müssen verschiedene Aufgaben ausgeführt werden. Dieser Abschnitt stellt die Schnittstellen vor, mit

denen Sie diese Aufgaben durchführen können. Außerdem finden Sie hier einen Überblick über diese Aufgaben und ein Szenario, das diese Aufgaben verdeutlicht.

Schnittstellen zu Spatial Extender und ihre Funktionalität

In diesem Abschnitt werden die Schnittstellen vorgestellt, über die Sie ein Spatial Extender-GIS erstellen (Ressourcen definieren, räumliche Daten abrufen, etc.) und verwenden (Informationen zu geografischen Merkmalen generieren und analysieren) können.

Sie können ein Spatial Extender-GIS wie folgt erstellen:

- Mit den Spatial Extender-Fenstern und Menüauswahlpunkten der DB2-Steuerzentrale. Anleitungen hierzu finden Sie in folgenden Kapiteln:
 - „Kapitel 3. Ressourcen einrichten“ auf Seite 33
 - „Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren“ auf Seite 45
 - „Kapitel 5. Räumliche Spalten ausfüllen“ auf Seite 53
 - „Kapitel 6. Räumliche Indizes erstellen“ auf Seite 67
- Durch Ausführen eines Anwendungsprogramms, das gespeicherte Prozeduren des Spatial Extender aufruft. Hinweise zum Entwickeln eines solchen Programms finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.
- Mit der Steuerzentrale und einem Anwendungsprogramm. Sie können beispielsweise über die Steuerzentrale den Standard-Geocodierer aufrufen. Wenn Sie darüber hinaus einen weiteren Geocodierer verwenden wollen, müssen Sie ihn zunächst in Spatial Extender registrieren, indem Sie die gespeicherte Prozedur `db2gse.gse_register_gc` in einem Anwendungsprogramm aufrufen. (Informationen zu Geocodierern über den Standardcodierer hinaus finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53. Informationen zu der gespeicherten Prozedur `db2gse.gse_register_gc` finden Sie im Abschnitt „`db2gse.gse_register_gc`“ auf Seite 108.)
- Mit der Steuerzentrale, einem Anwendungsprogramm oder beiden Oberflächen in Verbindung mit weiteren Schnittstellen. Zum Erstellen einer Tabelle, die die durch eine räumliche Funktion wie beispielsweise einen Geocodierer zu generierenden Daten enthalten soll, können Sie den Befehlszeilenprozessor oder die Schnittstellen der Steuerzentrale verwenden.

Sie können ein Spatial Extender-GIS auf folgende Arten verwenden:

- Durch die grafische Wiedergabe von Daten mit einem Geobrowser; ein Beispiel hierfür ist der vom Environmental Systems Research Institute (ESRI) angebotene ArcExplorer Java Version 3.0.
- Durch das explizite Übergeben von SQL-Abfragen über die DB2-Steuerzentrale oder den Befehlszeilenprozessor
- Durch das Übergeben von SQL-Abfragen aus einem Anwendungsprogramm

Aufgaben zum Erstellen und Verwenden eines Spatial Extender-GIS

Dieser Abschnitt bietet einen Überblick über die Aufgaben, über die Sie ein Spatial Extender-GIS erstellen und verwenden. Die Aufgaben, über die Sie das GIS erstellen, umfassen das Einrichten des Spatial Extender und die Entwicklung von GIS-Projekten. Die Aufgaben zum Verwenden des GIS umfassen das Implementieren der Projekte. Diese Übersicht beginnt mit dem Einrichten des Spatial Extender; im Anschluss daran werden die Entwicklung und Implementierung einer GIS-Projekts beschrieben. Der Abschnitt schließt mit einem Hinweis darauf, wie sich die in der Übersicht beschriebenen Aufgaben in der täglichen Praxis unterscheiden können.

Den Spatial Extender einrichten

Das Einrichten des Spatial Extender umfasst folgende Punkte:

1. Planen und Vorbereiten (Festlegen, welche GIS-Projekte entwickelt werden sollen, welche Datenbank für den Spatial Extender aktiviert werden soll, Auswählen der Mitarbeiter für die Verwaltung des Spatial Extender und die Entwicklung der Projekte, etc.).
2. Installieren des Spatial Extender.
3. Bereitstellen der Ressourcen zur Unterstützung der GIS-Projekte, z. B.:

Vom Spatial Extender bereitgestellte Ressourcen

Hierzu gehören ein Systemkatalog, Typen von räumlichen Daten, räumliche Funktionen (einschließlich eines Standard-Geocodierers) etc. Das Einrichten dieser Ressourcen wird als *Aktivieren der Datenbank für räumliche Operationen* bezeichnet.

Von Benutzern, Lieferanten oder beiden entwickelte Geocodierer

Der Standard-Geocodierer wandelt Adressen in den USA um in räumliche Daten. Ihre Organisation und andere Lieferanten können Geocodierer bereitstellen, die Adressen aus anderen Regionen sowie andere Arten attributiver Daten in räumliche Daten umsetzen.

Anweisungen zum Installieren des Spatial Extender finden Sie in „Kapitel 2. Spatial Extender installieren“ auf Seite 17. Anweisungen zum Verwenden der Steuerzentrale zum Bereitstellen von Ressourcen finden Sie in „Kapitel 3. Ressourcen einrichten“ auf Seite 33. Richtlinien zur Verwendung eines Anwendungsprogramms zu diesem Zweck finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73. Ein Szenario, das den Gesamtaufwand beim Einrichten des Spatial Extender verdeutlicht, finden Sie im Abschnitt „Ein System zur Integration räumlicher und traditioneller Daten“ auf Seite 13.

Ein GIS-Projekt entwickeln und implementieren

Die Entwicklung und Implementierung eines GIS-Projekts umfasst folgende Aufgaben:

1. Planen und Vorbereiten (Projektziele definieren, Ermitteln, welche Tabellen und Daten erforderlich sind, Festlegen der zu verwendenden Koordinatensysteme etc.).
2. Festlegen, welche räumlichen Bezugssysteme verwendet werden sollen. Die Koordinatenwerte umfassen normalerweise positive ganze Zahlen, negative Zahlen sowie Dezimalzahlen. Spatial Extender muss jedoch alle Koordinatenwerte in Form von ganzen Zahlen speichern. Ein *räumliches Bezugssystem* ist eine Gruppe von Parametern, die definiert, wie negative Zahlen und Dezimalzahlen in einem bestimmten Koordinatensystem in positive ganze Zahlen umgewandelt werden sollen, so dass Spatial Extender sie speichern kann. Nachdem Sie festgelegt haben, welches Koordinatensystem für eine räumliche Spalte verwendet werden soll, müssen Sie das räumliche Bezugssystem angeben, über das die erforderliche Umwandlung für diese Spalte durchgeführt werden kann. Wenn ein vorhandenes räumliches Bezugssystem Ihren Anforderungen entspricht, können Sie es verwenden; andernfalls können Sie ein solches System erstellen.
3. Definieren einer oder mehrerer Spalten für räumliche Daten, Registrieren der Spalten für Spatial Extender und Aktivieren eines Geocodierers, der diese Spalten automatisch verwaltet.

Die Registrierung einer räumlichen Spalte umfasst ihre Aufzeichnung im Spatial Extender-Katalog. Von dem Zeitpunkt der Registrierung ab wird die Spalte als *Schicht* bezeichnet, da durch die aus ihr generierten Daten der von Ihrem GIS erstellten virtuellen geografischen Landschaft eine geologische Schicht hinzugefügt wird. Nach der Registrierung können Sie räumliche Operationen mit der Spalte durchführen; Sie können sie beispielsweise ausfüllen und einen räumlichen Index dafür definieren.

4. Ausfüllen räumlicher Spalten:
 - Legen Sie für ein Projekt, das einen Geocodierer erfordert, Parameter für den Geocodierer fest. Führen Sie das Projekt anschließend aus, so dass es in einer einzigen Operation alle verfügbaren Quelldaten geocodiert und die resultierenden Koordinaten in einer Schicht lädt.
 - Importieren Sie die räumlichen Daten, falls dies für das Projekt erforderlich ist.
5. Ermöglichen Sie den Zugriff auf räumliche Spalten. Dies umfasst insbesondere das Definieren von Indizes, die DB einen schnellen Zugriff auf die räumlichen Daten ermöglichen, sowie das Definieren von Sichten, über die die Benutzer die in Wechselbeziehung zueinander stehenden Daten effizient abrufen können. Nach dem Definieren einer solchen Sicht müssen Sie ihre räumlichen Spalten als Schichten registrieren.
6. Generieren und Analysieren räumlicher Informationen und der dazugehörigen Geschäftsinformationen. Dies umfasst das Abfragen räumlicher Spalten.

ten und der dazugehörigen Attributspalten. Bei solchen Abfragen können Sie Spatial Extender-Funktionen verwenden, die eine Vielzahl verschiedener Informationen zurückgeben, beispielsweise den Mindestabstand zwischen zwei geografischen Merkmalen oder die Koordinaten, die einen Bereich um ein geografisches Merkmal herum definieren. Informationen zu der Funktion, die solche Koordinaten zurückgibt, ST_Buffer, finden Sie in den Abschnitten „Andere räumliche Daten als Quelldaten verwenden“ auf Seite 7 und „ST_Buffer“ auf Seite 229. Beispiele zu den Abfragen, die räumliche Funktionen verwenden, finden Sie in „Kapitel 7. Räumliche Informationen abrufen und analysieren“ auf Seite 69 und „Kapitel 14. Räumliche Funktionen für SQL-Abfragen“ auf Seite 197.

Anweisungen zum Verwenden der Steuerzentrale zur Ausführung der Aufgaben bei der Entwicklung eines GIS-Projekts finden Sie in den folgenden Kapiteln:

- „Kapitel 3. Ressourcen einrichten“ auf Seite 33
- „Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren“ auf Seite 45
- „Kapitel 5. Räumliche Spalten ausfüllen“ auf Seite 53
- „Kapitel 6. Räumliche Indizes erstellen“ auf Seite 67

Richtlinien zur Verwendung der Steuerzentrale bei der Implementierung eines GIS-Projekts finden Sie in „Kapitel 7. Räumliche Informationen abrufen und analysieren“ auf Seite 69.

Richtlinien zur Verwendung eines Anwendungsprogramms bei der Entwicklung und Implementierung eines GIS-Projekts finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Ein Szenario, das den Gesamtaufwand verdeutlicht, finden Sie im Abschnitt „Ein Projekt zum Einrichten von Niederlassungen und zum Anpassen von Prämien“ auf Seite 14.

Unterschiedliche Gruppen von Aufgaben

Die Gruppe von Aufgaben, die Sie zum Erstellen und Verwenden eines Spatial Extender-GIS ausführen, kann in Inhalt und Reihenfolge variieren, je nach Ihren Anforderungen und den verwendeten Schnittstellen. Betrachten Sie beispielsweise die Aufgabe zum Definieren von Spalten für räumliche Daten, das Registrieren der Spalten als Schichten und das Aktivieren eines Geocodierers, der diese Spalten automatisch verwaltet. Über die Steuerzentrale können Sie diese Ausgaben über ein einziges Fenster zusammen ausführen. Wenn Sie jedoch gespeicherte Prozeduren von einem Programm aus aufrufen, können Sie diese Aufgaben separat ausführen und nach eigenem Ermessen zeitlich steuern.

Szenario: Eine Versicherungsgesellschaft aktualisiert ihr GIS

Dieser Abschnitt zeigt ein Szenario zur Verdeutlichung der Gruppe von Aufgaben, die im vorigen Abschnitt beschrieben wurden.

Die Informationssystemumgebungen der Versicherungsgesellschaft Safe Harbor Real Estate umfasst ein allgemeines DB2-Datenbanksystem und ein separates GIS-Datenbankverwaltungssystem. Bis zu einem gewissen Grad können Abfragen Kombinationen von Daten aus beiden Systemen liefern. Eine DB2-Tabelle speichert beispielsweise Informationen zum Umsatz, und eine GIS-Tabelle speichert die Standorte der Zweigstellen des Unternehmens. Somit können auch die Standorte der Niederlassungen ermittelt werden, die einen Umsatz einer bestimmten Höhe erzielen. Die Daten aus den beiden Systemen können jedoch nicht integriert werden (die Benutzer können beispielsweise nicht DB2-Spalten mit GIS-Spalten verbinden), und DB2-Dienste wie beispielsweise die Optimierung von Abfragen stehen für das GIS nicht zur Verfügung. Zur Überwindung dieser Nachteile kauft Safe Harbor Spatial Extender und richtet eine neue GIS-Entwicklungsabteilung ein. In den folgenden Abschnitten wird beschrieben, wie diese Abteilung Spatial Extender einrichtet und ihr erstes Projekt realisiert.

Ein System zur Integration räumlicher und traditioneller Daten

Die GIS-Abteilung von Safe Harbor geht zum Einrichten des Spatial Extender wie folgt vor:

1. Die Abteilung bereitet das Einbeziehen des Spatial Extender in seine DB2-Umgebung vor. Zum Beispiel:
 - a. Das Management-Team der Abteilung benennt ein Team für die räumliche Verwaltung mit der Aufgabe, Spatial Extender zu installieren und zu implementieren. Ein anderes Team für die räumliche Analyse soll räumliche Daten generieren und analysieren.
 - b. Da die unternehmerischen Entscheidungen von Safe Harbor hauptsächlich durch die Kundenanforderungen bestimmt sind, beschließt das Management-Team, Spatial Extender in der Datenbank zu installieren, die Informationen über die Kunden enthält. Die meisten dieser Informationen sind in der Tabelle CUSTOMERS gespeichert.
Die Mitglieder der GIS-Entwicklungsabteilung nennen diese ausgewählte Datenbank einfach eine *GIS-Datenbank*. Dabei ist ihnen durchaus bewusst, dass diese Datenbank nicht ausschließlich für GIS-Zwecke reserviert ist; sie kann nach wie vor auch von anderen Anwendungen verwendet werden.
2. Das Team für die räumliche Verwaltung installiert den Spatial Extender.
3. Das Team für die räumliche Verwaltung richtet die für GIS-Projekte erforderlichen Ressourcen ein:

- Das Team stellt über die Steuerzentrale die Ressourcen bereit, die die GIS-Datenbank für räumliche Operationen aktiviert. Hierzu gehören der Spatial Extender-Katalog, Typen von räumlichen Daten, räumliche Funktionen etc.
- Da Safe Harbor seine Aktivitäten auch nach Kanada ausdehnen will, beginnt das Team für die räumliche Verwaltung damit, Angebote von kanadischen Anbietern über Geocodierer einzuholen, die Adressen in Kanada in räumliche Daten umsetzen.

Ein Projekt zum Einrichten von Niederlassungen und zum Anpassen von Prämien

Zur Ausführung ihres ersten GIS-Projekts mit Spatial Extender geht die GIS-Entwicklungsabteilung wie folgt vor:

1. Die Abteilung bereitet die Entwicklung des Projekts vor, z.B.:
 - Das Management-Team definiert die Ziele für das Projekt:
 - Ermitteln, wo neue Zweigstellen eingerichtet werden sollen
 - Anpassen der Prämien entsprechend der Nähe des Kundenwohnorts zu Gefahrengebieten (Gebieten mit hoher Verkehrsunfallquote, hoher Kriminalitätsrate, Überschwemmungsgebiet, Erdbebengebiet, etc.).
 - Das GIS-Projekt umfasst Kunden und Niederlassungen in den USA. Das Team für die räumliche Verwaltung beschließt daher die Verwendung folgender Systeme:
 - Koordinatensysteme, die Standorte in den Teilen der USA exakt beschreiben, in denen Safe Harbor präsent ist.
 - Standard-Geocodierer, da dieser speziell für die Geocodierung von Adressen in den USA konzipiert wurde.
 - Das Team für die räumliche Verwaltung legt fest, welche Daten zum Erreichen der Projektziele erforderlich sind und welche Tabellen diese Daten enthalten sollen.
2. Über die Steuerzentrale erstellt das Team für die räumliche Verwaltung zwei räumliche Bezugssysteme. Das eine System legt fest, wie Koordinaten, die die Standorte der Niederlassungen definieren, in Datenelemente umgewandelt werden sollen, die der Spatial Extender speichern kann. Das andere System legt fest, wie Koordinaten, die die Wohnorte der Kunden definieren, in Datenelemente umgewandelt werden sollen, die der Spatial Extender speichern kann.
3. Über die Steuerzentrale definiert das Team für die räumliche Verwaltung die Spalten für die räumlichen Daten, registriert sie als Schichten und aktiviert einen Geocodierer, der diese Spalten automatisch verwaltet:
 - Das Team fügt der Tabelle CUSTOMERS eine Spalte LOCATION hinzu. Die Tabelle enthält bereits Kundenadressen. Der Standard-Geocodierer setzt diese Adressen um in räumliche Daten und lädt diese Daten in eine Spalte LOCATION.

- Das Team erstellt eine Tabelle OFFICES für die Daten, die jetzt in dem separaten GIS enthalten sind. Diese Daten enthalten die Adressen der Zweigstellen von Safe Harbor, räumliche Daten, die über einen Geocodierer aus diesen Adressen abgeleitet wurden, und räumliche Daten, die eine Zone mit einem Radius von fünf Meilen um jede Niederlassung definieren. Die von einem Geocodierer generierten Daten werden in eine Spalte LOCATION geschrieben. Die Daten zur Definition der Zone werden in die Spalte SALES_AREA geschrieben.
 - Das Team registriert die beiden LOCATION-Spalten und die beiden SALES_AREA-Spalten als Schichten.
 - Das Team aktiviert den Standard-Geocodierer zur automatischen Verwaltung der beiden LOCATION-Spalten.
4. Das Team für die räumliche Verwaltung füllt die Spalte LOCATION der Tabelle CUSTOMER, die gesamte Tabelle OFFICES und eine neue Tabelle HAZARD_ZONES aus:
- Das Team füllt über die Steuerzentrale die Spalte LOCATION der Tabelle CUSTOMER aus:
 - a. Das Team weist den Geocodierer an, nur unter der folgenden Bedingung räumliche Daten für eine Adresse in die Spalte LOCATION einzutragen: wenn eine Adresse und ihr Pendant in den Aufzeichnungen der United States Census Bureau zu 100 Prozent übereinstimmen. (Eine Datei der vom Census Bureau bereitgestellten Adressen wird zusammen mit Spatial Extender ausgeliefert. Bevor der Geocodierer eine Adresse aus den Quelldaten in räumliche Daten umwandeln kann, muss der Geocodierer versuchen, diese Adresse dem entsprechenden Pendant in der Datei zuzuordnen. Die Benutzer geben an, wie hoch der übereinstimmende Prozentsatz sein muss, damit die räumlichen Daten in eine Tabelle geschrieben werden. Dieser Prozentsatz wird als *Genauigkeit* bezeichnet.)
 - b. Das Team führt den Geocodierer im Stapelbetrieb aus, damit alle Adressen in der Tabelle in einem Arbeitsgang geocodiert werden können. Zum Leidwesen des Teams weist der Geocodierer ungefähr jede zehnte Adresse zurück!
 - c. Das Team vermutet, dass es sich bei den zurückgewiesenen Daten um neue Adressen handelt, für die keine genaue Entsprechung in der Datei des Census Bureau vorhanden ist. Zur Lösung des Problems reduziert das Team die Genauigkeit auf 85.
 - d. Das Team führt den Geocodierer erneut im Stapelbetrieb aus. Der Anteil der zurückgewiesenen Adressen sinkt auf ein akzeptables Maß.

- Mit einem von dem separaten GIS bereitgestellten Dienstprogramm lädt das Team die Niederlassungsdaten in eine Datei. Anschließend importiert das Team diese Daten über die Steuerzentrale aus der Datei in die neue Tabelle OFFICES.
 - Über die Steuerzentrale erstellt das Team eine Tabelle HAZARD ZONES, registriert ihre räumlichen Spalten als Schichten und importiert Daten in diese Spalten. Die Daten stammen aus einer Datei, die der Anbieter der Karte zur Verfügung gestellt hat.
5. Über die Steuerzentrale ermöglicht das Team für die räumliche Verwaltung den Zugriff auf die neuen Schichten:
 - Das Team erstellt Indizes für diese Schichten.
 - Das Team erstellt eine Sicht, die Spalten aus den Tabellen CUSTOMERS und HAZARD ZONES miteinander verknüpft. Anschließend registriert das Team die räumlichen Spalten der Sichten als Schichten.
 6. Das Team für die räumliche Verwaltung führt Abfragen aus, um Informationen abzurufen, mit denen die ursprünglichen Ziele erreicht werden können: Ermitteln, ob neue Zweigstellen eingerichtet werden sollen, und Anpassen der Prämien entsprechend der Nähe der Kunden zu Gefahrengebieten.

Kapitel 2. Spatial Extender installieren

Dieses Kapitel enthält Anweisungen zur Installation von DB2 Spatial Extender für AIX, Windows NT und Windows 2000. Außerdem wird auf folgende Themen eingegangen:

- Konfiguration von DB2 Spatial Extender
- Systemvoraussetzungen
- DB2 Spatial Extender für Windows NT und Windows 2000 installieren
- DB2 Spatial Extender für AIX installieren
- Installation überprüfen
- Überlegungen nach Abschluss der Installation
- Spatial Extender aufrufen

Konfiguration von DB2 Spatial Extender

Ein Spatial Extender-System besteht aus einer allgemeinen DB2-Datenbank, dem Spatial Extender und einem Geobrowser (z. B. ArcExplorer Java Version 3.0). Normalerweise befindet sich eine für räumliche Operationen aktivierte Datenbank auf dem Server. Mit Hilfe von Client-Anwendungen können Sie über die gespeicherten Prozeduren des Spatial Extender und über räumliche Abfragen auf räumliche Daten zugreifen. Sie können DB2 Spatial Extender auch in einer eigenständigen Umgebung konfigurieren. Bei dieser Konfiguration befinden sich Client und Server auf derselben Maschine. Sowohl bei Client/Server- als auch bei eigenständigen Konfigurationen können Sie räumliche Daten mit einem Geobrowser anzeigen.

IBM bietet zur Zeit keinen Geobrowser an, der visuelle Abfrageergebnisse erzeugen kann. Weitere Informationen über Geobrowser sowie eine Bezugsquelle finden Sie unter „Herunterladen von ArcExplorer“ auf Seite 30.

Abb. 5 auf Seite 18 verdeutlicht die Architektur des Spatial Extender.

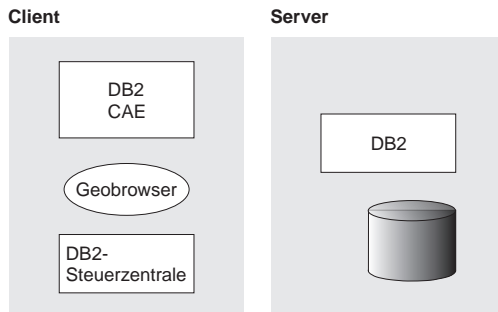


Abbildung 5. Client/Server-Konfiguration

Systemvoraussetzungen

In diesem Abschnitt werden die Software- und Hardwarevoraussetzungen für DB2 Spatial Extender erläutert.

Unterstützte Betriebssysteme

Spatial Extender kann unter AIX Version 4.2 (oder höher), unter Windows NT Version 4.0 (oder höher) mit Service Pack 5 und unter Windows 2000 installiert werden.

Anmerkung: Wenn Sie beabsichtigen, räumliche Daten mit ArcSDE wiederzugeben und anzuzeigen, benötigen Sie AIX Version 4.3.3 oder höher.

Erforderliche Datenbanksoftware

Als Voraussetzung für die Installation von DB2 Spatial Extender muss die folgende DB2-Software auf dem Client und auf dem Server installiert und konfiguriert sein:

Client-Software

Für DB2 Spatial Extender-Client-Produkte:

- DB2 Administration Client, Version 7.1
Wenn Sie nicht beabsichtigen, die DB2-Steuerzentrale zu verwenden, mit einem Geobrowser auf räumliche Daten zuzugreifen oder das DB2 Spatial Extender-Beispielprogramm zu verwenden, können Sie DB2 Administration Client Version 6.0 installieren und verwenden.
- FixPak 1

DB2 Administration Client mit FixPak 1 wird automatisch auf Ihrem System installiert, wenn Sie den DB2 Spatial Extender-Client von der CD-ROM installieren.

Wichtig: Wenn DB2 Universal Database Enterprise Edition Version 7.1 oder DB2 Universal Database Enterprise - Extended Edition Version 7.1 auf dem Client installiert ist, müssen Sie FixPak 1 *vor* DB2 Spatial Extender installieren.

Serversoftware

Für DB2 Spatial Extender-Serverprodukte muss eines der folgenden Serverprodukte auf Ihrem System installiert sein, *bevor* Sie DB2 Spatial Extender installieren:

- DB2 Universal Database Enterprise Edition Version 7.1 mit FixPak 1
- DB2 Universal Database Enterprise - Extended Edition Version 7.1 mit FixPak 1

Wenn Sie die Verwendung der DB2-Steuerzentrale beabsichtigen, erstellen und konfigurieren Sie den DB2-Verwaltungsserver. Weitere Informationen zum Erstellen und Konfigurieren des DB2-Verwaltungsservers finden Sie im Handbuch *IBM DB2 Universal Database Systemverwaltung: Implementierung*.

Anmerkung: Sie können zwar den DB2 Spatial Extender mit DB2 Universal Database Enterprise - Extended Edition verwenden, der räumliche Index kann jedoch nicht über mehrere Knoten hinweg partitioniert werden wie in einer massiven Parallelverarbeitungs-umgebung (MPP).

Erforderlicher Plattenspeicherplatz

In Tabelle 1 wird der erforderliche Plattenspeicherplatz für den Spatial Extender aufgelistet.

Tabelle 1. Erforderlicher Plattenspeicherplatz

Spatial Extender-Produkte	Plattenspeicherplatz
Serverprodukte für DB2 Spatial Extender:	
<ul style="list-style-type: none"> • Code der Spatial Extender-Serverbibliothek, Geocodierer-Bezugsdaten und Dokumentation • Optional und auf separater CD-ROM verfügbar: Geocodierer-Bezugsdaten (USA) <p>Weitere Informationen zur Verwendung der Geocodierer-Bezugsdaten finden Sie unter „DB2 Spatial Extender Geocoder Reference Data“ auf Seite 31.</p>	<p>594 MB Gesamtplattenspeicherplatz</p> <ul style="list-style-type: none"> • 31 MB (Code der Spatial Extender-Serverbibliothek, Geocodierer-Beispielbezugsdaten und Dokumentation) • 563 MB (Geocodierer-Bezugsdaten USA)
Client-Produkte für DB2 Spatial Extender (einschließlich Daten für das Beispielprogramm)	1 MB

Tabelle 1. Erforderlicher Plattenspeicherplatz (Forts.)

Spatial Extender-Produkte	Plattenspeicherplatz
	<p>Anmerkung: Die Berechnungen des erforderlichen Plattenspeicherplatzes in dieser Tabelle gehen von der Annahme aus, dass Sie DB2 Universal Database und DB2 Spatial Extender in einer typischen Windows NT- oder Windows 2000-Installation oder unter AIX mit vorab ausgewählten Komponenten installiert haben. Wenn Sie eine andere Art von Installation für DB2 Spatial Extender oder DB2 Universal Database wählen oder gewählt haben, kann Ihr Plattenspeicherbedarf von den angegebenen Werten abweichen.</p>

DB2 Spatial Extender für Windows NT und Windows 2000 installieren

So installieren Sie den Spatial Extender für Windows NT und Windows 2000:

1. Legen Sie die Spatial Extender-CD-ROM in das Laufwerk ein. Daraufhin wird die DB2-Klickstartleiste, eine Schnittstelle für die Installation des DB2 Spatial Extender, geöffnet.
2. Klicken Sie **Install** an. Nach der Initialisierung des Installationsprogramms wird das Fenster **Select Products** geöffnet.

Hinweise:

- a. Sie können die Installation jederzeit beenden und verlassen, indem Sie **Cancel** anklicken.
- b. Wenn eine Warnung ausgegeben wird, dass DB2 zur Zeit aktiv und durch mehrere Prozesse gesperrt ist, klicken Sie **Yes** an, allerdings *nur* unter der Voraussetzung, dass die Datenbank nicht aktiv ist und keine wichtigen Benutzer mit der Datenbank verbunden sind. DB2 beendet die betreffenden Prozesse ohne Datensicherung. Beenden Sie gesperrte DB2-Prozesse mit Hilfe anderer Verwaltungsfunktionen, wenn Sie die Installation bei aktivem System durchführen.
- c. Wenn Sie den DB2 Spatial Extender in einer eigenständigen Umgebung installieren, wählen Sie unter Windows NT und Windows 2000 **DB2 Administration Client** im Installationsprogramm aus. In der Standardeinstellung des Installationsprogramms ist **DB2 Administration Client** *nicht* vorausgewählt.

3. Wählen Sie aus der Liste die Produkte aus, die Sie installieren wollen:
 - Wenn Sie den DB2 Spatial Extender in einer eigenständigen Umgebung installieren, wählen Sie **DB2 Spatial Extender-Server** und **DB2 Administration Client** aus.
 - Wenn Sie den DB2 Spatial Extender auf einer Serverplattform installieren, wählen Sie **DB2 Spatial Extender-Server** aus.
 - Wenn Sie den DB2 Spatial Extender in einer Client-Umgebung installieren, wählen Sie **DB2 Administration Client** aus.

Bei Auswahl der Option **DB2 Administration Client** werden die Unterkomponenten von DB2 Universal Database, das Beispielprogramm und die Beispieldaten des Spatial Extender und die Steuerzentrale von DB2 Universal Database installiert. Die Option **DB2 Spatial Extender-Server** umfasst den Code der DB2 Spatial Extender-Serverbibliothek, Beispiele für Geocodierer-Bezugsdaten und eine Dokumentation.

Anmerkung: Das Produkt DB2 Spatial Extender-Server umfasst *nur* einen Teil der verfügbaren Geocodierer-Daten. Die vollständigen Geocodierer-Bezugsdaten für die USA befinden sich auf einer separaten CD-ROM, die zum Lieferumfang des DB2 Spatial Extender gehört.

4. Klicken Sie **Next** an. Das Fenster **Select Installation Type** wird geöffnet.
5. Wählen Sie die Installationsart aus. Wenn Sie **Custom** auswählen, wird das Fenster **Select Components** geöffnet. In diesem Fenster können Sie die Komponenten auswählen, die installiert werden sollen. Dies setzt Kenntnisse über die DB2-Komponenten und die Einstellungen voraus.
6. Klicken Sie **Next** an, um das Fenster **Choose Destination Location** auszuwählen.
7. Wählen Sie den Ordner aus, in dem DB2 Spatial Extender installiert werden soll. Klicken Sie **Browse** an, wenn Sie nicht den Standardordner verwenden wollen.

Anmerkung: Wenn DB2 Universal Database bereits auf Ihrem System installiert ist, können Sie in diesem Fenster keine neue Installationsposition auswählen und auch keinen neuen Ordner erstellen. Im Fenster **Choose Destination Location** werden folgende Informationen angezeigt:

- Die Speicherposition des Ordners, in dem DB2 Spatial Extender installiert wird
- Der erforderliche Plattenspeicherplatz für die Installation von DB2 Spatial Extender

8. Klicken Sie **Next** an, um DB2 Spatial Extender zu installieren. Das Fenster **Install Progress** wird geöffnet und zeigt den Verlauf des Installationsprozesses an.

DB2 Spatial Extender für AIX installieren

In diesem Abschnitt werden die Schritte zur Installation von DB2 Spatial Extender für AIX beschrieben. Dabei können Sie das DB2-Installationsprogramm (interaktiv oder nicht überwacht), SMIT (System Management Interface Tool) oder den Befehl **installp** verwenden. Folgende Themen werden in diesem Abschnitt behandelt:

- CD-ROM anhängen
- DB2-Installationsprogramm verwenden
- SMIT oder den Befehl **installp** verwenden
- Exemplarumgebung des DB2 Spatial Extender einrichten

CD-ROM anhängen

Bei den verschiedenen Schritten der Installation von DB2 Spatial Extender unter AIX müssen Sie die CD-ROM anhängen. Jedesmal, wenn in diesem Dokument die Angabe */cdrom* auftaucht, müssen Sie *vor* Beginn des jeweiligen Schritts die hier beschriebene Prozedur zum Anhängen der CD-ROM ausführen. Die folgenden Anweisungen beziehen sich auf Installationen, bei denen das DB2-Installationsprogramm, SMIT und der Befehl **installp** verwendet werden.

So hängen Sie die CD-ROM an:

1. Melden Sie sich als Benutzer mit Root-Berechtigung an.
2. Legen Sie die CD-ROM in das Laufwerk ein.
3. Erstellen Sie mit dem folgenden Befehl ein Verzeichnis zum Anhängen der CD-ROM:

```
mkdir -p /cdrom
```

Dabei ist */cdrom* das Mount-Verzeichnis der CD-ROM.

4. Ordnen Sie mit dem folgenden Befehl ein Dateisystem für die CD-ROM zu.

```
smitty storage
```
5. Wählen Sie **File System** aus.
6. Wählen Sie **Add/Change/Show/Delete File System/** aus.
7. Wählen Sie **CD-ROM File System** aus.
8. Wählen Sie **Add CDROM File System** aus.
9. Wählen Sie **Device Name** aus.

Einheitennamen für CD-ROM-Dateisysteme müssen eindeutig sein. Wenn der Name, den Sie auswählen, bereits einem vorhandenen System zugeordnet ist, löschen Sie das vorhandene CD-ROM-Dateisystem oder verwenden Sie einen anderen Namen für Ihr Verzeichnis.

10. Geben Sie im Dialogfenster den folgenden Mount-Punkt ein:
/cdrom
11. Hängen Sie mit dem folgenden Befehl das CD-ROM-Dateisystem an:
smit mountfs
12. Geben Sie den Namen des Dateisystems ein (z. B. /dev/cd0).
13. Geben Sie den Verzeichnisnamen ein, z. B. (cdrom).
14. Geben Sie den Dateisystemtyp ein, z. B. (cdrfs).
15. Setzen Sie für das angehängte System Lesezugriff (READ-ONLY) auf **Yes**.
16. Melden Sie sich ab.

DB2-Installationsprogramm verwenden

In diesem Abschnitt werden die folgenden Themen behandelt:

- Interaktive Installation
- Nicht überwachte Installation
- Trace-Protokoll oder -Datei generieren

Bei der Installation unter AIX mit dem DB2-Installationsprogramm können Sie die Installation wahlweise interaktiv oder in einer nicht überwachten Umgebung ausführen. Bei einer interaktiven Installation nehmen Sie die Einrichtung und Konfiguration von DB2 Spatial Extender über eine Reihe von Anzeigen vor. Bei einer nicht überwachten Installation geben Sie die Informationen für Einrichtung und Konfiguration in einer Antwortdatei vor, die Sie vor dem Aufrufen des DB2-Installationsprogramms erstellen. Wenn Sie DB2 Spatial Extender auf mehreren Maschinen installieren müssen, können Sie die Datei anpassen und für die Installation auf mehreren Workstations verwenden.

So installieren Sie DB2 Spatial Extender interaktiv:

1. Melden Sie sich an der Ziel-Client-Maschine oder an der Servermaschine als Root an.
2. Legen Sie die CD-ROM in das CD-ROM-Laufwerk ein.
3. Geben Sie `cd /cdrom` ein.
4. Geben Sie `./db2setupein`. Das DB2-Installationsprogramm wird gestartet.
5. Wählen Sie die Produkte aus, die Sie installieren wollen:
 - Wählen Sie **Spatial Extender-Server** und **Spatial Extender-Client** aus, wenn Sie DB2 Spatial Extender in einer eigenständigen Konfiguration installieren.
 - Wählen Sie **DB2 Spatial Extender-Server** aus, wenn Sie DB2 Spatial Extender auf einer Serverplattform installieren.

- Wählen Sie **DB2 Spatial Extender-Client** aus, wenn Sie DB2 Spatial Extender in einer Client-Umgebung installieren.
- Wählen Sie **DB2 Administration Client** aus, wenn Sie eine zusätzliche DB2 Client-Unterstützung wünschen.

Navigieren Sie mit der Tabulatortaste zu der Komponente, die Sie auswählen wollen, und drücken Sie die Eingabetaste. Für jedes ausgewählte Produkt ist ein Anpassungsfenster verfügbar.

6. Wählen Sie die Landessprache für die ausgewählten Komponenten aus.
7. Springen Sie mit der Tabulatortaste zur Taste **OK**, und drücken Sie die Eingabetaste, um DB2 Spatial Extender zu installieren.

So installieren Sie DB2 Spatial Extender ohne Überwachung:

1. Erstellen Sie eine Antwortdatei für die nicht überwachte Installation.
2. Starten Sie die nicht überwachte Installation.

Diese Schritte werden nachfolgend im Detail beschrieben.

Schritt 1. Erstellen Sie eine Antwortdatei für die nicht überwachte Installation.

Anmerkung: Wenn Sie die Standardwerte in der Beispielantwortdatei bestätigen, können Sie Schritt 1 überspringen und mit Schritt 2 fortfahren.

1. Öffnen Sie die Beispielantwortdatei für die Produkte, die Sie installieren wollen. Die Beispielantwortdateien befinden sich im Verzeichnis `/cdrom/db2/install/samples`, wobei `/cdrom` die Speicherposition der installierbaren Version von DB2 Spatial Extender bezeichnet. Es gibt zwei Beispielantwortdateien, eine für den Spatial Extender-Server (`db2gse.rsp`) und eine für den Spatial Extender-Client (`db2gsec.rsp`). Die Beispielantwortdateien enthalten:
 - Für die Installation eindeutige Schlüsselwörter
 - Registrierdatenbankwerte/Einstellungen für Umgebungsvariablen
 - Konfigurationsparameter des Datenbankmanagers
2. Ändern Sie einen Wert in der Antwortdatei, indem Sie ein Element aktivieren. Zur Aktivierung eines Elements:
 - a. Entfernen Sie den Stern (*) links der Tastatur-/Umgebungsvariablen.
 - b. Löschen Sie die aktuelle Einstellung rechts des Werts.
 - c. Geben Sie eine neue Einstellung ein.
 - d. Wenn Sie Änderungen vornehmen, speichern Sie Ihre Datei unter einem neuen Namen, damit die ursprüngliche Beispielantwortdatei

erhalten bleibt. Wenn Sie direkt von der CD-ROM installieren, müssen Sie die umbenannte Antwortdatei in einem lokalen Dateisystem speichern.

Schritt 2. Nicht überwachte Installation mit einer Antwortdatei starten.

1. Melden Sie sich als Benutzer mit Root-Berechtigung an.
2. Geben Sie den Befehl **db2setup** ein:

```
/cdrom/db2setup -r antwortdatei_verzeichnis/antwortdatei_datei
```

Dabei ist */cdrom* die Speicherposition des installierbaren Abbilds von DB2 Spatial Extender, *antwortdatei_verzeichnis* das Verzeichnis, in dem sich die Antwortdatei befindet, und *antwortdatei_datei* der Name der Antwortdatei.

3. Überprüfen Sie nach Abschluss der Installation die Nachrichten in der Protokolldatei. Die Standardspeicherposition der Protokolldatei ist */tmp/db2setup.log*.

Trace-Protokolldatei für das DB2-Installationsprogramm generieren: Wenn bei der Ausführung des DB-Installationsprogramms Probleme auftreten, können Sie eine Trace-Protokolldatei (*db2setup.trc*) generieren. Die Trace-Protokolldatei können Sie zusammen mit der Datei *db2setup.log* zur weiteren Diagnose an die IBM Softwareunterstützungsfunktion senden. Beide Dateien werden im Verzeichnis */tmp* generiert.

Führen Sie zum Generieren einer Trace-Protokolldatei wie folgt den Befehl **db2setup** mit der Markierung **-d** aus: *./db2setup -d*. Hierauf wird das DB2-Installationsprogramm im Trace-Modus ausgeführt. Arbeiten Sie weiter mit der Schnittstelle, um das aufgetretene Problem zu reproduzieren. Wenn Sie Ihre Arbeit beenden, wird die Trace-Protokolldatei */tmp/db2setup.trc* erstellt.

SMIT oder den Befehl *installp* verwenden

So installieren Sie DB2 Spatial Extender mit SMIT (System Management Interface Tool) oder mit dem Befehl **installp**:

1. Melden Sie sich an der Ziel-Client-Maschine oder an der Servermaschine als Root an.
2. Legen Sie die CD-ROM in das CD-ROM-Laufwerk ein.
3. Installieren Sie DB2 Spatial Extender, indem Sie SMIT oder den Befehl **installp** ausführen.

So führen Sie SMIT aus:

- a. Geben Sie den Befehl **smit install_latest** ein. Das SMIT-Tool-Menü wird geöffnet.
- b. Geben Sie */cdrom/db2* als Eingabeeinheit/Eingabeverzeichnis für die Software ein.
- c. Klicken Sie **DO** an oder drücken Sie die Eingabetaste, um zu überprüfen, ob das Installationsverzeichnis existiert.

- d. Geben Sie im Feld **Software to install** an, ob die Client- oder Serverkomponenten installiert werden sollen.
Geben Sie für Client Folgendes ein: `db2_07_01.gc1n`.
Geben Sie für Server Folgendes ein: `db2_07_01.gsrv`.
- e. Klicken Sie auf **DO** oder drücken Sie die Eingabetaste (Sie werden aufgefordert, die Installationsparameter zu bestätigen).
- f. Bestätigen Sie mit der Eingabetaste.
Die Produktdateien werden von der CD-ROM auf Ihrer Festplatte installiert. Dies kann einige Minuten dauern.
- g. Melden Sie sich ab.

Exemplarumgebung des DB2 Spatial Extender einrichten

Der Befehl `db2icrt` dient zur Erstellung neuer DB2-Exemplare. Bei allen neuen DB2-Exemplaren, die Sie nach der Installation von DB2 Spatial Extender erstellen, ist DB2 Spatial Extender Bestandteil der Exemplarumgebung.

Bei DB2-Exemplaren, die Sie vor der Installation von Spatial Extender erstellt haben, ist DB2 Spatial Extender nicht Bestandteil der Exemplarumgebung. Mit dem Befehl `db2iupdt` können Sie vorhandene DB2-Exemplare mit Spatial Extender aktualisieren.

Geben Sie als Benutzer mit Root-Berechtigung den folgenden Befehl ein:

```
db2iupdt exemplarname
```

Der Parameter *exemplarname* ist der Name des Exemplars.

Unter AIX befindet sich dieses Dienstprogramm im Verzeichnis `/usr/lpp/db2_07_01/instance`. Wenn Sie Hilfe benötigen, geben Sie `db2iupdt -h` in der Befehlszeile ein, um ein Hilfemenü zu öffnen.

Anmerkung: Nehmen Sie die Aktualisierung der Exemplarumgebung mit Spatial Extender vor, bevor Sie die Installation überprüfen.

Installation überprüfen

Erstellen Sie nach der Installation von DB2 Spatial Extender eine Datenbank und führen Sie das Installationsprüfprogramm aus, um sicherzustellen, dass DB2 Spatial Extender korrekt installiert und konfiguriert wurde.

Anmerkung: Überprüfen Sie bei AIX-Installationen, ob Sie die Exemplarumgebung von DB2 Spatial Extender eingerichtet haben, bevor Sie das Installationsprüfprogramm ausführen.

Sie können die Installation anhand des DB2 Spatial Extender-Beispielprogramms (runGseDemo) überprüfen. Datenbankkonfigurationsparameter können in der Befehlszeile mit DB2-Tools oder über die Benutzerschnittstelle in der DB2-Steuerzentrale geändert werden. Die folgenden Anweisungen beziehen sich auf AIX, Windows NT und Windows 2000.

So überprüfen Sie die Installation:

1. Melden Sie sich als Exemplareigner an (nur AIX).
2. Erhöhen Sie in der Datenbankmanagerkonfiguration die UDF-Speichergröße auf einen Mindestwert von 2048. Geben Sie z. B. `db2 update dbm cfg using UDF_MEM_SZ 2048` ein. Wenn 2048 nicht geeignet ist, erhöhen Sie den Wert des Parameters `UDF_MEM_SZ` in Schritten zu 256.

Anmerkung: Die erforderliche UDF-Speichergröße steigt mit der wachsenden Zahl der UDFs, auf die in einer Anwendung verwiesen wird. Dies gilt insbesondere, wenn räumliche UDFs mit räumlichen Datentypen als Eingabe- und/oder Ausgabe-parameter verwendet werden.

3. Erstellen Sie eine Datenbank. Geben Sie z. B. `db2 create database meinedb` ein, wobei *meinedb* der Datenbankname ist.
4. Erhöhen Sie die Größe der DB2-Protokolldatei für Ihre Datenbank.

Gehen Sie dazu folgendermaßen vor:

- a. Stellen Sie eine Verbindung zu der erstellten Datenbank her. Geben Sie z. B. `db2 connect to meinedb` ein, wobei *meinedb* der Datenbankname ist.
- b. Erhöhen Sie die Größe der Protokolldatei. Geben Sie z. B. `db2 update db logfilesize using LOGFILE 1000` ein.
- c. Trennen Sie die Verbindung zur Datenbank. Geben Sie z. B. `db2 connect reset` ein.

Anmerkung: Die Größe der DB2-Protokolldatei muss jedesmal erhöht werden, wenn eine Datenbank für räumliche Operationen aktiviert wird.

5. Suchen Sie das Installationsprüfprogramm. Geben Sie z. B. `runGseDemo` ein. Geben Sie unter AIX `cd $HOME/sql1lib/samples/spatial` ein, wobei *\$HOME* das Ausgangsverzeichnis des Exemplareigners ist. Geben Sie unter Windows NT und Windows 2000 `cd c:\sql1lib\samples\spatial` ein, wobei *c:\sql1lib* das Installationsverzeichnis von DB2 Spatial Extender ist.
6. Führen Sie das Installationsprüfprogramm aus. Geben Sie z. B. Folgendes ein:
`runGseDemo meinedb benutzerID kennwort`

Der Parameter *meinedb* ist der Datenbankname.

Hinweise zur Fehlerbehebung für das Beispielprogramm

Das DB2-Beispielprogramm hat die Aufgabe, Installationsfehler zu ermitteln. Bei der Überprüfung der Installation erhalten Sie u. U. Fehlernachrichten, die Ihnen bei der Diagnose spezifischer Systemprobleme helfen können. Die meisten Fehlernachrichten werden durch einige typische Benutzerfehler ausgelöst. Führen Sie zur Vermeidung dieser Fehler bei jeder Ausführung des Installationsprüfprogramms die folgenden Schritte aus:

- Stellen Sie sicher, dass Sie den DB2 Spatial Extender-Client und den DB2 Spatial Extender-Server in den entsprechenden Umgebungen installiert haben. Bei einer eigenständigen Konfiguration muss sowohl die Client- als auch die Serverkomponente installiert sein.
- Verwenden Sie eine neue Datenbank, der keine räumlichen Operationen zugeordnet sind.
- Erhöhen Sie in der Datenbankmanagerkonfiguration die UDF-Speichergröße.
- Erhöhen Sie die Größe der Protokolldatei.

Administration Client

Wenn Sie bei der Installation von DB2 Spatial Extender weder die Option **DB2 Administration Client** (bei NT-Installationen) noch die Option **Spatial Extender Client** (bei AIX-Installationen) ausgewählt haben, wird die folgende Fehlernachricht ausgegeben: "The name specified is not recognized as an internal or external command, operable program or batch file."

Dieser Fehler ist darauf zurückzuführen, dass das Beispielprogramm nicht im System vorhanden ist. Das Beispielprogramm wird mit dem DB2 Administration Client und dem Spatial Extender-Client ausgeliefert. Wenn der DB2 Administration Client oder der Spatial Extender-Client nicht auf Ihrem System installiert ist, ist auch das Beispielprogramm nicht vorhanden.

Gehen Sie folgendermaßen vor, um dieses Problem zu beheben:

1. Installieren Sie DB2 Spatial Extender erneut. Wählen Sie unter Windows NT oder Windows 2000 im Installationsprogramm **DB2 Administration Client** bzw. unter AIX im DB2-Installationsprogramm **Spatial Extender-Client** aus.
2. Führen Sie das Beispielprogramm erneut aus, indem Sie die Schritte unter „Installation überprüfen“ auf Seite 26 wiederholen.

Die Datenbank ist bereits für räumliche Operationen aktiviert

Wenn die Datenbank, auf die das Beispielprogramm zugreift, bereits für räumliche Operationen aktiviert ist, erhalten Sie die folgende Fehlernachricht:

```
Enabling database logst...
Returning from ENABLE_DB:
Return code = -14
Return message text =
GSE0014E The database has already been enabled for spatial operations.
```

Zur Behebung dieses Fehlers löschen Sie die Datenbank und wiederholen die Schritte unter „Installation überprüfen“ auf Seite 26.

Anmerkung: Stellen Sie sicher, dass die Datenbank, für die Sie die Installation überprüfen, neu ist und dass ihr keine räumlichen Operationen zugeordnet sind. Wenn dies nämlich der Fall ist, kann das Programm nicht ausgeführt werden.

Datenbankmanagerkonfiguration

Wenn Sie die UDF-Speichergröße in der Datenbankmanagerkonfiguration nicht erhöht haben, erhalten Sie die folgende Fehlermeldung:

```
An unexpected SQL error ("SQL0973N Not enough storage is available
in the "UDF_MEM" heap to process the statement. ") has occurred.
SQLSTATE=57011
```

Anweisungen zur Erhöhung der Größe in der Datenbankmanagerkonfiguration finden Sie in Schritt 2 auf Seite 27 unter „Installation überprüfen“ auf Seite 26.

Größe der Protokolldatei

Wenn Sie die Größe der Protokolldatei nicht erhöht haben, erhalten Sie die folgende Fehlermeldung:

```
Enabling database logst...
Returning from EN
ABLE_DB:
Return code = -8
Return message text =
GSE0008E An unexpected SQL error ("SQL3306N An SQL error "-964"
occurred while inserting a row into ") has occurred.
```

Anweisungen zur Erhöhung der Größe der Protokolldatei finden Sie in Schritt 4 auf Seite 27 unter „Installation überprüfen“ auf Seite 26.

Überlegungen nach Abschluss der Installation

Nach Abschluss der Installation des Spatial Extender sollten Sie Folgendes in Betracht ziehen:

- Herunterladen von ArcExplorer Java Version 3.0
- CD-ROMs mit Bezugsdaten, Daten und Karten für den DB2 Spatial Extender-Geocodierer verwenden

Herunterladen von ArcExplorer

ESRI (Environmental Systems Research Institute) bietet einen Browser an, der visuelle Abfrageergebnisse für DB2 Spatial Extender-Daten erzeugen kann. Bei diesem Browser handelt es sich um ArcExplorer Java Version 3.0. Eine Kopie des ArcExplorer Java Version 3.0 können Sie von der ESRI-Website unter <http://www.esri.com> herunterladen. Voraussetzung für den ArcExplorer ist Standard Edition oder Enterprise Edition Java[®] 2 Runtime Environment (JRE), Version 1.2.2.

Weitere Informationen zum Installieren und Verwenden von ArcExplorer Java Version 3.0 finden Sie im Handbuch *Using ArcExplorer*, das Sie ebenfalls über die ESRI-Website abrufen können.

Wichtig: DB2 Universal Database Version 7.1 wird mit IBM Java Development Kit (JDK) Version 1.1.8 ausgeliefert. Wenn Sie JRE Version 1.2.2 für ArcExplorer installieren, verwenden Sie dafür ein anderes Verzeichnis als für DB2. Denken Sie daran, die Umgebungsvariable CLASS-PATH einzustellen.

CD-ROMs mit Bezugsdaten, Daten und Karten für den DB2 Spatial Extender-Geocodierer verwenden

Zum Lieferumfang von DB2 Spatial Extender gehören fünf CD-ROMs mit Daten und Karten sowie eine CD-ROM mit Geocodierer-Daten.

DB2 Spatial Extender Data and Maps

Die Daten und Karten mit der Bezeichnung "DB2 Spatial Extender Data and Maps 1 – 5" werden auf fünf CD-ROMs bereitgestellt. Tabelle 2 enthält eine Übersicht über die Daten, die sich auf den einzelnen CD-ROMs befinden.

Tabelle 2. CD-ROMs mit Daten und Karten

CD-ROMs mit Daten und Karten	Art der Kartendaten
CD-ROM 1	Kanada, Europa, Mexiko, USA und Welt
CD-ROM 2	USA (detailliert)
CD-ROM 3	USA (Westen)
CD-ROM 4	USA (Osten)
CD-ROM 5	USA (Süden) und Beispielabbildaten

Eine detaillierte Beschreibung der von ESRI bereitgestellten Daten finden Sie in der ESRI-Hilfedatei auf der CD-ROM DB2 Spatial Extender Data and Maps.

- Unter Windows NT und Windows 2000 können Sie die Hilfedatei unter *x*: esridata.hlp anzeigen. Dabei ist *x*: das CD-ROM-Laufwerk.
- Unter AIX können Sie die Hilfedatei auf der CD-ROM unter */cdrom/esridata.hlp* anzeigen oder drucken. Dabei ist */cdrom* Ihr Mount-Punkt.

DB2 Spatial Extender Geocoder Reference Data

Die Geocoderer-Bezugsdaten auf der CD-ROM "DB2 Spatial Extender Geocoder Reference Data" wurden speziell für den Standard-Geocoder von DB2 Spatial Extender konzipiert. Sie umfassen Basiskarten mit Straßennetzdaten für die USA, die der Geocoder verwendet, um Breiten- und Längengrad von Adressen in einer für räumliche Operationen aktivierten Datenbank zu ermitteln. Diese Basiskartendaten werden allgemein als "Bezugsdaten" bezeichnet. Der Standard-Geocoder vergleicht die (nicht räumlichen) Adressdaten in Ihrer Datenbank mit den Bezugsdaten, gleicht sie ab und setzt die Adressdaten in Koordinaten um, die DB2 Spatial Extender speichern kann. Dieser Prozess wird als *Geocodierung* bezeichnet.

Weitere Informationen zur Geocodierung finden Sie unter „Geocodierer verwenden“ auf Seite 53.

Auf Geocodierer-Bezugsdaten zugreifen: Sie können entweder direkt auf die Geocodierer-Daten auf der CD-ROM zugreifen oder die Daten auf Ihr Festplattenlaufwerk kopieren. Führen Sie die nachfolgend beschriebenen Schritte aus, um Geocodierer-Dateien von der CD-ROM in Ihre DB2 Spatial Extender-Serverumgebung zu kopieren.

Unter AIX:

1. Hängen Sie die CD-ROM an. Anweisungen zum Anhängen einer CD-ROM finden Sie unter „CD-ROM anhängen“ auf Seite 22.
2. Melden Sie sich an der Zielservermaschine als Benutzer mit Root-Berechtigung an.
3. Geben Sie Folgendes ein:

```
cp /cdrom/db2/* /usr/lpp/db2_07_01/gse/refdata/
```
4. Melden Sie sich ab.

Unter Windows NT und Windows 2000 können Sie entweder das Befehlsfenster oder den Windows-Explorer verwenden.

So greifen Sie über das Befehlsfenster auf Geocodierer-Daten zu:

1. Klicken Sie **Start** -> **Programme** -> **IBM DB2** -> **Command Window** an.
2. Geben Sie Folgendes ein:

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

Ersetzen Sie *d:* durch den Laufwerksbuchstaben Ihres CD-ROM-Laufwerks.

So greifen Sie über den Windows-Explorer auf Geocodierer-Daten zu:

Kopieren Sie alle Dateien aus *d:\db2* to *c:\sqllib\gse\refdata*, wobei *d:* das CD-ROM-Laufwerk und *c:\sqllib* das Installationsverzeichnis von DB2 ist.

Datei EDGELocator.loc für den Standard-Geocodierer verfügbar machen:

Zu den Bezugsdaten auf der CD-ROM gehört die Datei EDGELocator.loc. Anhand der Datei EDGELocator.loc lokalisiert der Standard-Geocodierer bestimmte Bezugsdaten. Wenn Sie z. B. Adressen in Kalifornien, Kentucky und Oregon geocodieren, ermittelt der Standard-Geocodierer anhand der Querverweisdatei die Speicherposition der Adressen auf der CD-ROM.

Wenn Sie die schrittweise steigende Geocodierung (auch automatische Geocodierung genannt) für den Standard-Geocodierer aktivieren oder den Standard-Geocodierer im Stapelbetrieb ausführen, müssen Sie den Eingabeparameter **vendorSpecific** verwenden. Wenn Sie den Eingabeparameter **vendorSpecific** angeben, müssen der Verzeichnispfad und der Dateiname der Querverweisdatei an den Geocodierer übermittelt werden.

Mit dem folgenden **gseadm**-Beispielbefehl wird beispielsweise die Geocodierung im Stapelbetrieb mit dem Standard-Geocodierer aufgerufen:

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

Weitere Informationen über die Ausführung des Geocodierers und die Verwendung des Parameters **vendorSpecific** finden Sie in „Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren“ auf Seite 45 und „Kapitel 5. Räumliche Spalten ausfüllen“ auf Seite 53.

Spatial Extender aufrufen

Nach der Installation von Spatial Extender können Sie mit der DB2-Steuerzentrale die GIS-Umgebung einrichten und die Arbeit mit räumlichen Informationen beginnen.

So rufen Sie den Spatial Extender über die DB2-Steuerzentrale auf:

1. Klicken Sie im Fenster **Control Center** den Server an, auf dem der Spatial Extender ausgeführt werden soll.
2. Klicken Sie den Ordner **Databases** an. Die Datenbanken werden im Inhalts-Teilfenster angezeigt.
3. Klicken Sie mit der rechten Maustaste die Datenbank an, mit der Sie arbeiten wollen, und klicken Sie dann die auszuführende räumliche Operation im Kontextmenü an.

Kapitel 3. Ressourcen einrichten

Nach der Installation des Spatial Extender können Sie damit beginnen, in Ihre Datenbank Ressourcen einzutragen, die Sie beim Erstellen räumlicher Spalten und beim Bearbeiten räumlicher Daten benötigen. Dieses Kapitel fasst diese Ressourcen zusammen und beschreibt zwei der Aufgaben, über die Sie die Ressourcen verfügbar machen: Aktivieren der Datenbank für räumliche Operationen und Erstellen räumlicher Bezugssysteme.

Bestand der Ressourcen

Die Ressourcen, die Sie zum Erstellen räumlicher Spalten und zum Bearbeiten räumlicher Daten nutzen:

- *Bezugsdaten*: Adressen, die der Spatial Extender überprüft, um die Adressen für die Geocodierung zu überprüfen
- Ressourcen, die eine Datenbank für räumliche Operationen aktivieren: gespeicherte Prozeduren, räumliche Funktionen und andere
- Geocodierer, die von Benutzern und anderen Anbietern bereitgestellt werden (nicht der Standardcodierer)
- Räumliche Bezugssysteme

Dieser Abschnitt beschreibt Bezugsdaten und Ressourcen, die eine Datenbank für räumliche Operationen aktivieren. Informationen zu Geocodierern über den Standardcodierer hinaus finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53. Informationen zu räumlichen Bezugssystemen finden Sie im Abschnitt „Informationen zu Koordinaten und räumlichen Bezugssystemen“ auf Seite 35.

Bezugsdaten

Bezugsdaten bestehen aus den neuesten Adressen in den USA, die vom United States Census Bureau erfasst wurden. Bevor der Standard-Geocodierer eine Adresse in Ihrer Datenbank in Koordinaten umsetzen kann, muss er zunächst einen Teil oder alle Adressen mit den Adressen in den Bezugsdaten abgleichen.

Bezugsdaten werden verfügbar, wenn Sie den Spatial Extender installieren. Die Größe des für diese Daten erforderlichen Plattenspeicherplatzes ist im Abschnitt „Erforderlicher Plattenspeicherplatz“ auf Seite 19 beschrieben. Zur Überprüfung, ob die Daten unter AIX richtig geladen wurden, suchen Sie sie im Verzeichnis `$DB2INSTANCE/sqlib/gse/refdata/`. Zur Überprüfung, ob die Daten unter Windows NT richtig geladen wurden, suchen Sie sie im Verzeichnis `%DB2PATH%\gse\refdata\`.

Ressourcen, die eine Datenbank für räumliche Operationen aktivieren

Die erste Aufgabe, die Sie nach der Installation des Spatial Extender ausführen, ist das Aktivieren Ihrer Datenbank für räumliche Operationen. Hierzu gehört das Aufrufen einer Aktion, die bewirkt, dass der Spatial Extender die Datenbank mit den folgenden Ressourcen lädt:

- Gespeicherte Prozeduren. Wenn Sie eine Aktion über die Steuerzentrale anfordern, ruft der Spatial Extender zur Ausführung der Aktion eine dieser gespeicherten Prozeduren auf.
- Typen räumlicher Daten. Sie müssen jeder Tabelle oder Ansichtsspalte, in der räumliche Daten gespeichert werden sollen, einen Typ von räumlichen Daten zuordnen. Weitere Informationen hierzu finden Sie im Abschnitt „Informationen zu Typen von räumlichen Daten“ auf Seite 45.
- Katalogtabellen und Sichten des Spatial Extender. Bestimmte Operationen hängen von dem Spatial Extender-Katalog ab. Bevor eine Spalte mit einem Typ von räumlichen Daten ausgefüllt werden kann, muss sie beispielsweise im Katalog als Schicht registriert werden. Informationen zu Schichten finden Sie im Abschnitt „Ein GIS-Projekt entwickeln und implementieren“ auf Seite 11.
- Ein Typ eines räumlichen Index. Er ermöglicht Ihnen das Definieren von Indizes für Schichten.
- Räumliche Funktionen. Sie verwenden diese Funktionen zum Arbeiten mit räumlichen Daten auf verschiedene Arten, beispielsweise zum Ermitteln der Beziehung zwischen geografischen Merkmalen und zum Generieren weiterer räumlicher Daten. Eine dieser Funktionen ist ein Standard-Geocodierer. Er wandelt Adressen in den USA in Koordinaten um und fügt diese Koordinaten anschließend in räumliche Spalten ein. Weitere Informationen zu räumlichen Funktionen finden Sie in „Kapitel 13. Geometrien und zugeordnete räumliche Funktionen“ auf Seite 157 und „Kapitel 14. Räumliche Funktionen für SQL-Abfragen“ auf Seite 197. Weitere Informationen zum Standard-Geocodierer finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53.
- Ein Schema mit dem Namen DB2GSE, das die eben aufgelisteten Objekte enthält.

Anweisungen zur Verwendung der Steuerzentrale zum Initiieren des Aufrufs dieser Ressourcen finden Sie im Abschnitt „Eine Datenbank für räumliche Operationen aktivieren“ auf Seite 35. Richtlinien zur Verwendung einer Routine in einem Anwendungsprogramm zur Ausführung der gleichen Aufgabe finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Eine Datenbank für räumliche Operationen aktivieren

Informationen darüber, welche Berechtigungen zum Aktivieren einer Datenbank für räumliche Operationen erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 95.

So aktivieren Sie eine Datenbank für räumliche Operationen über die Steuerzentrale:

1. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Databases** unter dem Server finden, auf dem der Spatial Extender ausgeführt werden soll.
2. Klicken Sie den Ordner **Databases** an. Die Datenbanken werden im Inhalts-Teilfenster auf der rechten Seite des Fensters angezeigt.
3. Klicken Sie mit der rechten Maustaste die Datenbank an, mit der Sie arbeiten wollen, und klicken Sie **Spatial Extender** → **Enable** im Kontextmenü an. Spatial Extender stellt in der Datenbank die Ressourcen zur Verfügung, die Ihnen das Erstellen und Arbeiten mit räumlichen Spalten und Daten ermöglichen.

Erinnerung: Bevor Sie eine Datenbank für räumliche Operationen aktivieren können, muss Spatial Extender auf dem Server installiert sein, auf dem sich die Datenbank befindet.

Ein räumliches Bezugssystem erstellen

Dieser Abschnitt beschreibt die Beziehung zwischen räumlichen Bezugssystemen und Koordinatensystemen. Außerdem wird in diesem Abschnitt erläutert, wie ein räumliches Bezugssystem von der Steuerzentrale aus erstellt wird.

Informationen zu Koordinaten und räumlichen Bezugssystemen

Dieser Abschnitt setzt die im Abschnitt „Das Wesen der räumlichen Daten“ auf Seite 6 begonnene Beschreibung von Koordinatensystemen fort. Anschließend wird die Definition räumlicher Bezugssysteme aus Abschnitt „Ein GIS-Projekt entwickeln und implementieren“ auf Seite 11 erweitert. Der Abschnitt schließt mit Richtlinien für die Festlegung, welche Werte den Parametern eines räumlichen Bezugssystems zugeordnet werden sollen.

Koordinatensysteme, Koordinaten und Maße

Sie können sich ein Koordinatensystem wie ein imaginäres Gitter vorstellen, das einen bestimmten geografischen Bereich abdeckt. Beispiele hierfür sind ein Gitter über die gesamte Erde, über einen Staat oder über eine Region eines Landes. Jedes geografische Merkmal in diesem Bereich liegt im Schnittbereich einer Ost-West-Gitterlinie und einer Nord-Süd-Gitterlinie. Ein Wert, die *X-Koordinate*, gibt an, wo der Standort auf der Ost-West-Gitterlinie liegt. Ein

anderer Wert, eine *Y-Koordinate*, gibt an, wo der Standort auf der Nord-Süd-Gitterlinie liegt. Beide Werte stellen eine Beziehung des Standorts zum Mittelpunkt des Gitters, dem *Ursprung*, her.

Die X- und Y-Koordinaten sind am Ursprung beide Null. Vom Ursprung aus nach Osten sind die X-Koordinaten positiv, nach Westen hin sind sie negativ. Ebenso sind die Y-Koordinaten in nördlicher Richtung vom Ursprung aus positiv, in südlicher Richtung dagegen negativ. Eine Illustration dieser Verteilung bietet das folgende allgemeine Beispiel: Koordinatensystem A enthält ein Gitter, das ein großes Stadtgebiet abdeckt. Eine X-Koordinate von 7 kennzeichnet beispielsweise eine Position, die sieben Einheiten östlich vom Ursprung dieses Gitters liegt. Eine X-Koordinate -9.5 kennzeichnet zum Beispiel eine Position, die neun und eine halbe Einheit westlich vom Ursprung dieses Gitters liegt.

Jedes Datenelement in einer räumlichen Spalte enthält entweder (1) eine X-Koordinate und eine Y-Koordinate, die den Standort eines geografischen Merkmals definieren, oder (2) mehrere X- und Y-Koordinaten, die die Standorte von Teilen eines Merkmals definieren oder das von einem Merkmal abgedeckte Gebiet. Zwei weitere Arten von Werten — eine *Z-Koordinate* und ein *Maß* — können ebenfalls angegeben werden. Im Gegensatz zu X- und Y-Koordinaten werden Z-Koordinaten und Maß in Spatial Extender nicht zur Definition von Standorten oder Gebieten verwendet. Statt dessen enthalten sie Informationen, die von einer GIS-Anwendung benötigt werden. Eine Z-Koordinate gibt normalerweise die Höhe oder Tiefe eines geografischen Merkmals an. Z-Koordinaten über dem Ursprung sind positiv, darunter sind sie negativ. Ein Maß ist numerisch; es kann beliebige Arten von Informationen enthalten. Angenommen, Sie stellen in Ihrem GIS Ölquellen dar. Wenn Ihre Anwendungen Werte verarbeiten sollen, die Bohrstellen-IDs für seismische Daten kennzeichnen, können diese Daten als Maß gespeichert werden.

Räumliche Bezugssysteme, Abstände und Maßstabsfaktoren

Wie in „Koordinatensysteme, Koordinaten und Maße“ auf Seite 35 beschrieben, können Koordinaten negativ sein und werden als Dezimalzahlen ausgedrückt. Dasselbe gilt für Maße. Zur Reduzierung des Speicheraufwands speichert Spatial Extender jede Koordinate und jedes Maß als nicht negative ganze Zahl (also als positive ganze Zahl oder als Null). Tatsächliche negative und dezimale Koordinaten und Maße müssen daher in nicht negative ganze Zahlen umgewandelt werden, damit der Spatial Extender sie speichern kann. Darüber hinaus müssen Sie Spatial Extender mitteilen, wie die Umwandlung erfolgen soll. Hierzu legen Sie bestimmte Parameter fest. Die zur Umwandlung von Koordinaten und Maßen innerhalb eines bestimmten geografischen Bereichs erforderlichen Parametereinstellungen werden zusammen als *räumliches Bezugssystem* bezeichnet.

Sie können ein räumliches Bezugssystem wie folgt erstellen:

- Ermitteln Sie zunächst die niedrigsten negativen Koordinaten und Maße für die dargestellten Merkmale. (Je weiter ein negativer Wert von Null entfernt ist, desto niedriger ist er. Eine X-Koordinate von -10 ist niedriger als eine X-Koordinate von -5 ; ein Maß von -100 ist niedriger als ein Maß von -50 .)
- Geben Sie dann *Abstandsfaktoren* (kurz: *Abstände*) an: Werte, die bei Subtraktion von negativen Koordinaten und Maßen nicht negative Zahlen ergeben.
- Geben Sie anschließend *Maßstabsfaktoren* an: Werte, die bei Multiplikation mit dezimalen Koordinaten und Maßen ganze Zahlen ergeben, deren Genauigkeit mindestens der Genauigkeit der Koordinaten oder Maße entspricht. Betrachten Sie beispielsweise eine Koordinate mit der Genauigkeit vier: 92.77 . Sie können diese Koordinate mit dem Maßstabsfaktor 100 multiplizieren, um eine ganze Zahl mit der Genauigkeit vier zu erhalten: 9277 . Beachten Sie, dass bei der Erstellung eines räumlichen Bezugssystems Abstandsfaktoren vor Maßstabsfaktoren angewandt werden.

Die niedrigsten negativen Koordinaten und Maße ermitteln

DB2 Spatial Extender kann Koordinaten und Maße speichern, wenn diese positive ganze Zahlen sind, nicht aber, wenn sie negative Zahlen oder Dezimalzahlen sind. Aus diesem Grund müssen negative Koordinaten und Maße in positive Zahlen, dezimale Koordinaten und Maße in ganze Zahlen umgewandelt werden. Zur Durchführung dieser Umwandlung definieren Sie eine Gruppe von Parametern, die bei Anwendung auf negative oder dezimale Koordinaten und Maße positive ganze Zahlen ergeben. Diese Gruppe von Parametern wird als *räumliches Bezugssystem* bezeichnet. Die Parameter zur Umwandlung negativer Werte werden als *Abstandsfaktoren*, die Parameter zur Umwandlung von Dezimalwerten als *Maßstabsfaktoren* bezeichnet.

Wenn Sie eine räumliche Funktion aufrufen, die eine dezimale Koordinate, ein dezimales Maß und die Kennung eines räumlichen Bezugssystems als Eingabe akzeptiert, multipliziert die Funktion die dezimale Koordinate bzw. das dezimale Maß mit einem Maßstabsfaktor innerhalb des Systems. Das Ergebnis ist eine ganze Zahl, die von DB2 Spatial Extender gespeichert wird. Der Maßstabsfaktor muss so groß gewählt werden, dass die Genauigkeit dieser ganzen Zahl mit der Genauigkeit der dezimalen Koordinate übereinstimmt.

Gehen Sie z. B. davon aus, dass eine Eingabe für die Funktion `ST_Point` die X-Koordinate 10.01 , die Y-Koordinate 20.03 und die Kennung eines räumlichen Bezugssystems umfasst. Wenn die Funktion `ST_Point` aufgerufen wird, multipliziert sie den Wert 10.01 und den Wert 20.03 mit dem Maßstabsfaktor des räumlichen Bezugssystems für X- und Y-Koordinaten. Wenn dieser Maßstabsfaktor 10 ist, lauten die resultierenden ganzen Zahlen, die DB2 Spatial Extender speichert, 100 bzw. 200 .

Da die Genauigkeit dieser ganzen Zahlen (3) niedriger ist als die der Koordinaten (4), kann DB2 Spatial Extender diese ganzen Zahlen nicht wieder in die ursprünglichen Koordinaten umwandeln und keine Werte von ihnen ableiten, die mit dem Koordinatensystem konsistent sind, zu dem diese Koordinaten gehören. Ist der Maßstabsfaktor jedoch 100, speichert DB2 Spatial Extender die ganzen Zahlen 1001 und 2003 — Werte, die wieder in die ursprünglichen Koordinaten umgewandelt und von denen kompatible Koordinaten abgeleitet werden können.

Bevor Sie die Parameter für ein räumliches Bezugssystem festlegen, müssen Sie die niedrigste negative X-, Y- und Z-Koordinate sowie das Maß in dem geografischen Bereich, der die entsprechenden Merkmale enthält, ermitteln. Sie können diese Werte durch Beantworten der folgenden Fragen ermitteln:

- Liegen irgendwelche der dargestellten Merkmale westlich vom Ursprung des verwendeten Koordinatensystems? Falls ja, welche X-Koordinaten geben die Position des westlichen Rands des westlichsten Merkmals an? (Die Antwort gibt die niedrigste negative X-Koordinate an, die Sie hier behandeln.) Wenn Sie beispielsweise Ölquellen darstellen und einige dieser Quellen westlich des Ursprungs liegen, welche X-Koordinate gibt die Position der westlichsten Ölquelle an?
- Liegen irgendwelche der Merkmale südlich des Ursprungs? Falls ja, welche Y-Koordinaten geben die Position des südlichsten Rands des südlichsten Merkmals an? (Die Antwort gibt die niedrigste negative Y-Koordinate an, die Sie hier behandeln.) Wenn Sie beispielsweise Ölquellen darstellen und einige dieser Quellen südlich des Ursprungs liegen, welche X-Koordinate gibt die Position der südlichsten Ölquelle an?
- Wenn Sie die Tiefe mit Z-Koordinaten definieren wollen, welches Merkmal ist das tiefste, und welche Z-Koordinate kennzeichnet den tiefsten Punkt dieses Merkmals? (Die Antwort gibt die niedrigste negative Z-Koordinate an, die Sie hier behandeln.)
- Wenn Sie Maße in Ihre räumlichen Daten einbeziehen wollen, können dabei auch negative Maße vorkommen? Falls ja, welches ist der niedrigste Wert für ein negatives Maß?

Nachdem Sie die niedrigsten negativen Koordinaten bestätigt haben, addieren Sie dazu einen Wert von fünf bis zehn Prozent. Wenn die niedrigste negative X-Koordinate beispielsweise -100 ist, könnten Sie -5 addieren. Die so erhaltene Zahl wird in diesem Handbuch als *erweiterter Wert* bezeichnet.

Anmerkung: Die Kennung des räumlichen Standardbezugssystems von DB2 Spatial Extender ist 0 (null). DB2 Spatial Extender umfasst ein räumliches Bezugssystem zur Verwendung mit dem Standard-Geocoder. Die Kennung dieses Systems ist 1.

Abstandsfaktoren angeben

Geben Sie als nächstes an, welche Abstandsfaktoren der Spatial Extender zur Umwandlung negativer Koordinaten und Maße in nicht negative Zahlen verwenden soll:

- Nachdem Sie festgelegt haben, was Ihr erweiterter X-Wert sein soll, geben Sie einen Abstand an, der nach Subtraktion von diesem Wert Null ergibt. Der Spatial Extender subtrahiert diese Zahl von allen negativen X-Koordinaten, um einen positiven Wert zu erhalten. Der Spatial Extender subtrahiert diese Zahl auch von allen anderen X-Koordinaten.

Wenn der erweiterte X-Wert beispielsweise -105 lautet, müssen Sie -105 davon subtrahieren, um Null zu erhalten. Der Spatial Extender subtrahiert dann -105 von allen X-Koordinaten, die den dargestellten Merkmalen zugeordnet sind. Da keine dieser Koordinaten größer als -100 ist, sind alle durch diese Subtraktion erhaltenen Werte positiv.

- Geben Sie in der gleichen Weise Abstandswerte an, die nach der Subtraktion von dem erweiterten Y-Wert, dem erweiterten Z-Wert oder dem erweiterten Maß Null ergeben.

Der von den X-Koordinaten subtrahierte Abstand wird als *falsches X* bezeichnet. Die von den Y-Koordinaten, den Z-Koordinaten und dem Maß zu subtrahierenden Abstände werden entsprechend als *falsches Y*, *falsches Z* und *falsches M* bezeichnet. Anweisungen zum Angeben dieser Parameter über die Steuerzentrale finden Sie im Abschnitt „Über die Steuerzentrale ein räumliches Bezugssystem erstellen“ auf Seite 40.

Maßstabsfaktoren angeben

Geben Sie als nächstes an, welche Maßstabsfaktoren der Spatial Extender zur Umwandlung dezimaler Koordinaten und Maße in ganze Zahlen verwenden soll:

- Geben Sie einen Maßstabsfaktor an, der bei Multiplikation mit einer dezimalen X-Koordinate oder einer dezimalen Y-Koordinate eine ganze 32-Bit-Zahl ergibt. Es empfiehlt sich, für diesen Maßstabsfaktor eine Potenz von zehn zu verwenden: 10 hoch eins (10), 10 hoch zwei (100), 10 hoch drei (1000) oder ggf. eine höhere Zehnerpotenz. Legen Sie die Zehnerpotenz wie folgt fest:
 1. Ermitteln Sie, welche X- und Y-Koordinaten Dezimalzahlen sind bzw. sein werden. Angenommen, Sie stellen fest, dass drei der verschiedenen X- und Y-Koordinaten, mit denen Sie arbeiten, Dezimalzahlen sind: 1.23, 5.1235 und 6.789.
 2. Verwenden Sie die Dezimalkoordinate mit der längsten Dezimalgenauigkeit. Stellen Sie anschließend fest, mit welcher Zehnerpotenz diese Koordinate multipliziert werden kann, um eine ganze Zahl mit entsprechender Genauigkeit zu erhalten. In unserem Beispiel hat 5.1235

von den drei Dezimalkoordinaten die längste Dezimalgenauigkeit. Durch eine Multiplikation mit zehn hoch vier (10000) ergibt sich die Dezimalzahl 51235.

3. Stellen Sie fest, ob die durch die Multiplikation erhaltene ganze Zahl zu lang ist, um in sie in einem 32-Bit-Datenelement zu speichern. 51235 ist nicht zu lang. Aber nehmen Sie an, neben 1.23, 5.11235 und 6.789 umfasst der Bereich Ihrer X- und Y-Koordinaten noch einen vierten Dezimalwert 10006.789876. Die Dezimalgenauigkeit dieser Koordinate ist länger als die der anderen drei Dezimalkoordinaten; Sie multiplizieren daher *diese* Koordinate — nicht 5.1235 — mit dem Faktor 10. Zur Umwandlung in eine ganze Zahl könnten Sie sie mit zehn hoch sechs (1000000) multiplizieren. Der dadurch erhaltene Wert 10006789876 ist jedoch zu lang, um ihn in einem 32-Bit-Datenelement speichern zu können. Wenn der Spatial Extender versucht, diesen Wert zu speichern, sind die Ergebnisse nicht vorhersehbar.

Vermeiden Sie dieses Problem, indem Sie eine Potenz von zehn wählen, die bei der Multiplikation mit der ursprünglichen Koordinate eine Dezimalzahl ergibt, die der Spatial Extender so abschneiden kann, dass sich bei minimalem Genauigkeitsverlust eine ganze Zahl ergibt, die gespeichert werden kann. In diesem Fall können Sie zehn hoch vier (10000) wählen. Die Multiplikation von 10000 mit 10006.789876 ergibt 100067898.76. Der Spatial Extender schneidet diese Nummer auf 100067898 ab und reduziert damit ihre Genauigkeit um eine praktisch unbedeutende Größe.

- Wenn die dargestellten Merkmale dezimale Z-Koordinaten enthalten, ermitteln Sie anhand der vorangegangenen Prozedur einen geeigneten Maßstabsfaktor für diese Koordinaten. Wenn die Merkmale dezimalen Maßen zugeordnet sind, verwenden Sie diese Prozedur auch zum Ermitteln eines Maßstabsfaktors für diese Maße.

Der Maßstabsfaktor für X- und Y-Koordinaten wird als *XY-Einheit* bezeichnet. Die Skalierungsfaktoren für Z-Koordinaten und Maße werden als *Z-Einheiten* bzw. *M-Einheiten* bezeichnet. Anweisungen zum Angeben dieser Parameter über die Steuerzentrale finden Sie im Abschnitt „Über die Steuerzentrale ein räumliches Bezugssystem erstellen“.

Über die Steuerzentrale ein räumliches Bezugssystem erstellen

Dieser Abschnitt enthält eine Übersicht über die erforderlichen Schritte zum Erstellen eines räumlichen Bezugssystems über die Steuerzentrale. Im Anschluss an diese Übersicht finden Sie Details zur Durchführung der einzelnen Schritte.

Zur Ausführung dieser Schritte ist keine spezielle Berechtigung erforderlich.

Übersicht über die erforderlichen Schritte zum Erstellen eines räumlichen Bezugssystems über die Steuerzentrale:

1. Öffnen Sie das Fenster **Create Spatial Reference**.
2. Geben Sie an, welches Koordinatensystem verwendet werden soll.
3. Geben Sie die Kennungen für das zu erstellende räumliche Bezugssystem an.
4. Stellen Sie fest, welche Bereiche von Koordinaten und Maßen für die geografischen Merkmale gelten, zu denen Sie Informationen erhalten wollen.
5. Geben Sie Werte an für die Umwandlung negativer oder dezimaler Koordinaten und Maße in Datenelemente, die der Spatial Extender speichern kann.
6. Weisen Sie den Spatial Extender an, das gewünschte räumliche Bezugssystem zu erstellen.

Ausführliche Schritte zum Erstellen eines räumlichen Bezugssystems über die Steuerzentrale:

1. Öffnen Sie das Fenster **Create Spatial Reference**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Databases** unter dem Server finden, auf dem der Spatial Extender ausgeführt werden soll.
 - b. Klicken Sie den Ordner **Databases** an. Die Datenbanken werden im Inhalts-Teilfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die Datenbank an, die Sie für die räumlichen Daten aktiviert haben. Klicken Sie anschließend **Spatial Extender** → **Spatial References** im Kontextmenü an. Das Fenster "Spatial References" wird geöffnet.
 - d. Klicken Sie im Fenster **Spatial References** den Befehl **Create** an. Das Fenster **Create Spatial Reference** wird geöffnet.
2. Geben Sie im Fenster **Create Spatial Reference** im Feld **Coordinate system** an, welches Koordinatensystem verwendet werden soll.
3. Geben Sie die Kennungen für das zu erstellende räumliche Bezugssystem an.
 - Geben Sie im Feld **Name** einen Namen aus 1 - 64 Zeichen für das System ein.

Einschränkung: Geben Sie nicht den Namen eines anderen räumlichen Bezugssystems an. Zwei räumliche Bezugssysteme in der Datenbank können nicht den gleichen Namen haben.

- Geben Sie im ID-Feld eine numerische Kennung an. Diese Kennung muss eine ganze Zahl sein.

Einschränkung: Geben Sie nicht die ID eines anderen räumlichen Bezugssystems an. Zwei räumliche Bezugssysteme in der Datenbank können nicht die gleiche ID haben.

4. Ermitteln Sie mit einem Medium außerhalb der Steuerzentrale — beispielsweise auf Papier oder einer weißen Tafel — die niedrigsten negativen Koordinaten und Maße für die dargestellten geografischen Merkmale. Anweisungen hierzu finden Sie im Abschnitt „Die niedrigsten negativen Koordinaten und Maße ermitteln“ auf Seite 37.
5. Geben Sie im Fenster **Create Spatial Reference** Werte an für die Umwandlung negativer oder dezimaler Koordinaten und Maße in Datenelemente, die vom Spatial Extender unterstützt werden — also nicht negative ganze 32-Bit-Zahlen.
 - a. Geben Sie Werte an für die Umwandlung negativer oder dezimaler X-Koordinaten in nicht-negative ganze Zahlen:
 - Geben Sie in der Spalte **Offset** in dem Feld neben **X** ein "falsches X" ein:
 - Wenn irgendwelche Werte im Bereich der in Schritt 4 angegebenen X-Koordinaten negativ sind, geben Sie ein "falsches X" ein, das bei der Subtraktion von der niedrigsten negativen Koordinate eine positive Zahl ergibt. Anweisungen hierzu finden Sie im Abschnitt „Abstandsfaktoren angeben“ auf Seite 39.
 - Wenn keine der X-Koordinaten negativ ist, geben Sie ein falsches X von 0 ein.
 - Geben Sie in der Spalte **Scale factor** eine XY-Einheit in dem Feld ganz rechts von **X** ein. Diese XY-Einheit sollte bei Multiplikation mit einer beliebigen dezimalen X- oder Y-Koordinate eine ganze Zahl ergeben, die mit minimalem Genauigkeitsverlust als 32-Bit-Datenelement gespeichert werden kann. Anweisungen hierzu finden Sie im Abschnitt „Maßstabsfaktoren angeben“ auf Seite 39.

Nach der Angabe der XY-Einheit in dem Feld ganz rechts von **X** erscheint diese Einheit auch in dem Feld ganz rechts von **Y**.
 - b. Geben Sie ein "falsches Y" ein, das dem Spatial Extender die Umwandlung negativer Y-Koordinaten in positive Werte ermöglicht. Geben Sie diesen Wert in der Spalte **Offset** in dem Feld neben **Y** ein:
 - Wenn irgendwelche Werte im Bereich der in Schritt 4 angegebenen Y-Koordinaten negativ sind, geben Sie ein "falsches Y" ein, das bei der Subtraktion von der niedrigsten negativen Koordinate eine positive Zahl ergibt. Anweisungen hierzu finden Sie im Abschnitt „Abstandsfaktoren angeben“ auf Seite 39.
 - Wenn alle Y-Koordinaten positiv sind, geben Sie ein "falsches Y" von 0 ein.

- c. Wenn Sie Z-Koordinaten in Ihre räumlichen Daten einbeziehen wollen, geben Sie Werte an zur Umwandlung negativer oder dezimaler Z-Koordinaten in nicht-negative ganze Zahlen.
- Geben Sie in der Spalte **Offset** in dem Feld neben **Z** ein "falsches Z" ein:
 - Wenn irgendwelche Werte im Bereich der in Schritt 4 auf Seite 42 angegebenen Z-Koordinaten negativ sind, geben Sie ein "falsches Z" ein, das bei der Subtraktion von der niedrigsten negativen Koordinate eine positive Zahl ergibt. Anweisungen hierzu finden Sie im Abschnitt „Abstandsfaktoren angeben“ auf Seite 39.
 - Wenn keine der Z-Koordinaten negativ ist, geben Sie ein falsches Z von 0 ein.
 - Geben Sie in der Spalte **Scale factor** eine Z-Einheit in dem Feld ganz rechts von **Z** ein. Diese Z-Einheit sollte bei Multiplikation mit einer beliebigen dezimalen Z-Koordinate eine ganze Zahl ergeben, die mit minimalem Genauigkeitsverlust als 32-Bit-Datenelement gespeichert werden kann. Anweisungen hierzu finden Sie im Abschnitt „Maßstabsfaktoren angeben“ auf Seite 39.
- d. Wenn Sie Maße in Ihre räumlichen Daten einbeziehen wollen, geben Sie Werte an zur Umwandlung negativer oder dezimaler Maße in nicht-negative ganze Zahlen.
- Geben Sie in der Spalte **Offset** in dem Feld neben **Linear** ein "falsches M" ein:
 - Wenn irgendwelche Werte im Bereich der in Schritt 4 auf Seite 42 angegebenen Maße negativ sind, geben Sie ein "falsches M" ein, das bei der Subtraktion von dem niedrigsten negativen Maß eine positive Zahl ergibt. Anweisungen hierzu finden Sie im Abschnitt „Abstandsfaktoren angeben“ auf Seite 39.
 - Wenn alle Maße positiv sind, geben Sie ein "falsches M" von 0 ein.
 - Geben Sie in der Spalte **Scale factor** eine M-Einheit in dem Feld ganz rechts von der Bezeichnung **Linear** ein. Diese M-Einheit sollte bei Multiplikation mit einem beliebigen Maß eine ganze Zahl ergeben, die mit minimalem Genauigkeitsverlust als 32-Bit-Datenelement gespeichert werden kann. Anweisungen hierzu finden Sie im Abschnitt „Maßstabsfaktoren angeben“ auf Seite 39.
6. Klicken Sie **OK** an, um das gewünschte räumliche Bezugssystem zu erstellen.

Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren

Nachdem Sie Ressourcen für Ihr Spatial Extender-GIS eingerichtet haben, können Sie Objekte erstellen, die räumliche Daten enthalten. Wenn Sie beispielsweise neue Tabellen benötigen, die räumliche Daten enthalten sollen, können Sie diese definieren, indem Sie den Spalten Typen von räumlichen Daten zuordnen. Oder Sie können vorhandenen Tabellen räumliche Spalten hinzufügen.

Wenn Sie in einer neuen oder einer vorhandenen Tabelle eine räumliche Spalte definieren, müssen Sie diese Spalte als Schicht registrieren. Darüber hinaus können Sie, wenn die Spalte über einen Geocodierer ausgefüllt werden soll, bei der Registrierung der Spalte als Schicht den Geocodierer aktivieren, damit er die Spalte automatisch verwaltet. Diese Aktivierung erfolgt auf die folgende Weise: Der Spatial Extender definiert Auslöser, die den Geocodierer immer aufrufen, wenn die entsprechende attributive Spalte (bzw. Spalten) der räumlichen Spalte neue oder aktualisierte Daten empfängt. Beim Aufruf wandelt der Geocodierer die neuen oder aktualisierten Daten in räumliche Daten um und legt diese Daten in der räumlichen Spalte ab.

Nach der Definition einer räumlichen Spalte für die Tabelle können Sie bei Bedarf eine Sichtspalte über dieser Tabellenspalte erstellen. Sie müssen die Sichtspalte als Schicht registrieren, nachdem Sie die Tabellenspalte als Schicht registriert haben.

In diesem Kapitel wird das Wesen und die Verwendung der Datentypen, die Sie der räumlichen Spalte zuordnen können, beschrieben. Als nächstes beschreibt das Kapitel, wie mit der Steuerzentrale eine räumliche Spalte für eine Tabelle definiert wird, um diese Spalte als Spalte zu registrieren, und wie ein Geocodierer für die Verwaltung der Schicht aktiviert wird. Das Kapitel schließt mit einer Erläuterung, wie über die Steuerzentrale eine Sichtspalte als Schicht registriert wird.

Informationen zu Typen von räumlichen Daten

Dieser Abschnitt stellt die für räumliche Spalten erforderlichen Datentypen vor und bietet Richtlinien zur Auswahl eines geeigneten Datentyps für eine räumliche Spalte.

Wenn Sie eine Datenbank für räumliche Operationen aktivieren, stellt der Spatial Extender der Datenbank eine Hierarchie strukturierter Datentypen zur

Verfügung. Abb. 6 zeigt diese Hierarchie. In dieser Abbildung haben die Exemplartypen einen weißen Hintergrund; die nicht exemplarfähigen Typen sind auf einem schraffierten Hintergrund dargestellt.

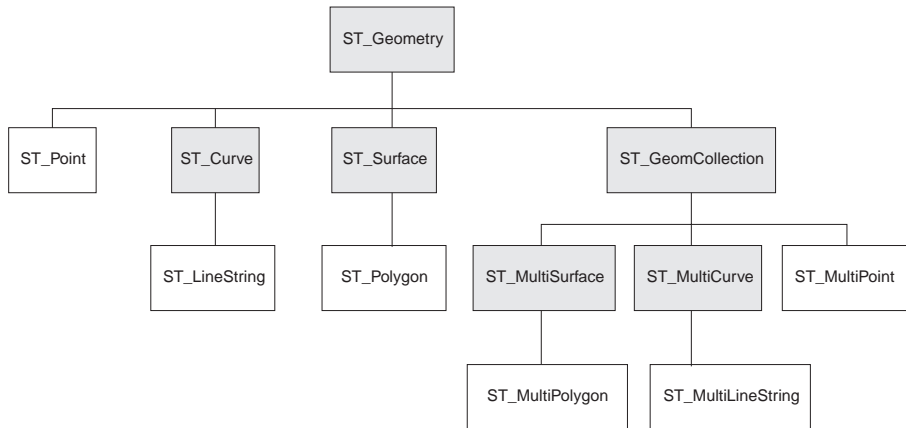


Abbildung 6. Hierarchie der Typen räumlicher Daten. Die Datentypen in weißen Kästchen sind exemplarfähig. Die Datentypen in schraffierten Kästchen sind nicht exemplarfähig.

Die Hierarchie in Abb. 6 umfasst:

- Datentypen für geografische Merkmale, die eine Einheit bilden, beispielsweise einzelne Wohngebiete und isolierte Seen.
- Datentypen für geografische Merkmale, die aus mehreren Einheiten oder Komponenten bestehen, beispielsweise Fernstraßensysteme und Gebirgszüge.
- Ein Datentyp für geografische Merkmale aller Art.

Datentypen für Einheitenmerkmale

Verwenden Sie `ST_Point`, `ST_LineString` und `ST_Polygon` zum Speichern von Koordinaten zur Definition des Raumes, der von Merkmalen belegt wird, die eine einzelne Einheit bilden:

- Verwenden Sie `ST_Point`, wenn Sie einen Punkt im Raum kennzeichnen wollen, der von einem eindeutigen geografischen Merkmal belegt wird. Das Merkmal kann ein sehr kleines sein, beispielsweise eine Wasserquelle, ein großes wie beispielsweise eine Stadt, oder ein mittleres wie ein Gebäudekomplex oder ein Park. In jedem Fall kann der Punkt im Raum am Kreuzungspunkt der Ost-West-Koordinatenlinie (z. B. einem Breitengrad) mit einer Nord-Süd-Koordinatenlinie (z. B. einem Längengrad) gefunden werden. Ein `ST_Point`-Datenelement enthält Werte — eine X-Koordinate und eine Y-Koordinate — zur Definition eines solchen Kreuzungspunkts. Die X-Koordinate gibt an, wo der Punkt auf der Ost-West-Linie liegt; die Y-Koordinate gibt an, wo er auf der Nord-Süd-Linie liegt.

- Verwenden Sie `ST_LineString` für Koordinaten, die den Raum definieren, der von linearen Funktionen belegt wird, beispielsweise Straßen, Kanäle oder Rohrleitungen.
- Verwenden Sie `ST_Polygon`, wenn Sie einen Bereich kennzeichnen wollen, der von einem mehrseitigen Merkmal abgedeckt wird, beispielsweise ein Industriegebiet, einen Wald oder ein Naturschutzgebiet. Ein `ST_Polygon`-Datenelement besteht aus den Koordinaten, die den Umfang eines solchen Merkmals definieren.

In manchen Fällen können `ST_Polygon` und `ST_Point` für dieselbe Funktion verwendet werden. Angenommen, Sie benötigen räumliche Informationen über verschiedene Apartmentkomplexe. Wenn Sie einen Punkt im Raum darstellen wollen, an dem sich die einzelnen Komplexe befinden, verwenden Sie `ST_Point` zum Speichern der X- und Y-Koordinaten, die einen solchen Punkt definieren. Wenn Sie andererseits den Bereich darstellen wollen, den die einzelnen Komplexe abdecken, verwenden Sie `ST_Polygon` zum Speichern der Koordinaten, die den Umfang des Bereichs angeben.

Datentypen für Merkmale mit mehreren Einheiten

Verwenden Sie `ST_MultiPoint`, `ST_MultiLineString` und `ST_MultiPolygon` zum Speichern von Koordinaten für den Raum, der von Merkmalen belegt wird, die aus mehreren Einheiten bestehen:

- Verwenden Sie `ST_MultiPoint`, wenn Sie Merkmale darstellen wollen, die aus eindeutigen Einheiten bestehen, und wenn Sie außerdem den Punkt im Raum angeben wollen, der von den einzelnen Komponenten belegt wird. Ein `ST_MultiPoint`-Datenelement enthält die Paare der X- und Y-Koordinaten, die die Position jeder Komponente eines solchen Merkmals definieren. Betrachten Sie beispielsweise eine Tabelle, deren Zeilen Inselketten darstellen und deren Spalten eine `ST_MultiPoint`-Spalte enthalten. Jedes Datenelement in der Spalte enthält die Paare der X- und Y-Koordinaten, die die Position der Inseln in den einzelnen Ketten definieren.
- Verwenden Sie `ST_MultiLineString`, wenn Sie Merkmale darstellen wollen, die aus linearen Einheiten bestehen, und wenn Sie außerdem Informationen zu dem von jeder einzelnen Einheit belegten Raum abfragen wollen. Ein `ST_MultiLineString`-Datenelement besteht aus den Koordinaten, die solche Räume definieren. Betrachten Sie beispielsweise eine Tabelle, deren Zeilen Fluss-Systeme darstellen und deren Spalten eine `ST_MultiLineString`-Spalte enthalten. Jedes Datenelement in dieser Spalte enthält die Koordinatengruppen, die den Verlauf der Flüsse in den einzelnen Systemen definieren.
- Verwenden Sie `ST_MultiPolygon`, wenn Sie Merkmale darstellen wollen, die aus mehrseitigen Einheiten bestehen, und wenn Sie außerdem Informationen zu dem von jeder einzelnen Einheit belegten Raum abfragen wollen. Betrachten Sie beispielsweise eine Tabelle, deren Zeilen Landkreise im Norden Deutschlands darstellen und deren Spalten eine `ST_MultiPolygon`-Spalte enthalten. Diese Spalte enthält Informationen zu Bauernhöfen.

Jedes Datenelement in der Spalte enthält die Koordinatengruppen, die den Umfang der Bauernhöfe in einem bestimmten Landkreis definieren.

Ein Datentyp für alle Merkmale

Sie können ST_Geometry verwenden, wenn Sie nicht sicher sind, welcher der anderen Datentypen geeignet ist. Da ST_Geometry der Ausgangspunkt der Hierarchie ist, zu dem die anderen Datentypen gehören, kann eine ST_Geometry-Spalte einige oder alle Werte speichern, die in Spalten gespeichert werden können, denen andere Datentypen zugeordnet sind.

Achtung: Wenn Sie beabsichtigen, den Standardgeocodierer zum Ausfüllen einer räumlichen Spalte zu verwenden, muss die Spalte den Typ ST_Point oder ST_Geometry aufweisen.

Eine räumliche Spalte für eine Tabelle definieren, diese Spalte als Schicht registrieren und einen Geocodierer zum Verwalten aktivieren

Dieser Abschnitt gibt einen Überblick über die erforderlichen Schritte zum Definieren einer räumlichen Spalte für eine Tabelle, zum Registrieren dieser Spalte als Schicht und zum Aktivieren eines Geocodierers für die Verwaltung. Im Anschluss an diese Übersicht finden Sie Details zu Durchführung der einzelnen Schritte.

Informationen darüber, welche Berechtigung Sie zum Registrieren einer Tabellenspalte als Schicht benötigen, finden Sie im Abschnitt „Autorisierung“ auf Seite 110. Informationen darüber, welche Berechtigung Sie zum Aktivieren eines Geocodierers zum Verwalten einer Tabellenspalte als Schicht benötigen, finden Sie im Abschnitt „Autorisierung“ auf Seite 91.

Überblick über die Schritte zum Definieren einer räumlichen Spalte als Tabelle, zum Registrieren dieser Spalte als Schicht und zum Aktivieren eines Geocodierers für die Verwaltung:

1. Wenn die räumliche Spalte Teil einer neuen Tabelle sein soll, erstellen Sie diese Tabelle.
2. Öffnen Sie das Fenster **Create Spatial Layer**.
3. Fügen Sie entweder eine räumliche Spalte einer Tabelle hinzu und geben Sie an, dass diese Spalte als Schicht registriert werden soll, oder geben Sie an, dass eine vorhandene Spalte als Schicht registriert werden soll.
4. Geben Sie an, welches räumliche Bezugssystem für die Schicht verwendet werden soll.
5. Wenn die Schicht importierte Daten enthält oder die Daten aus einer anderen räumlichen Spalte generiert werden, weisen Sie den Spatial Extender an, die Schicht zu erstellen.

6. Wenn die Schicht Daten enthalten soll, die aus attributiven Daten abgeleitet wurden:
 - a. Geben Sie an, welche Spalte bzw. Spalten diese attributiven Daten enthält.
 - b. Geben Sie an, dass Sie einen Geocodierer zur Verwaltung der Schicht aktivieren wollen.
 - c. Weisen Sie den Spatial Extender an, die Schicht zu erstellen.

Ausführliche Informationen zu den Schritten zum Definieren einer räumlichen Spalte als Tabelle, zum Registrieren dieser Spalte als Schicht und zum Aktivieren eines Geocodierers für die Verwaltung:

1. Wenn die räumliche Spalte Teil einer neuen Tabelle sein soll, erstellen Sie diese Tabelle:
 - Verwenden Sie eine Schnittstelle Ihrer Wahl (z. B. die Steuerzentrale oder den Befehlszeilenprozessor) zum Erstellen der Tabelle.
 - Wenn Sie einen Geocodierer verwenden wollen, beziehen Sie zwischen einer und zehn Spalten ein, mit denen der Geocodierer arbeiten kann. Ein Geocodierer kann nicht mehr als zehn Spalten als Eingabe verarbeiten.
 - Beziehen Sie entweder die räumliche Spalte, die Sie als Schicht registrieren wollen, ein, oder definieren Sie diese Spalte in Schritt 3.

Wenn Sie eine vorhandene Tabelle verwenden wollen fahren Sie mit dem nächsten Schritt fort.

2. Öffnen Sie das Fenster **Create Spatial Layer**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Tables** für die Tabellen in der Datenbank finden, die Sie für die räumlichen Operationen verwenden wollen.
 - b. Klicken Sie den Ordner **Tables** an. Die Tabellen werden im Inhaltsteilfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die gewünschte Tabelle an, und klicken Sie **Spatial Extender** → **Spatial Layers** im Kontextmenü an. Das Fenster **Spatial Layers** wird geöffnet.
 - d. Klicken Sie im Fenster **Spatial Layers** die Option **Create** an. Das Fenster **Create Spatial Layer** wird geöffnet.
3. Fügen Sie im Fenster **Create Spatial Layer** entweder eine räumliche Spalte einer Tabelle hinzu und geben Sie an, dass diese Spalte als Schicht registriert werden soll, oder geben Sie an, dass eine vorhandene Spalte als Schicht registriert werden soll.
 - So fügen Sie der Tabelle eine räumliche Spalte hinzu und definieren diese Spalte als Schicht:
 - a. Geben Sie im Feld **Layer column** einen Namen für die Spalte ein.

- b. Wählen Sie im Feld **Column type** den gewünschten Datentyp für die Spalte aus bzw. geben Sie ihn ein. Eine Beschreibung der zulässigen Datentypen finden Sie im Abschnitt „Informationen zu Typen von räumlichen Daten“ auf Seite 45.
- Wenn Sie vorhandene Spalte als Schicht definieren wollen, wählen Sie sie im Feld **Layer column** aus.

Einschränkung: Wählen Sie keine Spalte aus, die bereits als Schicht definiert wurde.

4. Geben Sie im Feld **Spatial reference name** den Namen des räumlichen Bezugssystems an, der für die Schicht verwendet werden soll.
5. Wenn die Schicht importierte Daten enthalten soll oder Daten, die aus einer anderen räumlichen Spalte generiert wurden, klicken Sie **OK** an, um sie zu registrieren.
6. Wenn die Schicht Daten enthalten soll, die aus attributiven Daten abgeleitet wurden, gehen Sie wie folgt vor:
 - a. Geben Sie an, welche Spalte bzw. Spalten diese attributiven Daten enthält:
 - 1) Wählen Sie die Spalte(n) im Feld **Available columns** aus. Sie können bis zu zehn Spalten auswählen.
 - 2) Klicken Sie den Druckknopf >, den Druckknopf >> oder beide an, um die ausgewählte Spalten im Feld **Selected columns** aufzulisten.
 - b. So aktivieren Sie einen Geocodierer zur Verwaltung der Schicht:
 - 1) Wählen Sie das Markierungsfeld **Enable automatic geocoder** aus.
 - 2) Wählen Sie im Feld **Name** den Namen des Geocodierers aus, den Sie verwenden wollen.
 - 3) Geben Sie im Feld **Precision level** den Grad in Prozent an, in dem die Eingabedatensätze mit den entsprechenden Datensätzen in den Bezugsdaten übereinstimmen müssen, damit sie verarbeitet werden können. Dieser Prozentsatz wird als *Genauigkeit* bezeichnet. Angenommen, der Geocodierer liest einen Eingabedatensatz mit der Adresse 557 Bailey, San Jose 94120, ein. Wenn die Genauigkeit 100 ist und die Übereinstimmung zwischen dieser Adresse und ihrem Pendant in den Bezugsdaten nicht 100 Prozent beträgt, weist der Geocodierer die Adresse zurück. Ist die Genauigkeit 75 und die Übereinstimmung zwischen dem Datensatz und dem Pendant in den Bezugsdaten beträgt mindestens 75 Prozent, verarbeitet der Geocodierer die Adresse.
 - 4) Wenn der Geocodierer von einem separaten Anbieter bereitgestellt wurde, geben Sie im Feld **Properties** alle herstellereigene Geocodierungsparameter an, die Sie verwenden wollen.

- c. Klicken Sie **OK** an, um die ausgewählte Spalte als Schicht zu registrieren und, falls Sie diese Funktion angefordert haben, zum Aktivieren des Geocodierers für die Verwaltung der Spalte.

Einschränkungen

Sie können die Spalten, die für den Geocodierer eingegeben werden, nicht ändern. Sie haben jedoch die Möglichkeit, andere Merkmale der Geocodierung zu ändern. Beispielsweise können Sie die schrittweise steigende Geocodierung aktivieren oder inaktivieren.

Anmerkung: Das Fenster **Delete Layer** wird nicht verwendet, um Schichten zu löschen, sondern um ihre Registrierung aufzuheben, d. h. um alle Informationen über die Schichten aus dem Systemkatalog des DB2 Spatial Extender zu entfernen. Wenn Sie die Registrierung einer Schicht aufheben, bleibt die für die Schicht verwendete Tabellen- bzw. Sichtspalte bestehen.

Eine Sichtspalte als Schicht registrieren

Informationen darüber, welche Berechtigung Sie zum Registrieren einer Sichtspalte als Schicht benötigen, finden Sie im Abschnitt „Autorisierung“ auf Seite 110.

So registrieren Sie eine Sichtspalte als Schicht:

1. Öffnen Sie das Fenster **Create Spatial Layer**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Views** für die Sichten in der Datenbank finden, die Sie für die räumlichen Operationen verwenden wollen.
 - b. Klicken Sie den Ordner **Views** an. Die Sichten werden im Inhaltsfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die gewünschte Sicht an, und klicken Sie **Spatial Extender** → **Spatial Layers** im Kontextmenü an. Das Fenster **Spatial Layers** wird geöffnet.
 - d. Klicken Sie im Fenster **Spatial Layers** die Option **Create** an. Das Fenster **Create Spatial Layer** wird geöffnet.
2. Geben Sie im Feld **Layer column** die Spalte an, die Sie als Schicht registrieren wollen.
3. Geben Sie im Feld **Underlying spatial layer** den Namen der Tabellenspalte ein, auf der die ausgewählte Sichtspalte basiert. Diese Tabellenspalte muss bereits als Schicht registriert worden sein.
4. Klicken Sie **OK** an, um die angegebene Sichtspalte als Schicht zu registrieren.

Anmerkung: Wenn Sie eine Tabelle löschen wollen, die eine als Schicht registrierte Spalte enthält, müssen Sie zuvor eine oder mehrere der folgenden Aktionen ausführen:

- Heben Sie die Registrierung der Schicht auf.
- Wenn für die Schicht ein räumlicher Index existiert, löschen Sie diesen Index.
- Wenn über der Tabellenspalte eine Sichtspalte definiert wurde und diese Sichtspalte als Schicht registriert wurde, heben Sie auch die Registrierung dieser Schicht auf.

Kapitel 5. Räumliche Spalten ausfüllen

Nachdem Sie räumliche Spalten als Schichten registriert haben, können Sie sie mit räumlichen Daten ausfüllen. Wie im Abschnitt „Woher räumliche Daten stammen“ auf Seite 6 angegeben, können diese Daten auf drei Arten eingetragen werden: über eine als Geocodierer bezeichnete Funktion, die die Daten aus attributiven Daten ableitet; mit Hilfe anderer Funktionen, die die Daten aus anderen räumlichen Daten ableiten; oder durch Importieren aus Dateien. Dieses Kapitel enthält Beschreibungen zu folgenden Punkten:

- Geocodierung und Verwendung der Steuerzentrale zum Geocodieren attributiver Daten im Stapelbetrieb
- Importieren und Exportieren von Daten und Verwendung der Steuerzentrale zum Importieren von Daten in Ihr GIS und zum Exportieren aus Ihrem GIS

Informationen zu anderen Funktionen, die neue räumliche Daten aus vorhandenen räumlichen Daten ableiten können, finden Sie im Abschnitt „Funktionen, die neue Geometrien aus vorhandenen generieren“ auf Seite 186.

Geocodierer verwenden

In diesem Abschnitt wird der Prozess der Geocodierung beschrieben und wie ein Geocodierer im Stapelbetrieb von der Steuerzentrale aus ausgeführt wird.

Informationen zur Geocodierung

Wie bereits erwähnt, verwendet DB2 Spatial Extender eine als *Geocodierer* bezeichnete Funktion für zwei Aktivitäten: die Umsetzung von Adresdaten in räumliche Daten und die Eintragung dieser räumlichen Daten in Tabellenspalten. Dieser Abschnitt erläutert die wesentlichen Unterschiede zwischen Geocodierern und ihren Quellen. Außerdem beschreibt er zwei Modi, in denen ein Geocodierer arbeiten kann, und stellt Faktoren vor, die bei der Planung der Arbeit mit einem Geocodierer berücksichtigt werden sollten.

Mit dem Spatial Extender haben Sie folgende Möglichkeiten:

- Den mit dem Spatial Extender bereitgestellten Standard-Geocodierer verwenden
- Von anderen Herstellern entwickelte Geocodierer installieren
- Eigene Geocodierer installieren

Der Standard-Geocodierer kann Adressen in den USA geocodieren und sie in ST_Point-Daten umsetzen. Wenn Sie Daten mit anderen räumlichen Datentypen speichern wollen, können Sie einen anderen Geocodierer installieren,

der solche Daten generiert. Wenn Sie räumliche Daten benötigen, die Standorte außerhalb der USA darstellen oder Standorte ohne Adresse, beispielsweise Ackerland mit unterschiedlichem Anteil an Muttererde, können Sie auch für diese Anforderung einen weiteren Geocodierer installieren.

Bevor Sie mit einem zusätzlichen Geocodierer arbeiten können, muss dieser registriert worden sein. Benutzer und Hersteller können einen solchen Geocodierer mit der gespeicherten Prozedur `db2gse.gse_register_gc` registrieren. Der Geocodierer kann nicht über die Steuerzentrale registriert werden.

Informationen zu `db2gse.gse_register_gc` finden Sie im Abschnitt „`db2gse.gse_register_gc`“ auf Seite 108. Allgemeine Informationen zur Verwendung der gespeicherten Prozeduren des Spatial Extender finden Sie in „Kapitel 9. Gespeicherte Prozeduren“ auf Seite 83.

Ein Geocodierer kann in zwei verschiedenen Modi arbeiten:

- Im *Stapelbetrieb* versucht der Geocodierer, in einem einzigen Arbeitsgang alle vorhandenen Quelldaten für räumliche Daten in räumliche Daten umzusetzen und die Spalten mit diesen Daten auszufüllen. Sie können diese Operation vom Fenster **Run Geocoder** aus starten. Alternativ dazu können Sie sie auch in einem Anwendungsprogramm starten, indem Sie das Programm für den Aufruf der gespeicherten Prozedur `db2gse.gse_run_gc` codieren.
- Im *schrittweise steigenden Modus* wandelt ein Geocodierer Daten um, wenn sie in eine Tabelle eingefügt oder aktualisiert werden. Dabei legt er die resultierenden räumlichen Daten in räumlichen Spalten ab, um die Spalte auf dem neuesten Stand zu halten. Der Geocodierer wird durch Einfüge- und Aktualisierungsauslöser aktiviert; diese Auslöser können Sie über das Fenster **Create Spatial Layer** anfordern. Alternativ dazu können Sie sie in einem Anwendungsprogramm anfordern, indem Sie das Programm für den Aufruf der gespeicherten Prozedur `db2gse.gse_enable_autogc` codieren.
Die schrittweise steigende Geocodierung wird auch als *automatische Geocodierung* bezeichnet.

Wenn Sie vorhaben, einen Geocodierer zu verwenden, sollten Sie eventuell die folgenden Faktoren berücksichtigen:

1. Wenn Sie mit der Steuerzentrale arbeiten, verwenden Sie normalerweise das Fenster **Create Spatial Layers**, bevor Sie das Fenster **Run Geocoder** aufrufen. Dies bedeutet, dass Sie den Spatial Extender anweisen können, mehrere Auslöser für die schrittweise steigende Geocodierung einzurichten, bevor Sie die Geocodierung im Stapelbetrieb starten. Es ist deshalb möglich, dass eine schrittweise Geocodierung einer Geocodierung im Stapelbetrieb vorausgeht. Bei der Verarbeitung aller Quelldaten im Stapelbetrieb werden die gleichen Daten geocodiert wie im schrittweise steigenden Modus. Diese Redundanz führt nicht zu Duplizierungen; wenn

räumliche Daten zweimal generiert werden, überschreibt das zweite Datenergebnis das erste. Die Leistung kann dadurch jedoch sinken. Eine Möglichkeit, diesen Leistungsabfall zu vermeiden, ist das verzögerte Einrichten von Auslösern bis nach der Geocodierung im Stapelbetrieb.

2. Wenn die Auslöser vorhanden sind, wenn Sie mit der Geocodierung im Stapelbetrieb beginnen wollen, empfiehlt es sich, diese Auslöser zu inaktivieren, bis die Geocodierung im Stapelbetrieb abgeschlossen ist. Sie können die Auslöser über das Fenster **Run Geocoder** oder in einem Anwendungsprogramm inaktivieren, indem Sie das Programm für den Aufruf der gespeicherten Prozedur `db2gse.gse_disable_autogc` codieren. Wenn Sie das Fenster **Run Geocoder** verwenden, reaktiviert der Spatial Extender die Auslöser automatisch, wenn die Geocodierung abgeschlossen ist. Falls Sie die gespeicherte Prozedur `db2gse.gse_disable_autogc` verwenden, können Sie die Auslöser über den Aufruf der gespeicherten Prozedur `db2gse.gse_enable_autogc` reaktivieren.
3. Wenn Sie einen Geocodierer im Stapelbetrieb ausführen wollen, um eine räumliche Spalte mit einem Index auszufüllen, inaktivieren Sie zunächst den Index oder geben Sie ihn frei. Andernfalls bleibt der Index während der Ausführung des Geocodierers betriebsbereit, was die Leistung beträchtlich verschlechtert. Wenn Sie mit der Steuerzentrale arbeiten, können Sie den Index über das Fenster **Run Geocoder** inaktivieren. Der Spatial Extender reaktiviert den Index automatisch, wenn die Geocodierung abgeschlossen ist. Wenn Sie ein Anwendungsprogramm verwenden, können Sie den Index mit der Anweisung `SQL DROP` freigeben. Notieren Sie sich in diesem Fall unbedingt die Parameter des Index, so dass Sie ihn nach Abschluss der Geocodierung wieder erstellen können.
4. Wenn der Geocodierer einen Datensatz einliest, versucht er, diesen Datensatz mit seinem Pendant in den Bezugsdaten abzugleichen. Dieser Abgleich muss mindestens eine bestimmte *Genauigkeit* aufweisen, damit der Geocodierer den Datensatz verarbeitet. Eine Genauigkeit von 85 bedeutet beispielsweise, dass ein Quelldatensatz und sein Pendant in den Bezugsdaten zu mindestens 85 Prozent übereinstimmen müssen, damit der Quelldatensatz verarbeitet wird.

Sie geben selbst an, wie hoch die Genauigkeit sein muss. Denken Sie daran, dass Sie die Genauigkeit eventuell anpassen müssen. Angenommen, die Genauigkeit ist 100. Wenn viele Quelldatensätze Adressen enthalten, die neuer sind als die Bezugsdaten, sind Übereinstimmungen von 100 Prozent zwischen diesen Datensätzen und den Bezugsdaten nicht möglich. Der Geocodierer weist diese Datensätze daher zurück. Insgesamt gilt: Wenn ein Geocodierer räumliche Daten generiert, die ungenügend oder weitgehend ungenau sind, können Sie dieses Problem eventuell umgehen, indem Sie die Genauigkeit ändern und den Geocodierer erneut starten.

Es gibt mehrere Möglichkeiten, die Anzahl oder den Bereich der Datensätze zu steuern, die ein Geocodierer vor dem Festschreiben bearbeitet:

Methode 1

Ein Geocodierer kann vor jeder Festschreibung Adressdaten in einer bestimmten Anzahl von Datensätzen geocodieren. Mit dieser Methode kann die Größe der Arbeitseinheiten genau gesteuert werden. Allerdings ist der Systemaufwand bei dieser Methode erheblich höher als bei den übrigen hier vorgestellten Methoden.

Geben Sie zur Verwendung von Methode 1 die Anzahl der Datensätze an, die vor jeder Festschreibung verarbeitet werden sollen. Wenn Sie mit der Steuerzentrale arbeiten, stellen Sie diese Zahl mit dem Drehknopf **Commit scope** im Fenster **Run Geocoder** ein. Wenn Sie ein Anwendungsprogramm schreiben, ordnen Sie diese Zahl dem Parameter `commitScope` der gespeicherten Prozedur `db2gse.gse_run_gc` zu.

Methode 2

Ein Geocodierer kann vor jeder Festschreibung Adressdaten in allen Datensätzen einer Tabelle geocodieren. Bei dieser Methode verarbeitet der Geocodierer die Datensätze ähnlich wie eine Stapeloperation, wobei der Systemaufwand pro Datensatz erheblich niedriger ist als bei Methode 1. Allerdings können Sie bei dieser Methode die Größe der Arbeitseinheit nicht im selben Umfang steuern wie bei Methode 1. Infolgedessen haben Sie keinen Einfluss darauf, wie viele Sperren beibehalten werden oder wie viele Protokolleinträge während der Ausführung des Geocodierers erstellt werden. Darüber hinaus müssen Sie den Geocodierer so einstellen, dass er im Fall eines Fehlers, der eine ROLLBACK-Operation erfordert, alle Datensätze erneut verarbeitet. Dies kann erhebliche Ressourcenkosten verursachen, wenn die Tabelle sehr groß ist und der Fehler sowie die ROLLBACK-Operation auftreten, nachdem die meisten Datensätze bereits verarbeitet wurden.

Zum Starten von Methode 2 von der Steuerzentrale aus stellen Sie den Drehknopf **Commit scope** im Fenster **Run Geocoder** auf Null. Wenn Sie ein Anwendungsprogramm schreiben, setzen Sie den Parameter `commitScope` der gespeicherten Prozedur `db2gse.gse_run_gc` auf Null.

Methode 3

Ein Geocodierer kann vor jeder Festschreibung Adressdaten in einer Untergruppe der Datensätze einer Tabelle geocodieren. Anschließend kann der Geocodierer die Adressdaten in einer zweiten und, falls erforderlich, in einer dritten, einer vierten Untergruppe usw. geocodieren.

Der Geocodierer verarbeitet jede Untergruppe ähnlich wie eine Stapeloperation, wobei der Systemaufwand pro Datensatz erheblich niedriger ist als bei Methode 1. Ebenso wie Methode 2 erlaubt jedoch auch Methode 3 keine direkte Einflussnahme auf die Größe der Arbeitseinheit. Außerdem fällt hier Systemaufwand für die mehrfache Einstellung und Ausführung des Geocodierers für jede Untergruppe von Datensätzen an.

So definieren Sie über die Steuerzentrale eine Untergruppe von Datensätzen:

- Codieren Sie im Fenster **Run Geocoder** im Feld **Where clause** eine SELECT WHERE-Klausel, die den Bereich der Datensätze zum jeweiligen Thema festlegt.
- Stellen Sie den Drehknopf **Commit scope** im Fenster **Run Geocoder** auf Null.

Zur Definition einer Untergruppe von Datensätzen in einem Anwendungsprogramm codieren Sie die gespeicherte Prozedur db2gse.gse_run_gc wie folgt:

- Geben Sie über den Parameter whereClause den Bereich der Datensätze in der Untergruppe an.
- Setzen Sie den Parameter commitScope auf Null.

Den Geocodierer im Stapelbetrieb ausführen

Dieser Abschnitt enthält eine Übersicht über die Schritte zur Ausführung eines Geocodierers im Stapelbetrieb von der Steuerzentrale aus. Im Anschluss an diese Übersicht finden Sie Details zur Durchführung der einzelnen Schritte.

Informationen darüber, welche Berechtigung Sie zum Ausführen eines Geocodierers im Stapelbetrieb benötigen, finden Sie im Abschnitt „Autorisierung“ auf Seite 117.

Übersicht über die Schritte zur Ausführung eines Geocodierers im Stapelbetrieb:

1. Öffnen Sie das Fenster **Run Geocoder**.
2. Geben Sie an, welcher Geocodierer verwendet werden soll.
3. Inaktivieren Sie Objekte, die die Leistung des Geocodierers beeinträchtigen könnten.
4. Geben Sie an, wie viele Datensätze geocodiert werden sollen, bevor DB2 die Datensätze festschreibt.
5. Geben Sie an, wie der Geocodierer arbeiten soll.
6. Weisen Sie den Spatial Extender an, den Geocodierer zu starten.

Ausführliche Schritte zur Ausführung eines Geocodierers im Stapelbetrieb:

1. Öffnen Sie das Fenster **Run Geocoder**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Tables** in der für die räumlichen Operationen aktivierten Datenbank finden.
 - b. Klicken Sie den Ordner **Tables** an. Die Tabellen werden im Inhaltsteilfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die Tabelle in dem Inhaltsteilfenster an, und klicken Sie **Spatial layers** im Kontextmenü an. Das Fenster **Spatial Layers** wird geöffnet.
 - d. Führen Sie im Fenster **Spatial Layers** folgende Aktionen aus:
 - 1) Wählen Sie die Schicht aus, die in der auszufüllenden Spalte definiert ist.
 - 2) Klicken Sie den Druckknopf **Run Geocoder** an. Das Fenster **Run Geocoder** wird geöffnet.
2. Wenn Sie den Standard-Geocodierer verwenden wollen, lassen Sie das Feld **Name**, in dem der Name dieses Standard-Codierers angezeigt wird, unverändert. Wählen Sie andernfalls mit dem Feld den gewünschten Geocodierer aus.
3. Inaktivieren Sie Objekte, die die Leistung des Geocodierers beeinträchtigen könnten:
 - Wenn die auszufüllende Spalte einen Index hat, wählen Sie das Markierungsfeld **Temporarily disable spatial indexes during geocoding process** aus.
 - Wenn Auslöser zum Aktivieren der schrittweise steigenden Geocodierung für diese Spalte festgelegt wurden, wählen Sie das Markierungsfeld **Temporarily disable spatial triggers during geocoding process** aus. Der Index und die Auslöser werden automatisch reaktiviert, wenn Sie **OK** im Fenster **Run Geocoder** anklicken.
4. Geben Sie mit dem Drehknopf **Commit scope** an, wie viele Datensätze geocodiert werden sollen, bevor DB2 die Datensätze festschreibt. Wenn DB2 beispielsweise 100 geocodierte Datensätze in einem Arbeitsgang festschreiben soll, geben Sie die Zahl 100 ein.

Tipp: Wenn DB2 die Datensätze erst festschreiben soll, nachdem alle Datensätze verarbeitet wurden, geben Sie Null ein.
5. Geben Sie mit den Feldern in der Auswahlgruppe **Geocoder parameters** an, wie der Geocodierer arbeiten soll:
 - Geben Sie mit dem Drehknopf **Precision level** an, wie genau (in Prozent) die Übereinstimmung zwischen den Quellendatensätzen und den entsprechenden Bezugsdaten sein soll. Weitere Informationen zur Genauigkeit finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53.

- Wenn Sie einen Geocodierer eines anderen Anbieters verwenden und mit den von diesem Geocodierer unterstützten Merkmalen arbeiten wollen, legen Sie diese Merkmale über das Feld **Properties** fest.
- Wenn nur ein Teil der Zeilen in der ausgewählten Tabelle geocodiert werden soll, wählen Sie mit dem Feld **WHERE clause** eine Klausel **SELECT WHERE** aus, die die Kriterien für die gewünschten Zeilen angibt. Diese Klausel kann auf beliebige Spalten in der Tabelle verweisen.

Geben Sie nur die Kriterien ein. Lassen Sie das Schlüsselwort **WHERE** weg. Beispiel: Wenn die Tabelle eine Spalte **STATE** enthält und nur die Zeilen geocodiert werden sollen, die in dieser Spalten den Wert **MA** enthalten, geben Sie ein:

```
STATE='MA'
```

6. Klicken Sie **OK** an, um den Geocodierer auszuführen.

Daten importieren und exportieren

Dieser Abschnitt beschreibt die Prozesse zum Importieren und Exportieren von Daten sowie die Verwendung der Steuerzentrale für folgende Aufgaben:

- Importieren von Daten aus einer Datenaustauschdatei in eine neue oder vorhandene Tabelle
- Importieren von Daten aus einer Datenaustauschdatei in eine vorhandene Tabelle
- Exportieren von Daten aus einer Tabelle in eine Datenaustauschdatei

Informationen zum Importieren und Exportieren

Dieser Abschnitt listet Gründe für das Importieren oder Exportieren räumlicher Daten auf. Außerdem enthält er eine Beschreibung der Datenaustauschdateien, die als Schnittstelle zwischen der Quelle und dem Ziel des Importvorgangs dienen.

Sie können mit dem Spatial Extender räumliche Daten aus Datenaustauschdateien importieren oder in Datenaustauschdateien exportieren. Betrachten Sie die folgenden Szenarien:

- Ihr GIS enthält räumliche Daten, die Ihre Büros, Ihre Kunden und andere Geschäftsdaten darstellen. Sie wollen diese Daten mit räumlichen Daten zur Infrastrukturumgebung Ihrer Organisation ergänzen — Städte, Straßen, Verkehrsknotenpunkte etc. Die gewünschten Daten sind über einen Anbieter von Karten verfügbar. Sie können mit dem Spatial Extender diese Daten aus einer von dem Anbieter bereitgestellten Datenaustauschdatei importieren.
- Sie wollen räumliche Daten aus einem Oracle-System in Ihr Spatial Extender-GIS migrieren. Zunächst laden Sie mit einem Oracle Dienstprogramm die Daten in eine Datenaustauschdatei. Anschließend importieren Sie mit dem Spatial Extender die Daten aus dieser Datei in die Datenbank, die Sie für räumliche Operationen aktiviert haben.

- Sie wollen mit einem GIS-Browser visuelle Darstellungen von räumlichen Informationen zu Kunden anzeigen. Der Browser benötigt lediglich Daten, mit denen er arbeiten kann; eine Verbindung zu einer Datenbank ist nicht erforderlich. Sie können mit dem Spatial Extender die Daten in eine Datenaustauschdatei exportieren und anschließend mit einem Browser-Dienstprogramm die Daten in den Browser laden.

Die Steuerzentrale unterstützt zwei Arten von Datenaustauschdateien für den Spatial Extender: Formdateien und ESRI_SDE-Transferdateien. Formdateien werden häufig zum Importieren von Daten aus Dateisystemen verwendet und zum Exportieren von Daten in Dateien, die in Dateisysteme geladen werden sollen. ESRI_SDE-Transferdateien werden häufig zum Importieren von Daten aus ESRI-Datenbanken verwendet.

Daten von der Datenbankebene in eine neue oder eine vorhandene Tabelle importieren

Dieser Abschnitt enthält eine Übersicht über die Schritte zum Importieren von Daten aus einer Form- oder ESRI_SDE-Transferdatei in eine neue oder bereits vorhandene Tabelle unter Verwendung des Ordners **Databases** der Steuerzentrale. Im Anschluss an diese Übersicht finden Sie Details zur Durchführung der einzelnen Schritte.

Wenn Sie eine Gruppe von ESRI-Formdarstellungen importieren, erhalten Sie mindestens zwei Dateien. Beide Dateien haben denselben Namen, weisen jedoch unterschiedliche Dateierweiterungen auf. Die Dateierweiterungen der beiden Dateien, die Sie immer erhalten, lauten .shp und .shx.

Um die Dateien für eine Gruppe von Formdarstellungen zu erhalten, geben Sie in das Feld **File name** im Fenster **Import Spatial Data** den gemeinsamen Namen der Dateien ein. Geben Sie keine Dateierweiterung ein. Auf diese Weise stellen Sie sicher, dass alle Dateien, die Sie benötigen – die Datei .shp, die Datei .shx sowie alle sonstigen eventuell dazugehörigen Dateien – importiert werden.

Gehen Sie beispielsweise davon aus, dass eine Gruppe von ESRI-Formdarstellungen in Dateien mit dem Namen Lakes.shp und Lakes.shx gespeichert ist. Beim Importieren dieser Darstellungen würden Sie in das Feld **File name** nur "Lakes" eingeben.

SDE-Transferdateien haben einen Namen, aber keine Dateierweiterung. Geben Sie daher beim Importieren einer SDE-Transferdatei den Namen, aber keine Dateierweiterung in das Feld **File name** ein. Entsprechend geben Sie auch in das Feld **Exception file** im Fenster **Import Spatial Data** nicht die Erweiterung der Datei ein, die Sie angeben wollen. Geben Sie nur den Namen der Datei ein.

Informationen darüber, welche Berechtigungen zum Importieren von Formdaten erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 105. Informationen darüber, welche Berechtigungen zum Importieren von ESRI_SDE-Daten erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 103.

Übersicht über die Schritte zum Importieren von Daten in eine neue oder vorhandene Tabelle:

1. Öffnen Sie das Fenster **Import Spatial Data**.
2. Geben Sie den Pfad, den Namen und das Format der Datei an, die die zu importierenden Daten enthält.
3. Geben Sie an, wie viele Datensätze vor jedem Festschreiben geocodiert werden sollen.
4. Wenn Sie räumliche Daten in eine zu erstellende Tabelle importieren wollen, geben Sie einen Namen für diese Tabelle und einen Namen für die Spalte, in der die Daten gespeichert werden sollen, an. Wenn Sie räumliche Daten in eine vorhandene Tabelle importieren, geben Sie an, in welcher Spalte die Daten gespeichert werden sollen.
5. Geben Sie an, welches räumliche Bezugssystem den Daten zugeordnet werden soll.
6. Geben Sie eine Datei an, in der die Datensätze gesammelt werden sollen, die nicht importiert werden konnten.
7. Weisen Sie den Spatial Extender an, die Daten zu importieren und, falls Sie über dieses Fenster eine Tabelle definiert hatten, eine Tabelle zu erstellen und die Spalte, für die die Daten bestimmt sind, als Schicht zu registrieren.

Details zu den Schritten zum Importieren von Daten in eine neue oder vorhandene Tabelle:

1. Öffnen Sie das Fenster **Import Spatial Data**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Databases** unter dem Server finden, auf dem der Spatial Extender ausgeführt wird.
 - b. Klicken Sie den Ordner **Databases** an. Die Datenbanken werden im Inhalts-Teilfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die Datenbank an, in die die Daten importiert werden sollen, und klicken Sie **Spatial Extender** → **Import Spatial Data** im Kontextmenü an. Das Fenster **Import Spatial Data** wird geöffnet.
2. Geben Sie den Pfad, den Namen und das Format der Datei an, die die zu importierenden Daten enthält:
 - a. Geben Sie im Feld **File name** den Pfad und den Namen der Datei an.

- b. Geben Sie das Format im Feld **File format** an. Das Format kann eines der folgenden sein:

Shape Dies ist der Standardwert.

ESRI_SDE

Wenn Sie dieses Format angeben, enthält das Feld **Spatial reference name** standardmäßig den Namen des räumlichen Bezugssystems, das diesem Format zugeordnet ist.

3. Geben Sie im Feld **Commit scope** die Anzahl der Datensätze an, die vor jedem Festschreiben importiert werden sollen. Wenn DB2 beispielsweise 100 Datensätze in einem Arbeitsgang festschreiben soll, geben Sie die Zahl 100 ein.

Tip: Wenn DB2 die Datensätze erst festschreiben soll, nachdem alle Datensätze verarbeitet wurden, geben Sie Null ein.

4. Geben Sie die Tabelle und die Spalte an, für die die Daten bestimmt sind.
- a. Geben Sie im Feld **Layer schema** das Schema für die Tabelle an, in die die Daten importiert werden sollen.
- b. Geben Sie die Tabelle und die Spalte an:
- Wenn die Tabelle noch nicht vorhanden ist:
 - 1) Geben Sie im Feld **Layer table** einen Namen für die Tabelle ein.
 - 2) Geben Sie im Feld **Layer column** einen Namen für die Spalte ein, die die importierten Daten enthalten soll. Der Spatial Extender registriert diese Spalte automatisch als Schicht.
 - Wenn die Tabelle bereits vorhanden ist:
 - 1) Geben Sie im Feld **Layer table** die Tabelle an. Die Tabelle muss bereits die Spalte enthalten, in der die importierten Daten gespeichert werden sollen. Darüber hinaus muss diese Tabellenspalte bereits als Schicht registriert worden sein.
 - 2) Geben Sie im Feld **Layer column** den Namen der Spalte an, für die die importierten Daten bestimmt sind.
5. Geben Sie im Feld **Spatial reference name** das räumliche Bezugssystem ein, das diesen Daten zugeordnet werden soll bzw. wählen Sie dieses Bezugssystem aus. (Wenn die Daten aus einer ESRI_SDE-Transferdatei stammen, wird der Name des zugeordneten räumlichen Bezugssystems automatisch in dem Feld angezeigt.)
6. Geben Sie im Feld **Exception file** den Pfad und den Namen für eine neue Datei ein, in der die Datensätze gesammelt werden können, die nicht importiert werden konnten. Später können Sie diese Datensätze korrigieren und aus dieser Datei importieren.

Der Spatial Extender erstellt diese Datei; Sie brauchen keine bereits vorhandene Datei auszuwählen.

7. Klicken Sie **OK** an, um die Daten zu importieren. Wenn Sie einen Namen für eine Tabelle angegeben haben, die noch nicht vorhanden ist, wird diese Tabelle ebenfalls erstellt, und die Spalte, für die die Daten bestimmt sind, wird als Schicht registriert. Darüber hinaus wird die angegebene Ausnahmedatei erstellt.

Daten von der Tabellenebene in eine vorhandene Tabelle importieren

Dieser Abschnitt enthält eine Übersicht über die Schritte zum Importieren von Daten aus einer Form- oder ESRI_SDE-Transferdatei in eine bereits vorhandene Tabelle unter Verwendung des Ordners **Tables** der Steuerzentrale. Im Anschluss an diese Übersicht finden Sie Details zur Durchführung der einzelnen Schritte.

Informationen darüber, welche Berechtigungen zum Importieren von Formdaten erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 105. Informationen darüber, welche Berechtigungen zum Importieren von ESRI_SDE-Daten erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 103.

Übersicht über die Schritte zum Importieren von Daten in eine vorhandene Tabelle:

1. Öffnen Sie das Fenster **Import Spatial Data**.
2. Geben Sie den Pfad und den Namen der Datei an, die die zu importierenden Daten enthält.
3. Geben Sie an, wie viele Datensätze vor jedem Festschreiben geocodiert werden sollen.
4. Geben Sie die Spalte an, die die zu importierenden räumlichen Daten enthält.
5. Geben Sie an, welches räumliche Bezugssystem diesen Daten zugeordnet werden soll.
6. Geben Sie eine Datei an, in der Datensätze gesammelt werden sollen, die nicht importiert werden konnten.
7. Weisen Sie den Spatial Extender an, die Daten zu importieren und, falls Sie eine bisher noch nicht vorhandene Spalte definiert hatten, diese Spalte zu erstellen und als Schicht zu registrieren.

Details zu den Schritten zum Importieren von Daten in eine vorhandene Tabelle:

1. Öffnen Sie das Fenster **Import Spatial Data**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Tables** für die Datenbank finden, in die Sie Daten importieren wollen.
 - b. Klicken Sie den Ordner **Tables** an. Die Tabellen werden im Inhaltsteilfenster auf der rechten Seite des Fensters angezeigt.

- c. Klicken Sie mit der rechten Maustaste die Tabelle an, in die die Daten importiert werden sollen, und klicken Sie **Spatial Extender** —> **Import Spatial Data** im Kontextmenü an. Das Fenster **Import Spatial Data** wird geöffnet.
2. Geben Sie im Feld **File name** den Pfad und den Namen der Datei an, die die zu importierenden Daten enthält.
3. Geben Sie im Feld **Commit scope** die Anzahl der Datensätze an, die vor jedem Festschreiben importiert werden sollen. Wenn DB2 beispielsweise 100 Datensätze in einem Arbeitsgang festschreiben soll, geben Sie die Zahl 100 ein.

Tip: Wenn DB2 die Datensätze erst festschreiben soll, nachdem alle Datensätze verarbeitet wurden, geben Sie Null ein.

4. Geben Sie die Spalte an, die die zu importierenden räumlichen Daten enthält.
 - Wenn die Spalte in der Tabelle noch nicht vorhanden ist, geben Sie im Feld **Layer column** einen Namen für die Spalte ein.
 - Wenn die Spalte bereits vorhanden ist, wählen Sie den Namen der Spalte im Feld **Layer column** aus bzw. geben Sie ihn ein.
5. Geben Sie im Feld **Spatial reference name** das räumliche Bezugssystem ein, das den importierten Daten zugeordnet werden soll.
 - Wenn Sie einer Tabelle eine Spalte hinzufügen, geben Sie den Namen des räumlichen Bezugssystems ein bzw. wählen Sie ihn aus.
 - Wenn die importierten Daten für eine bereits vorhandene Spalte bestimmt sind, lassen Sie das Feld **Spatial reference name** unverändert. Das Feld zeigt den Namen des standardmäßigen räumlichen Bezugssystems an.
6. Geben Sie im Feld **Exception file** den Pfad und den Namen für eine neue Datei ein, in der die Datensätze gesammelt werden können, die nicht importiert werden konnten. Später können Sie diese Datensätze korrigieren und aus dieser Datei importieren.
Der Spatial Extender erstellt diese Datei; geben Sie keine bereits vorhandene Datei an.
7. Klicken Sie **OK** an, um die Daten zu importieren. Wenn Sie eine noch nicht vorhandene Spalte angegeben haben, wird diese Spalte erstellt und als Schicht registriert. Darüber hinaus wird die angegebene Ausnahmedatei erstellt.

Daten in eine Formdatei exportieren

Dieser Abschnitt enthält eine Übersicht über die Schritte zum Exportieren von Daten in eine Formdatei. Im Anschluss an diese Übersicht finden Sie Details zur Durchführung der einzelnen Schritte.

Informationen darüber, welche Berechtigungen für diese Schritte erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 101.

Übersicht über die Schritte zum Exportieren von Daten in eine Formdatei:

1. Öffnen Sie das Fenster **Export Spatial Data**.
2. Geben Sie die Spalte an, die die zu exportierenden räumlichen Spalten enthält.
3. Wenn Sie eine Untermenge der Datenzeilen exportieren wollen, kennzeichnen Sie diese Untermenge für den Spatial Extender.
4. Geben Sie den Pfad und den Namen der Datei an, in die Sie Daten exportieren wollen.
5. Weisen Sie den Spatial Extender an, die Daten zu exportieren.

Details zu den Schritten zum Exportieren von Daten in eine Formdatei:

1. Öffnen Sie das Fenster **Export Spatial Data**.
 - a. Heben Sie im Fenster **Control Center** die Komprimierung des Objektbaums auf, bis Sie den Ordner **Tables** oder den Ordner **Views** in der Datenbank finden, die die räumlichen Daten enthält:
 - b. Klicken Sie den Ordner **Tables** oder **Views** an. Tabellen oder Sichten werden im Inhalts-Teilfenster auf der rechten Seite des Fensters angezeigt.
 - c. Klicken Sie mit der rechten Maustaste die Tabelle oder Sicht an, die die zu exportierenden Daten enthält, und klicken Sie **Spatial Extender** → **Export Spatial Data** im Kontextmenü an. Das Fenster **Export Spatial Data** wird geöffnet.
2. Geben Sie im Feld **Layer column** den Namen der Spalte an, die die zu exportierenden räumlichen Daten enthält.
3. Wenn Sie nur eine Untermenge der Tabellenzeilen exportieren wollen, geben Sie im Feld **WHERE clause** eine WHERE-Klausel ein, die die Kriterien für die gewünschten Zeilen angibt. Sie können in dieser Klausel nur auf Spalten in der Tabelle oder Sicht verweisen, aus der Sie Daten exportieren.

Geben Sie nur die Kriterien ein. Lassen Sie das Schlüsselwort WHERE weg. Beispiel: Wenn die Tabelle bzw. Sicht eine Spalte STATE enthält und nur die Zeilen geocodiert werden sollen, die in dieser Spalten den Wert MA enthalten, geben Sie ein:

```
STATE='MA'
```

4. Geben Sie im Feld **File name** den Pfad und den Namen der Datei ein, in die Sie Daten exportieren.
5. Klicken Sie **OK** an, um die Daten zu exportieren.

Kapitel 6. Räumliche Indizes erstellen

In diesem Kapitel wird beschrieben, wie über die Steuerzentrale ein Index für Ihre räumlichen Daten erstellt wird.

Nachdem Sie die räumlichen Daten mit Ihren Daten ausgefüllt haben, können Sie einen räumlichen Index erstellen. Typische Indexstrukturen wie beispielsweise die B-Baumstruktur führen lineare, eindimensionale Sortierungen für Tabellendaten durch. Für räumliche Operationen aktivierte Tabellendaten werden nicht als einzelner Eintrag gespeichert, sondern sind zweidimensional. Räumliche Geometrien wie Polygone bestehen beispielsweise aus verschiedenen Koordinatenwerten in einer räumlichen Spalte oder Schicht. Da ein B-Baumstruktur-Index keine Typen von räumlichen Daten verarbeiten kann, hat der Spatial Extender eine spezielle Indexierungstechnologie erstellt, die als *Gitterindex* bezeichnet wird. Dieser Gitterindex basiert auf dem B-Baumstruktur-Index, der entsprechend erweitert wurde, um auch zweidimensionale Daten verarbeiten und Indexierungen mit räumlichen Spalten ausführen zu können. Der Gitterindex unterstützt drei Schichten und ermöglicht eine hohe Leistung mit einer Vielzahl verschiedener Objekte, Größen und Verteilungen von Daten. Weitere Informationen zu räumlichen Indizes finden Sie in „Kapitel 12. Räumliche Indizes“ auf Seite 147.

Informationen darüber, welche Berechtigungen zum Erstellen eines räumlichen Index erforderlich sind, finden Sie im Abschnitt „Autorisierung“ auf Seite 96.

Über die Steuerzentrale einen räumlichen Index erstellen

So erstellen Sie über die Steuerzentrale einen räumlichen Index:

1. Wählen Sie im Objektbaum den Ordner **Tables** aus. Alle vorhandenen Tabellen werden im Inhalts-Teilfenster angezeigt.
2. Klicken Sie im Inhalts-Teilfenster mit der rechten Maustaste die Tabelle an, für die Sie einen Index erstellen wollen, und klicken Sie **Spatial Extender** → **Spatial Indexes** im Kontextmenü an. Das Fenster **Spatial Indexes** wird geöffnet.
3. Klicken Sie im Fenster **Spatial Indexes** die Option **Create** an. Das Fenster **Create Spatial Index** wird geöffnet.
4. Geben Sie im Feld **Name** den Namen des neuen räumlichen Index ein, den Sie erstellen wollen.

Anmerkung: Die Auswahl eines Schemennamen für einen Index ist im Spatial Extender nicht zulässig. Der Spatial Extender fügt den Schemennamen automatisch ein und erstellt einen vollständig qualifizierten Namen.

5. Wählen Sie im Feld **Layer column** die Schicht aus, für die Sie einen Index erstellen wollen. Pro Schicht ist nur ein räumlicher Index zulässig.

Eine Schicht ist eine räumliche Spalte, die für den Spatial Extender definiert oder registriert wurde.

6. Geben Sie in den Feldern **Grid size** die Größe des Gitters ein, die Sie den einzelnen Feldern zuordnen wollen.

Die Gitterstufen **Finest**, **Middle** und **Coarsest** werden durch Vergrößern der Zellengröße eingegeben. Somit muss die zweite Schicht eine höhere Zellengröße haben als die erste und die dritte Schicht eine höhere als die zweite.

Zum Einstellen einer Gittergröße im Feld **Finest**, im Feld **Middle** oder im Feld **Coarsest** drücken Sie den Aufwärtspfeil bzw. den Abwärtspfeil auf der Tastatur Ihrer Workstation. Drücken Sie den Aufwärtspfeil, um die Größe um ein Zehntelgrad zu erhöhen; drücken Sie den Abwärtspfeil, um die Größe um ein Zehntelgrad zu reduzieren.

Sie müssen nicht alle drei Gittergrößen angeben.

Größe der Gitterzellen festlegen

Die richtige Gittergröße wird über "Versuch und Irrtum" festgelegt. Es wird empfohlen, die Gittergröße in Relation zur ungefähren Größe des Objekts, das indexiert werden soll, festzulegen. Gittergrößen, die zu groß oder zu klein festgelegt wurden, können zu einer Verschlechterung der Leistung führen. Wenn die Größe zu klein festgelegt wird, wirkt sich dies auf das Verhältnis Schlüssel/Objekt bei der Indexsuche aus. In diesem Fall werden zu viele Schlüssel erstellt, und es wird eine große Anzahl Kandidaten zurückgegeben. Für Gittergrößen, die zu groß gesetzt wurden, gibt der Anfangsindex eine kleine Anzahl von Kandidaten zurück, aber die Leistung kann sich bei der endgültigen Tabellensuche verschlechtern.

Weitere Informationen zur Auswahl der Größe von Gitterzellen und der Anzahl von Gitterstufen finden Sie im Abschnitt „Gitterzellengröße auswählen“ auf Seite 154.

Kapitel 7. Räumliche Informationen abrufen und analysieren

Nach dem Konstruieren der räumlichen Indizes stehen die räumlichen Tabellen zum Arbeiten zur Verfügung. In diesem Kapitel werden Informationen zum Abrufen und Analysieren räumlicher Daten beschrieben. Das Kapitel enthält eine Übersicht über die verschiedenen Methoden zum Abrufen sowie Beispiele zu Tabellenabfragen, die räumliche Funktionen verwenden.

Methoden zum Ausführen einer räumlichen Analyse

Sie können eine räumliche Analyse mit SQL und räumlichen Funktionen mit einer der folgenden Programmierumgebungen durchführen:

- Mit einem Geobrowser (z. B. ESRI ArcExplorer).

Weitere Information zur Verwendung des ArcExplorer finden Sie unter *Using ArcExplorer*; dieses Dokument ist über die ESRI Web-Site verfügbar unter der Adresse <http://www.esri.com>.

- Mit interaktiven SQL-Anweisungen

Interaktive SQL-Anweisungen können Sie über die DB2-Befehlszentrale, das DB2-Befehlsfenster oder den Befehlszeilenprozessor eingeben.

- Mit benutzerentwickelten Anwendungen (z. B. ODBC, JDBC und eingebettetem SQL).

Räumliche Abfrage erstellen

In diesem Abschnitt wird das Erstellen räumlicher Abfragen, die räumliche Funktionen und Prädikate verwenden, beschrieben.

Räumliche Funktionen und SQL

Der Spatial Extender umfasst Funktionen, die verschiedene Operationen mit räumlichen Daten ausführen. Die Beispiele in diesem Abschnitt zeigen, wie Sie mit den räumlichen Funktionen eigene räumliche Abfragen erstellen können.

Tabelle 3 zeigt eine Liste der räumlichen Funktionen und der Arten von Operationen, die Sie ausführen können.

Tabelle 3. Räumliche Funktionen und Operationen

Funktionstyp	Beispiel zur Operation
Berechnung	Berechnet die Entfernung zwischen zwei Punkten

Tabelle 3. Räumliche Funktionen und Operationen (Forts.)

Funktionstyp	Beispiel zur Operation
Vergleich	Findet alle Kunden innerhalb eines Überflutungsbereichs
Datenaustausch	Umwandeln von Daten in unterstützte Formate
Umsetzung	Hinzufügen eines Fünf-Meilen-Radius zu einem Punkt

Weitere Informationen zu räumlichen Funktionen finden Sie in den Abschnitten „Kapitel 13. Geometrien und zugeordnete räumliche Funktionen“ auf Seite 157 und „Kapitel 14. Räumliche Funktionen für SQL-Abfragen“ auf Seite 197.

Beispiel 1: Vergleich

Die folgende Abfrage findet die durchschnittliche Entfernung vom Kunden bis zum Kaufhaus. Die in diesem Beispiel verwendeten räumlichen Funktionen sind ST_Distance and 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
```

Beispiel 2: Datenaustausch

Die folgende Abfrage findet die Kundenstandorte für die Kunden in der Gegend der San Francisco Bay. Die in diesem Beispiel verwendeten räumlichen Funktionen sind ST_AsText (Datenaustausch) und ST_Within. ST_AsText wandelt die räumlichen Daten in der Spalte c.location in das Format OGC TEXT um.

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

Beispiel 3: Berechnung

Die folgende Abfrage findet alle Straßen, die länger als 10,5 Meilen sind. Die in diesem Beispiel verwendete räumliche Funktion ist ST_Length.

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

Beispiel 4: Umsetzung

Diese Abfrage findet die Kunden, die in einem Überflutungsgebiet oder innerhalb von 2 Meilen von den Grenzen eines Überflutungsgebiets wohnen. Die in diesem Beispiel verwendeten räumlichen Funktionen sind ST_Buffer (Umsetzung) und ST_Within. Die Variable :floodzone ist eine Host-Variable in einem in C/C++ implementierten eingebetteten SQL-Programm.

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

Räumliche Prädikate und SQL

Mit einer speziellen Gruppe räumlicher Funktionen, die als räumliche Prädikate bezeichnet werden, kann die Leistung gesteigert werden. Räumliche Prädikate wie beispielsweise `ST_Overlaps`, das zwei Polygone vergleicht, um festzustellen, ob sie sich überlappen, können im Hinblick auf Zeit- und Speicheranforderungen sehr aufwendig sein. Optimierungstechniken zum Minimieren der Ausführungskosten sind daher von großer Bedeutung. Der DB2 Query Optimizer verwendet den räumlichen Index zum Verbessern der Abfrageleistung, wenn Sie mit den räumlichen Prädikaten entsprechend den weiter hinten in diesem Kapitel beschriebenen Regeln arbeiten.

Weitere Informationen zu räumlichen Prädikaten finden Sie im Abschnitt „Prädikatfunktionen“ auf Seite 173. Die zum Nutzen des räumlichen Index verwendeten räumlichen Prädikate sind:

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

Eine vollständige Liste alle räumlichen Funktionen und Prädikate finden Sie in „Kapitel 14. Räumliche Funktionen für SQL-Abfragen“ auf Seite 197.

Regeln zur Nutzung des Index

Die folgenden Regeln gelten, wenn Sie räumliche Abfragen mit räumlichen Prädikaten optimieren wollen:

- Das Prädikat muss in der WHERE-Klausel verwendet werden.
- Das Prädikat muss auf der linken Seite des Vergleichs stehen. Beispiel:
`WHERE db2gse.ST_Within(c.location,:BayArea)=1`
- Bei Vergleichen auf Gleichheit ist die ganzzahlige Konstante 1 zu verwenden.
`WHERE db2gse.ST_Within(c.location,:BayArea)=1`
- In dem Prädikat muss eine räumliche Spalte als Suchziel verwendet werden, und es muss mit dieser Spalte ein räumlicher Index erstellt werden.

Beispiele zur Nutzung des Index

Tabelle 4 zeigt richtige und falsche Beispiele zum Erstellen räumlicher Abfragen nur Nutzung des räumlichen Index.

Tabelle 4. Regeln zur Nutzung des Index

Räumliche Abfrage	Regel verletzt
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=1</pre>	In diesem Beispiel wurde keine Regel verletzt.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Distance(c.location,:SanJose)<10</pre>	In diesem Beispiel wurde keine Regel verletzt.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Length(c.location)>10</pre>	ST_Length ist kein räumliches Prädikat.
<pre>SELECT * FROM customers c WHERE 1=db2gse.ST_Within(c.location,:BayArea)</pre>	Das Prädikat muss auf der linken Seite des Vergleichs stehen.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=2</pre>	Bei Vergleichen auf Gleichheit ist die ganzzahlige Konstante 1 zu verwenden.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(:SanJose,:BayArea)=1</pre>	In dem Prädikat muss eine räumliche Spalte als Suchziel verwendet werden, und es muss mit dieser Spalte ein räumlicher Index erstellt werden. (SanJose und BayArea sind keine räumlichen Spalten, daher kann ihnen kein räumlicher Index zugeordnet werden.)

Kapitel 8. Anwendungen für Spatial Extender schreiben

In diesem Kapitel wird beschrieben, wie mit dem Spatial Extender-Beispielprogramm Anwendungen zum Arbeiten und Anpassen räumlicher Informationen geschrieben werden. Folgende Punkte werden beschrieben:

- Beispielprogramm verwenden
- Die Schritte des Beispielprogramms

Das Beispielprogramm verwenden

Das Spatial Extender-Beispielprogramm vereinfacht das Schreiben von Anwendungen. Mit dem Beispielprogramm haben Sie folgende Möglichkeiten:

- Räumliche Routineaufgaben automatisieren
- Mustercode ausschneiden und in eigene Anwendungen einfügen
- Erforderliche Schritte zum Erstellen und Verwalten einer für räumliche Funktionen aktivierten Datenbank verstehen

Verwenden Sie das Beispielprogramm zum Codieren komplexer Aufgaben für den Spatial Extender, beispielsweise zum Schreiben einer Anwendung, die eine Datenbankschnittstelle zum Aufruf gespeicherter Spatial Extender-Funktionen verwendet. Mit dem Beispielprogramm können Sie Ihre Anwendungen kopieren und anpassen. Wenn Sie mit den Schritten zur Programmierung des Spatial Extender nicht vertraut sind, können Sie das Beispielprogramm ausführen, in dem die einzelnen Schritte ausführlich beschrieben sind. Zunächst müssen Sie jedoch das Beispielprogramm erstellen. Hierzu verwenden Sie die Beispiel-Make-Datei. Anweisungen zum Erstellen und Ausführen des Beispielprogramms finden Sie im Abschnitt „Installation überprüfen“ auf Seite 26.

Die Schritte des Beispielprogramms

Tabelle 5 auf Seite 74 zeigt die Schritte des Beispielprogramms, die zugeordneten gespeicherten Prozeduren und eine Beschreibung der einzelnen Schritte. Die C-Funktionen zum Aufrufen der gespeicherten Prozeduren sind in der Spalte "Aktion" von Tabelle 5 auf Seite 74 angegeben und in Klammern eingeschlossen. Weitere Informationen zu gespeicherten Prozeduren finden Sie in „Kapitel 9. Gespeicherte Prozeduren“ auf Seite 83. Das Beispielprogramm basiert auf den im Abschnitt „Szenario: Eine Versicherungsgesellschaft aktualisiert ihr GIS“ auf Seite 13 vorgestellten Szenarien.

Tabelle 5. Spatial Extender-Beispielprogramm

Schritte des Beispielprogramms	Aktion	Beschreibung
Aktivieren / Inaktivieren der räumlichen Datenbank	<ol style="list-style-type: none"> 1. Aktivieren der räumlichen Datenbank (gseEnableDB) 2. Inaktivieren der räumlichen Datenbank (gseDisableDB) 3. Aktivieren der räumlichen Datenbank (gseEnableDB) 	<ol style="list-style-type: none"> 1. Dies ist der erste erforderliche Schritt zur Verwendung des Spatial Extender. Eine für räumliche Operationen aktivierte Datenbank enthält spezielle Typen, eine Gruppe räumlicher Funktionen, einen neuen Indextyp und eine Gruppe von Verwaltungstabellen und Sichten. 2. Dieser Schritt wird normalerweise ausgeführt, wenn Sie die räumlichen Funktionen für die falsche Datenbank aktiviert haben. Wenn Sie eine räumliche Datenbank aktivieren, entfernen Sie eine Gruppe räumlicher Typen, eine Gruppe räumlicher Prädikate, einen neuen Indextyp und eine Gruppe von Verwaltungstabellen und Sichten. Anmerkung: Das Inaktivieren der Datenbank schlägt fehl, wenn Objekte erstellt wurden, die von den Objekten abhängen, die beim Aktivieren der Datenbank erstellt wurden. Wenn Sie beispielsweise eine Tabelle mit einer räumlichen Spalte des Typs ST_Point erstellen, schlägt das Inaktivieren der Datenbank fehl. Dieser Fehler tritt auf, da die Tabelle vom Typ ST_Point abhängt, der durch das Inaktivieren der Datenbank freigegeben werden soll. 3. Wie 1.

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Räumliche Bezugssysteme registrieren	<ol style="list-style-type: none"> 1. Registrieren des räumlichen Bezugssystems für die Spalte LOCATION der Tabelle CUSTOMERS (gseEnableSref) 2. Registrieren des räumlichen Bezugssystems für die Spalte LOCATION der Tabelle OFFICES (gseEnableSref) 3. Registrierung des räumlichen Bezugssystems für die Spalte LOCATION der Tabelle OFFICES aufheben (gseDisableSref) 4. Erneut Registrieren des räumlichen Bezugssystems für die Spalte ZONE der Tabelle OFFICES (gseEnableSref) 	<ol style="list-style-type: none"> 1. Dieser Schritt definiert ein neues räumliches Bezugssystem (SRS), das zum Interpretieren der räumlichen Daten der Tabelle CUSTOMERS verwendet werden soll. Ein räumliches Bezugssystem enthält Geometriedaten in einer Form, die in einer Spalte einer räumlich aktivierten Datenbank gespeichert werden kann. Nachdem das SRS für eine bestimmte Schicht registriert wurde, können die entsprechenden Koordinaten für diese Schicht in der zugeordneten Spalte der Tabelle CUSTOMERS gespeichert werden. 2. Dieser Schritt definiert ein neues räumliches Bezugssystem (SRS), das zum Interpretieren der räumlichen Daten der Schicht von OFFICES verwendet werden soll. Für jede Tabellenschicht muss ein SRS definiert worden sein. Die Schichten der Tabelle OFFICES erfordern eventuell die Zuordnung eines anderen SRS als für die Schichten der Tabelle CUSTOMERS. 3. Dieser Schritt wird ausgeführt, wenn Sie die falschen SRS-Parameter für die Schicht oder die räumliche Spalte angeben. Wenn Sie die Registrierung eines SRS für die Tabelle OFFICES aufheben, entfernen Sie die Definition mit ihren zugeordneten Parametern. 4. Dieser Schritt definiert ein neues räumliches Bezugssystem (SRS), das zum Interpretieren der räumlichen Daten der Schicht von OFFICES verwendet werden soll.

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Räumliche Tabellen erstellen	<ol style="list-style-type: none"> 1. Ändern der Tabelle CUSTOMERS durch Hinzufügen der Spalte LOCATION (gseSetupTables) 2. Erstellen der Tabelle OFFICES (gseSetupTables) 	<ol style="list-style-type: none"> 1. Die Tabelle CUSTOMERS stellt Geschäftsdaten dar, die seit mehreren Jahren in der Datenbank gespeichert wurden. Die Anweisung ALTER TABLE fügt eine neue Spalte (LOCATION) des Typs ST_Point hinzu. Diese Spalte wird durch eine Geocodierung der Adress-Spalten in einem späteren Schritt ausgefüllt. 2. Die Tabelle OFFICES stellt neben anderen Daten die Vertriebszone für jede Niederlassung eines Versicherungsunternehmens dar. Die gesamte Tabelle wird in einem späteren Schritt mit den attributiven Daten aus einer Nicht-DB2-Datenbank ausgefüllt. Dieser Schritt umfasst das Importieren attributiver Daten in die Tabelle OFFICES auf einer Datei SHAPE.
Registrieren der räumlichen Schichten	<ol style="list-style-type: none"> 1. Registrieren der Spalte LOCATION in der Tabelle CUSTOMERS als Schicht (gseRegisterLayer) 2. Aufheben der Registrierung der Spalte LOCATION der Tabelle CUSTOMERS als Schicht (gseUnregisterLayer) 3. Registrieren der Spalte ZONE der Tabelle OFFICES als Schicht (gseRegisterLayer) 	<p>Mit diesen Schritten werden die Spalten LOCATION und ZONE als Schichten im Spatial Extender registriert. Bevor eine räumliche Spalte ausgefüllt oder über die Spatial Extender-Dienstprogramme (z. B. den Geocodierer) aufgerufen werden kann, müssen Sie sie als Schicht registrieren. Sie können die Registrierung einer Schicht auch wieder aufheben, nachdem Sie die Schicht für die Dienstprogramme des Spatial Extender zugänglich gemacht haben. Die zugehörige Spalte bleibt auch nach der Aufhebung ihrer Registrierung als Schicht bestehen.</p>

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Ausfüllen der räumlichen Schichten	<ol style="list-style-type: none"> 1. Geocodieren der Adressdaten für die Spalte LOCATION der Tabelle CUSTOMERS (gseRunGC) 2. Laden der Tabelle OFFICES mit dem Append-Modus (gseImportShape) 3. Laden der Tabelle HAZARD_ZONE mit dem Create-Modus (gseImportShape) 	<ol style="list-style-type: none"> 1. Mit diesem Schritt wird durch den Aufruf des Geocodierer-Dienstprogramms eine Geocodierung im Stapelbetrieb ausgeführt. Eine Stapel-Geocodierung wird normalerweise ausgeführt, wenn ein erheblicher Anteil der Tabelle geocodiert oder erneut geocodiert werden muss. 2. Mit diesem Schritt wird die Tabelle OFFICES mit räumlichen Daten ausgefüllt, die in Form einer einer SHAPE-Datei vorliegen. Da die Tabelle OFFICES vorhanden ist und die Schicht OFFICES/ZONE registriert wurde, hängt das Lade-Dienstprogramm die neuen Datensätze an die vorhandene Tabelle an. 3. Mit diesem Schritt wird die Schicht HAZARD_ZONE mit räumlichen Daten ausgefüllt, die in Form einer einer SHAPE-Datei vorliegen. Da die Tabelle und die Schicht nicht vorhanden sind, erstellt das Lade-Dienstprogramm die Tabelle und registriert die Schicht, bevor die Daten geladen werden.
Registrieren des Geocodierers	<ul style="list-style-type: none"> • Registrieren des Geocodierers, sofern es sich nicht um den Standardgeocodierer handelt (gseRegisterGc) • Aufheben der Registrierung eines eventuell registrierten Geocodierers (gseUnregisterGc) • Registrieren des Geocodierers, sofern es sich nicht um den Standardgeocodierer handelt (gseRegisterGc) 	

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Räumliche Indizes aktivieren	<ol style="list-style-type: none"> 1. Aktivieren des räumlichen Index für die Spalte LOCATION der Tabelle CUSTOMERS (gseEnableIdx) 2. Aktivieren des räumlichen Index für die Spalte ZONE der Tabelle OFFICES (gseEnableIdx) 3. Aktivieren des räumlichen Index für die Spalte LOCATION der Tabelle OFFICES (gseEnableIdx) 4. Aktivieren des räumlichen Index für die Spalte BOUNDRY der Tabelle HAZARD_ZONE (gseEnableIdx) 	Mit diesen Schritten wird der räumliche Index für die Tabellen CUSTOMERS, OFFICES und HAZARD_ZONE aktiviert.
Automatische Geocodierung aktivieren	<ol style="list-style-type: none"> 1. Aktivieren der automatischen Geocodierung für die Spalten LOCATION und ADDRESS der Tabelle CUSTOMERS (gseEnableAutoGC) 	Mit diesem Schritt wird der automatische Aufruf des Geocodierers eingeschaltet. Durch die automatische Geocodierung werden die Spalten LOCATION und ADDRESS der Tabelle CUSTOMERS miteinander synchronisiert für spätere Operationen zum Einfügen und Aktualisieren.
Einfügen /Aktualisieren /Löschen der Tabelle CUSTOMERS	<ol style="list-style-type: none"> 1. Einfügen einiger Datensätze mit einer anderen Straße (gseInsDelUpd) 2. Aktualisieren einiger Datensätze mit einer neuen Adresse (gseInsDelUpd) 3. Löschen aller Datensätze aus der Tabelle (gseInsDelUpd) 	Diese Schritte demonstrieren eine Einfügung, eine Aktualisierung und eine Löschung in der Spalte LOCATION der Tabelle CUSTOMERS. Sobald die automatische Geocodierung aktiviert ist, werden Informationen aus der Spalte ADDRESS automatisch geocodiert, wenn sie in die Spalte LOCATION eingefügt bzw. aktualisiert werden. Dieser Prozess wurde im vorherigen Schritt aktiviert.
Automatische Geocodierung inaktivieren	<ol style="list-style-type: none"> 1. Inaktivieren der automatischen Geocodierung für die Schicht CUSTOMERS (gseDisableAutoGC) 2. Inaktivieren des räumlichen Index für die Schicht CUSTOMERS (gseDisableIdxCustomersLayer) 	Mit diesen Schritten wird der automatische Aufruf des Geocodierers und des räumlichen Index in Vorbereitung des nächsten Schritts inaktiviert (der nächste Schritt umfasst die erneute Geocodierung der gesamten Tabelle CUSTOMERS). Wenn Sie eine große Menge von Geodaten laden, sollten Sie vor dem Laden der Daten den räumlichen Index inaktivieren und ihn nach Abschluss des Ladevorgangs wieder aktivieren.

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Erneut Geocodieren der Tabelle CUSTOMERS	<ol style="list-style-type: none"> 1. Erneutes Geocodieren der Schicht CUSTOMERS mit einer niedrigeren Genauigkeitsstufe – 90% statt 100% (gseRunGC) 2. Erneut Aktivieren des räumlichen Index für die Schicht CUSTOMERS (gseEnableIdx) 3. Erneut Aktivieren der automatischen Geocodierung mit einer niedrigeren Genauigkeitsstufe – 90% statt 100% (gseEnableAutoGC) 	<p>Mit diesen Schritten werden der Geocodierer erneut im Stapelbetrieb ausgeführt, die automatische Geocodierung mit einer niedrigeren Genauigkeitsstufe erneut aktiviert und der räumliche Index und die automatische Geocodierung wieder aktiviert. Diese Aktion wird empfohlen, wenn der Administrator für die räumlichen Daten im Geocodierungsprozess einen hohen Fehleranteil feststellt. Wenn die Genauigkeitsstufe auf 100% eingestellt ist, kann die Geocodierung einer Adresse fehlschlagen, da in den Bezugsdaten keine entsprechende Adresse gefunden wird. Durch die Reduzierung der Genauigkeitsstufe hat der Geocodierer eine bessere Chance, entsprechende Daten zu finden. Nachdem die Tabelle im Stapelbetrieb geocodiert wurde, werden sowohl die automatische Geocodierung als auch der räumliche Index erneut aktiviert, um die schrittweise steigende Wartung des räumlichen Index und der räumlichen Spalte für spätere Einfüge- und Aktualisierungsoperationen zu ermöglichen.</p>
Erstellen einer Sicht und Registrieren ihrer räumlichen Spalten als Sichtschichten	<ol style="list-style-type: none"> 1. Erstellen einer Sicht, HIGHRISK_CUSTOMERS, basierend auf der Verknüpfung der Tabellen CUSTOMERS und HAZARD_ZONE (gseCreateView) 2. Registrieren der räumlichen Spalten der Sicht als Sichtschichten (gseRegisterLayer) 	<p>Mit diesen Schritten werden eine Sicht erstellt und ihre räumlichen Spalten als Sichtschichten registriert.</p>

Tabelle 5. Spatial Extender-Beispielprogramm (Forts.)

Schritte des Beispielprogramms	Aktion	Beschreibung
Räumliche Analyse ausführen	<ol style="list-style-type: none"> 1. Ermitteln der durchschnittlichen Entfernung von jeder Niederlassung (ST_Within, ST_Distance) 2. Ermitteln des durchschnittlichen Einkommens pro Kunde und der Prämie für jede Niederlassung (ST_Within) 3. Ermitteln der Kunden, die nicht durch eine vorhandene Niederlassung betreut werden (ST_Within) 4. Ermitteln der Gefahrenzonen, die jede Niederlassungszone überlappt (ST_Overlaps) 5. Ermitteln der nächsten Niederlassung von einem bestimmten Kundenstandort aus unter der Annahme, dass sich die Niederlassung im Zentrum der Niederlassungszone befindet (ST_Distance, ST_Centroid) 6. Ermitteln der Kunden, deren Standort nahe an der Grenze einer bestimmten Gefahrenzone liegt. (ST_Buffer, ST_Overlaps) 7. Ermitteln der Kunden mit hohem Risiko, die von einer bestimmten Niederlassung betreut werden <p>(Alle diese Schritte verwenden gseRunSpatialQueries)</p>	<p>Mit diesen Schritten wird eine räumliche Analyse durchgeführt mit den räumlichen Prädikaten und Funktionen in der DB2 SQL-Sprache. Der DB2 Query Optimizer nutzt nach Möglichkeit den räumlichen Index mit den räumlichen Spalten zur Verbesserung der Abfrageleistung.</p>
Exportieren der räumlichen Schichten in Dateien	Exportieren der Schicht highRiskCustomers (gseExportShape)	<p>Der Schritt zeigt ein Beispiel zum Exportieren der Ergebnisse Ihrer Abfrage in eine SHAPE-Datei. Das Exportieren der Abfrageergebnisse in ein anderes Dateiformat ermöglicht die Verwendung dieser Informationen durch ein Tool eines anderen Herstellers (z. B. ESRI ArcExplorer Java Version 3.0).</p>

Teil 2. Referenzmaterial

Kapitel 9. Gespeicherte Prozeduren

Dieses Kapitel dokumentiert die gespeicherten Prozeduren, mit deren Hilfe Sie ein geografisches Informationssystem mit dem Spatial Extender erstellen können. Wenn Sie den Spatial Extender über die Steuerzentrale aktivieren und verwenden, rufen Sie diese gespeicherten Prozeduren implizit auf. Wenn Sie beispielsweise **OK** in einem Spatial Extender-Fenster anklicken, ruft DB2 die diesem Fenster zugeordnete gespeicherte Prozedur bzw. die Prozeduren auf. Alternativ dazu können Sie die gespeicherten Prozeduren explizit in einem Anwendungsprogramm aufrufen. Es empfiehlt sich, in einem solchen Programm die Kopfdatei db2gse.h anzugeben. Diese Datei enthält die Makrodefinitionen für die Konstanten, die Sie den Parametern der gespeicherten Prozeduren zuordnen. Unter AIX sind diese Definitionen in dem Verzeichnis \$DB2INSTANCE/sqlib/include/ gespeichert. Unter Windows NT sind sie im Verzeichnis %DB2PATH%\include\ gespeichert.

Achtung:

Bei allen Zeichenfolgenkonstanten der Eingabeparameter für gespeicherte Prozeduren wird zwischen Groß- und Kleinschreibung unterschieden. Anhand der Tabellen in diesem Kapitel können Sie feststellen, welche Parameter diese Konstanten erfordern.

In Großbuchstaben müssen Sie alle Schemennamen, Tabellennamen, Sichtnamen, Spaltennamen oder Schichtnamen angeben, die Sie einem Parameter zuordnen.

Bevor Sie eine gespeicherte Prozedur implizit oder explizit aufrufen können, müssen Sie eine Verbindung zu der Datenbank haben, in der der Spatial Extender installiert ist. Die erste gespeicherte Prozedur, die Sie verwenden, ist db2gse.gse_enable_db. Sie aktiviert die Datenbank für räumliche Operationen. Sie können die weiteren gespeicherten Prozeduren erst verwenden, nachdem die Datenbank aktiviert wurde.

Die Implementierung der gespeicherten Prozeduren sind in der Bibliothek db2gse des Spatial Extender-Servers archiviert.

In der folgenden Liste können Sie die gespeicherten Prozeduren nach Namen oder nach den ausgeführten Tasks nachschlagen. Die erste Liste zeigt die Namen an:

- „db2gse.gse_disable_autogc“ auf Seite 86
- „db2gse.gse_disable_db“ auf Seite 89
- „db2gse.gse_disable_sref“ auf Seite 90
- „db2gse.gse_enable_autogc“ auf Seite 91
- „db2gse.gse_enable_db“ auf Seite 95
- „db2gse.gse_enable_idx“ auf Seite 96
- „db2gse.gse_enable_sref“ auf Seite 98
- „db2gse.gse_export_shape“ auf Seite 101
- „db2gse.gse_import_sde“ auf Seite 103
- „db2gse.gse_import_shape“ auf Seite 105
- „db2gse.gse_register_gc“ auf Seite 108
- „db2gse.gse_register_layer“ auf Seite 110
- „db2gse.gse_run_gc“ auf Seite 117
- „db2gse.gse_unregist_gc“ auf Seite 119
- „db2gse.gse_unregist_layer“ auf Seite 120

Die nächste Liste zeigt die von den gespeicherten Prozeduren ausgeführten Tasks.

- Einen Index für eine räumliche Spalte erstellen (siehe „db2gse.gse_enable_idx“ auf Seite 96).
- Ein räumliches Bezugssystem erstellen (siehe „db2gse.gse_enable_sref“ auf Seite 98).
- Einen Geocodierer inaktivieren, so dass er räumliche Spalten nicht mehr automatisch mit den entsprechenden Attributspalten synchronisieren kann (siehe „db2gse.gse_disable_autogc“ auf Seite 86).
- Unterstützung für räumliche Operationen in einer Datenbank inaktivieren (siehe „db2gse.gse_disable_db“ auf Seite 89).
- Ein räumliches Bezugssystem freigeben (siehe „db2gse.gse_disable_sref“ auf Seite 90).
- Eine Datenbank zur Unterstützung räumlicher Operationen aktivieren (siehe „db2gse.gse_enable_db“ auf Seite 95).
- Einen Geocodierer aktivieren, so dass er räumliche Spalten automatisch mit den entsprechenden Attributspalten synchronisiert (siehe „db2gse.gse_enable_autogc“ auf Seite 91).

- Eine Schicht und ihre zugeordnete Tabelle in eine Formdatei exportieren (siehe „db2gse.gse_export_shape“ auf Seite 101).
- Eine Schicht und ihre zugeordnete Tabelle aus einer ESRI_SDE-Übertragungsdatei importieren (siehe „db2gse.gse_import_sde“ auf Seite 103).
- Eine Schicht und ihre zugeordnete Tabelle aus einer Formdatei importieren (siehe „db2gse.gse_import_shape“ auf Seite 105).
- Einen anderen Geocodierer als den Standard-Codierer registrieren (siehe „db2gse.gse_register_gc“ auf Seite 108).
- Eine räumliche Spalte als Schicht registrieren (siehe „db2gse.gse_register_layer“ auf Seite 110).
- Einen Geocodierer im Stapelbetrieb ausführen (siehe „db2gse.gse_unregist_gc“ auf Seite 119).
- Die Registrierung eines anderen Geocodierers als des Standard-Codierers aufheben (siehe „db2gse.gse_unregist_layer“ auf Seite 120).
- Die Registrierung einer Schicht aufheben (siehe „db2gse.gse_unregist_layer“ auf Seite 120).

Informationen zu der Reihenfolge, in der Sie diese Tasks ausführen können, finden Sie in „Kapitel 1. Informationen zum Spatial Extender“ auf Seite 3 und „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Verwenden Sie diese gespeicherte Prozedur zum Freigeben oder zum vorübergehenden Inaktivieren der Auslöser, die für die Synchronisation einer räumlichen Spalte mit den entsprechenden attributiven Spalten sorgen. Es empfiehlt sich beispielsweise, die Auslöser zu inaktivieren, während die Werte in den Attributspalten im Stapelbetrieb geocodiert werden. Weitere Informationen hierzu finden Sie im Abschnitt „Informationen zur Geocodierung“ auf Seite 53.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseDisableAutoGc` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine entsprechende Berechtigung haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, in der die freigegebenen bzw. vorübergehend inaktivierten Auslöser definiert sind.
- Die Berechtigung CONTROL auf diese Tabelle.
- Die Berechtigungen ALTER, SELECT und UPDATE auf diese Tabelle.

Eingabeparameter

Tabelle 6. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_disable_autogc`.

Name	Datentyp	Beschreibung
operMode	SMALLINT	<p>Gibt an, ob die Auslöser freigegeben oder vorübergehend inaktiviert werden sollen.</p> <p>Freigegebene Auslöser haben keine Wirkung auf SQL-Anweisungen.</p> <p>Vorübergehend inaktivierte Auslöser können wiederhergestellt werden, ohne dass zuvor festgelegte Parameter erneut angegeben werden müssen.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p> <p>Kommentar: Verwenden Sie zum Freigeben von Auslösern das Makro <code>GSE_AUTOGC_DROP</code>. Verwenden Sie zum vorübergehenden Inaktivieren der Auslöser das Makro <code>GSE_AUTOGC_INVALIDATE</code>. Wenn Sie feststellen wollen, welche Werte diesen Makros zugeordnet sind, schlagen Sie in der Tabelle <code>db2gse.h</code> nach. Unter AIX ist diese Datei in dem Verzeichnis <code>\$DB2INSTANCE/sqlib/include/</code> gespeichert. Unter Windows NT ist sie im Verzeichnis <code>%DB2PATH%\include\</code> gespeichert.</p>
layerSchema	VARCHAR(30)	<p>Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle oder Sicht gehört.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_disable_autogc</code> aufgerufen wurde.</p>
layerTable	VARCHAR(128)	<p>Name der Tabelle, in der die freizugehenden bzw. vorübergehend zu inaktivierenden Auslöser definiert sind.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>

Tabelle 6. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_disable_autogc. (Forts.)

Name	Datentyp	Beschreibung
layerColumn	VARCHAR(128)	Name der räumlich aktivierten Spalte, die von den Auslösern verwaltet wird, die Sie freigeben oder vorübergehend inaktivieren wollen. Dieser Parameter kann keine Nullwerte enthalten.

Ausgabeparameter

Tabelle 7. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_disable_autogc.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_disable_db

Verwenden Sie diese gespeicherte Prozedur zum Entfernen von Ressourcen, die dem Spatial Extender das Speichern räumlicher Daten und die Unterstützung von Operationen mit diesen Daten ermöglichen.

Der Zweck dieser gespeicherten Prozedur ist es, das Lösen von Problemen zu erleichtern, die nach dem Aktivieren der Datenbank für räumliche Operationen auftreten können, aber *vor* dem Hinzufügen von räumlichen Tabellenspalten oder Daten. Es könnte beispielsweise nach dem Aktivieren einer Datenbank für räumliche Operationen entschieden werden, dass statt dessen der Spatial Extender für eine andere Datenbank verwendet werden soll. Solange keine räumlichen Spalten definiert oder räumliche Daten importiert wurden, können Sie diese gespeicherte Prozedur aufrufen, um alle räumlichen Ressourcen aus der ersten Datenbank zu entfernen.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseDisableDB` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der Berechtigungen SYSADM oder DBADM für die Datenbank haben, von der die Spatial Extender-Ressourcen entfernt werden sollen.

Ausgabeparameter

Tabelle 8. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_disable_db.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_disable_sref

Verwenden Sie diese gespeicherte Prozedur zum Freigeben eines räumlichen Bezugssystems. Beim Verarbeiten dieser gespeicherten Prozedur werden Informationen zum räumlichen Bezugssystem aus der Katalogsicht DB2GSE.SPATIAL_REF_SYS entfernt. Informationen zu dieser Sicht finden Sie im Abschnitt „DB2GSE.SPATIAL_REF_SYS“ auf Seite 145.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion gseDisableSref im Beispielpogramm.

Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Keine erforderlich.

Eingabeparameter

Tabelle 9. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_disable_sref.

Name	Datentyp	Beschreibung
srld	INTEGER	Numerische Kennung des freizugebenden räumlichen Bezugssystems. Dieser Parameter kann keine Nullwerte enthalten.

Ausgabeparameter

Tabelle 10. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_disable_sref.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

Einschränkung

Bevor Sie ein räumliches Bezugssystem freigeben können, müssen Sie die Registrierung aller Schichten, die es verwenden, aufheben. Wenn solche Schichten nicht registriert werden, wird die Anforderung, das räumliche Bezugssystem freizugeben, zurückgewiesen.

db2gse.gse_enable_autogc

Verwenden Sie diese gespeicherte Prozedur für folgende Aufgaben:

- Erstellen von Auslösern, die für die Synchronisierung einer räumlichen Spalte mit ihren entsprechenden Attributspalten sorgen. Jedes Mal, wenn Werte in die Attributspalten eingefügt bzw. darin aktualisiert werden, ruft ein Auslöser einen registrierten Geocodierer auf, um die eingefügten bzw. aktualisierten Werte zu geocodieren und die resultierenden Daten in der räumlichen Spalten abzulegen.
- Reaktivieren der Auslöser, nachdem sie vorübergehend inaktiviert wurden.
- Festlegen, welche Funktion zum Geocodieren der eingefügten und aktualisierten Werte verwendet wird.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseEnableAutoGC` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine entsprechende Berechtigung haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, in der die von dieser gespeicherten Prozedur erstellten Auslöser definiert sind.
- Die Berechtigung CONTROL auf diese Tabelle.
- Die Berechtigungen ALTER, SELECT und UPDATE auf diese Tabelle.

Eingabeparameter

Tabelle 11. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_enable_autogc`.

Name	Datentyp	Beschreibung
operMode	SMALLINT	<p>Dieser Wert gibt an, ob die Auslöser, die die Geocodierung starten, zum ersten Mal erstellt werden oder nach einem vorübergehenden Inaktivieren reaktiviert werden sollen.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p> <p>Kommentar: Verwenden Sie zum Erstellen der Auslöser das Makro <code>GSE_AUTOGC_CREATE</code>. Verwenden Sie zum Reaktivieren der Auslöser das Makro <code>GSE_AUTOGC_RECREATE</code>. Wenn Sie feststellen wollen, welche Werte diesen Makros zugeordnet sind, schlagen Sie in der Tabelle <code>db2gse.h</code> nach. Unter AIX ist diese Datei in dem Verzeichnis <code>\$DB2INSTANCE/sqlib/include/</code> gespeichert. Unter Windows NT ist sie im Verzeichnis <code>%DB2PATH%\include\</code> gespeichert.</p>
layerSchema	VARCHAR(30)	<p>Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle gehört.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_enable_autogc</code> aufgerufen wurde.</p>
layerTable	VARCHAR(128)	<p>Name der Tabelle, mit der die von dieser gespeicherten Prozedur erstellten oder reaktivierten Auslöser arbeiten sollen.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
layerColumn	VARCHAR(128)	<p>Name der räumlichen Spalte, die von den in dieser gespeicherten Prozedur erstellten bzw. reaktivierten Auslösern verwaltet werden soll.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>

Tabelle 11. Eingabeparameter für die gespeicherte Prozedur *db2gse.gse_enable_autogc*. (Forts.)

Name	Datentyp	Beschreibung
gcId	INTEGER	<p>Kennung des Geocodierers, der von den in dieser gespeicherten Prozedur erstellten bzw. reaktivierten Einfüge- bzw. Aktualisierungsauslösern gestartet wird.</p> <p>Dieser Parameter kann keine Nullwerte enthalten, wenn der Parameter operMode auf GSE_AUTOGC_CREATE gesetzt wurde. Er kann Nullwerte enthalten, wenn operMode auf GSE_AUTOGC_RECREATE gesetzt ist.</p>
precisionLevel	INTEGER	<p>Die Genauigkeit, mit der die Quelldaten den entsprechenden Bezugsdaten entsprechen müssen, damit der Geocodierer die Quelldaten erfolgreich verarbeiten kann.</p> <p>Dieser Parameter kann keine Nullwerte enthalten, wenn der Parameter operMode auf GSE_AUTOGC_CREATE gesetzt wurde. Er kann Nullwerte enthalten, wenn operMode auf GSE_AUTOGC_RECREATE gesetzt ist.</p> <p>Kommentar: Die Genauigkeitsstufe kann von 1 bis 100 Prozent reichen.</p>
vendorSpecific	VARCHAR(256)	<p>Vom Hersteller bereitgestellte technische Informationen, beispielsweise der Pfad und Name einer Datei, über die Parameter eingestellt werden.</p> <p>Dieser Parameter kann keine Nullwerte enthalten, wenn der Parameter operMode auf GSE_AUTOGC_CREATE gesetzt wurde. Er kann Nullwerte enthalten, wenn operMode auf GSE_AUTOGC_RECREATE gesetzt ist.</p>

Ausgabeparameter

Tabelle 12. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_enable_autogc.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

Einschränkungen

- Der Parameter layerColumn muss auf eine Spalte verweisen, die als Tabellenschicht registriert wurde.
- Wenn der Parameter operMode auf GSE_AUTOGC_CREATE gesetzt wurde, müssen Sie dem Parameter gcId die Kennung eines registrierten Geocodierers zuordnen.

db2gse.gse_enable_db

Verwenden Sie diese gespeicherte Prozedur, um einer Datenbank die Ressourcen bereitzustellen, die sie zum Speichern räumlicher Operationen und zum Unterstützen von Operationen benötigt. Diese Ressourcen umfassen räumliche Datentypen, einen Typ eines räumlichen Index, Katalogtabellen und -sichten, bereitgestellte Funktionen und andere gespeicherte Prozeduren. Der externe Bibliotheks- und Funktionsname für diese gespeicherte Prozedur lautet `db2gse.gse_enable_db`.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseEnableDB` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der Berechtigungen `SYSADM` oder `DBADM` für die zu aktivierende Datenbank haben.

Ausgabeparameter

Tabelle 13. Ausgabeparameter für die gespeicherte Prozedur `db2gse.gse_enable_db`.

Name	Datentyp	Beschreibung
<code>msgCode</code>	<code>INTEGER</code>	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
<code>msgText</code>	<code>VARCHAR(1024)</code>	Vollständige Fehlermeldung, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_enable_idx

Verwenden Sie diese gespeicherte Prozedur zum Erstellen eines Index für eine räumliche Spalte.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseEnableIdx` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der folgenden Berechtigungen haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, für die der aktivierte Index verwendet werden soll.
- Die Berechtigung CONTROL oder INDEX auf diese Tabelle.

Eingabeparameter

Tabelle 14. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_enable_idx.

Name	Datentyp	Beschreibung
layerSchema	VARCHAR(30)	Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle gehört. Dieser Parameter kann Nullwerte enthalten. Kommentar: Sie müssen einen Wert für diesen Parameter angeben. Der Parameter kann ein Nullwert (NULL) sein.
layerTable	VARCHAR(128)	Name der Tabelle, in der der zu erstellende Index definiert werden soll. Dieser Parameter kann keine Nullwerte enthalten.
layerColumn	VARCHAR(128)	Name der räumlich aktivierten Spalte, die mit Hilfe des zu erstellenden Index durchsucht werden soll. Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 14. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_enable_idx. (Forts.)

Name	Datentyp	Beschreibung
indexName	VARCHAR(128)	<p>Name des zu erstellenden Index.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p> <p>Kommentar: Geben Sie keinen Schemanamen an. Der Spatial Extender ordnet den Index automatisch dem Schema zu, auf das vom Parameter layerSchema verwiesen wird.</p>
gridSize1	DOUBLE	<p>Diese Zahl gibt an, wie die Unterteilung des feinsten Gitters sein soll.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
gridSize2	DOUBLE	<p>Diese Zahl gibt an, (1) dass für diesen Index kein zweites Gitter vorhanden sein soll oder (2) wie die Unterteilung des zweiten Gitters sein soll.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn kein zweites Gitter vorhanden sein soll, geben Sie 0 an. Soll ein zweites Gitter verwendet werden, muss es feiner sein als das mit gridSize1 angegebene Gitter.</p>
gridSize3	DOUBLE	<p>Diese Zahl gibt an, (1) dass für diesen Index kein drittes Gitter vorhanden sein soll oder (2) wie die Unterteilung des dritten Gitters sein soll.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn kein drittes Gitter vorhanden sein soll, geben Sie 0 an. Soll ein drittes Gitter verwendet werden, muss es feiner sein als das mit gridSize2 angegebene Gitter.</p>

Ausgabeparameter

Tabelle 15. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_enable_idx.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_enable_sref

Geben Sie über diese gespeicherte Prozedur an, wie negative und Dezimalzahlen in einem spezifischen Koordinatensystem in positive ganze Zahlen umgewandelt werden sollen, so dass sie vom Spatial Extender gespeichert werden können. Ihre Angaben werden als *räumliches Bezugssystem* bezeichnet. Beim Verarbeiten dieser gespeicherten Prozedur werden der Katalogsicht DB2GSE.SPATIAL_REF_SYS Informationen zum räumlichen Bezugssystem hinzugefügt. Informationen zu dieser Sicht finden Sie im Abschnitt „DB2GSE.SPATIAL_REF_SYS“ auf Seite 145.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion gseEnableSref im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Keine erforderlich.

Eingabeparameter

Tabelle 16. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_enable_sref.

Name	Datentyp	Beschreibung
srId	INTEGER	Eine numerische Kennung für dieses räumliche Bezugssystem. Dieser Parameter kann keine Nullwerte enthalten. Kommentar: Diese Kennung muss innerhalb der räumlich aktivierten Datenbank eindeutig sein.
srName	VARCHAR(64)	Kurzbeschreibung des räumlichen Bezugssystems. Dieser Parameter kann keine Nullwerte enthalten. Kommentar: Diese Beschreibung muss innerhalb der räumlich aktivierten Datenbank eindeutig sein.
falsex	DOUBLE	Eine Zahl, die von einem negativen X-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null). Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 16. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_enable_sref. (Forts.)

Name	Datentyp	Beschreibung
falsey	DOUBLE	<p>Eine Zahl, die von einem negativen Y-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
xyunits	DOUBLE	<p>Eine Zahl, die mit einer dezimalen X-Koordinate oder einer dezimalen Y-Koordinate multipliziert wird, um eine ganze Zahl zu erhalten, die als 32-Bit-Datenelement gespeichert werden kann.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
falsez	DOUBLE	<p>Eine Zahl, die von einem negativen Z-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
zunits	DOUBLE	<p>Eine Zahl, die mit einer dezimalen Z-Koordinate multipliziert wird, um eine ganze Zahl zu erhalten, die als 32-Bit-Datenelement gespeichert werden kann.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
falsem	DOUBLE	<p>Eine Zahl, die von einem negativen Maß abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
munits	DOUBLE	<p>Eine Zahl, die mit einem dezimalen Maß multipliziert wird, um eine ganze Zahl zu erhalten, die als 32-Bit-Datenelement gespeichert werden kann.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>

Tabelle 16. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_enable_sref. (Forts.)

Name	Datentyp	Beschreibung
scId	INTEGER	Numerische Kennung des Koordinatensystems, aus dem das räumliche Bezugssystem abgeleitet wird. Die numerische Kennung des Koordinatensystems können Sie in der Katalogsicht DB2GSE.COORD_REF_SYS im Abschnitt „DB2GSE.COORD_REF_SYS“ auf Seite 143 ermitteln. Dieser Parameter kann keine Nullwerte enthalten.

Ausgabeparameter

Tabelle 17. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_enable_sref.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_export_shape

Verwenden Sie diese gespeicherte Prozedur zum Exportieren einer Schicht und ihrer zugeordneten Tabelle in eine Formdatei oder zum Erstellen einer neuen Formdatei und zum Exportieren einer Schicht und ihrer zugeordneten Tabelle in diese neue Datei.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseExportShape` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss die Berechtigung `SELECT` auf die zu exportierende Tabelle haben.

Eingabeparameter

Tabelle 18. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_export_shape`.

Name	Datentyp	Beschreibung
<code>layerSchema</code>	<code>VARCHAR(30)</code>	Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle gehört. Dieser Parameter kann Nullwerte enthalten. Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_export_shape</code> aufgerufen wurde.
<code>layerTable</code>	<code>VARCHAR(128)</code>	Name der zu exportierenden Tabelle. Dieser Parameter kann keine Nullwerte enthalten.
<code>layerColumn</code>	<code>VARCHAR(30)</code>	Name der Spalte, die als die zu exportierende Schicht registriert wurde. Dieser Parameter kann keine Nullwerte enthalten.
<code>fileName</code>	<code>VARCHAR(128)</code>	Name der Formdatei, in die die angegebene Schicht exportiert werden soll. Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 18. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_export_shape. (Forts.)

Name	Datentyp	Beschreibung
whereClause	VARCHAR(1024)	Der Hauptteil der WHERE-Klausel. Er definiert eine Einschränkung für die Gruppe der zu exportierenden Zeilen. Die Klausel kann auf eine Attributspalte in der zu exportierenden Tabelle verweisen. Das Schlüsselwort WHERE ist in dieser Klausel nicht erforderlich.
Dieser Parameter kann Nullwerte enthalten.		

Ausgabeparameter

Tabelle 19. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_export_shape.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

Einschränkung

Sie können immer nur eine Schicht auf einmal exportieren.

db2gse.gse_import_sde

Verwenden Sie diese gespeicherte Prozedur zum Importieren einer SDE-Übertragungsdatei in eine Datenbank, die für räumliche Operationen aktiviert wurde. Die gespeicherte Prozedur kann auf eine von zwei verschiedenen Arten arbeiten:

- Wenn die SDE-Übertragungsdatei für eine vorhandene Tabelle bestimmt ist, die eine registrierte Schichtspalte enthält, lädt der Spatial Extender die Tabelle mit den Daten der Datei.
- Andernfalls erstellt der Spatial Extender eine Tabelle mit einer räumlichen Spalte, registriert diese Spalte als Schicht und lädt die Schicht sowie die anderen Spalten der Tabelle mit den Daten aus der Datei.

Das in der SDE-Übertragungsdatei angegebene räumliche Bezugssystem wird mit den räumlichen Bezugssystemen verglichen, die für den Spatial Extender registriert wurden. Wenn das angegebene System einem registrierten System entspricht, werden die negativen und Dezimalwerte in der Übertragungsdatei nach dem Laden auf die durch das registrierte System vorgegebene Weise geändert. Entspricht das angegebene System keinem der registrierten Systeme, erstellt der Spatial Extender ein neues räumliches Bezugssystem zur Definition der Änderungsvorschriften.

Autorisierung

Wenn Sie Daten in eine vorhandene Tabelle importieren, muss die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, eine der folgenden Berechtigungen haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, in die die Daten importiert werden sollen.
- Die Berechtigung CONTROL auf diese Tabelle.

Wenn die Tabelle, in die Daten importiert werden sollen, zunächst erstellt werden muss, muss die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, eine der folgenden Berechtigungen haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die zu erstellende Tabelle enthält.

Eingabeparameter

Tabelle 20. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_import_sde`.

Name	Datentyp	Beschreibung
layerSchema	VARCHAR(30)	Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle oder Sicht gehört. Dieser Parameter kann Nullwerte enthalten. Er darf maximal 30 Zeichen umfassen. Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_import_sde</code> aufgerufen wurde.
layerTable	VARCHAR(128)	Name der Tabelle, in die die Daten aus der SDE-Übertragungsdatei geladen werden sollen. Dieser Parameter kann keine Nullwerte enthalten.
layerColumn	VARCHAR(30)	Name der Spalte, die als die Schicht registriert wurde, in die die räumlichen Daten aus der SDE-Übertragungsdatei geladen werden sollen. Dieser Parameter kann keine Nullwerte enthalten. Er darf maximal 30 Zeichen umfassen.
fileName	VARCHAR(128)	Name der zu importierenden SDE-Übertragungsdatei. Dieser Parameter kann keine Nullwerte enthalten.
commitScope	INTEGER	Anzahl der Datensätze pro Prüfpunkt. Dieser Parameter kann Nullwerte enthalten.

Ausgabeparameter

Tabelle 21. Ausgabeparameter für die gespeicherte Prozedur `db2gse.gse_import_sde`.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_import_shape

Verwenden Sie diese gespeicherte Prozedur zum Importieren einer ESRI-Formdatei in eine Datenbank, die für räumliche Operationen aktiviert wurde. Die gespeicherte Prozedur kann auf eine von zwei verschiedenen Arten arbeiten:

- Wenn die Formdatei für eine vorhandene Tabelle bestimmt ist, die eine registrierte Schichtspalte enthält, lädt der Spatial Extender die Tabelle mit den Daten der Datei.
- Andernfalls erstellt der Spatial Extender eine Tabelle mit einer räumlichen Spalte, registriert diese Spalte als Schicht und lädt die Schicht sowie die anderen Spalten der Tabelle mit den Daten aus der Datei.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseImportShape` im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Wenn Sie eine Gruppe von ESRI-Formdarstellungen importieren, erhalten Sie mindestens zwei Dateien. Alle Dateien haben dasselbe Dateinamenpräfix, weisen jedoch unterschiedliche Dateierweiterungen auf. Die Dateierweiterungen der beiden Dateien, die Sie immer erhalten, lauten `.shp` und `.shx`.

Um die Dateien für eine Gruppe von Formdarstellungen zu erhalten, geben Sie in das Feld **File name** im Fenster **Import Spatial Data** den gemeinsamen Namen der Dateien ein. Geben Sie keine Dateierweiterung an. Auf diese Weise stellen Sie sicher, dass alle Dateien, die Sie benötigen – die Datei `.shp`, die Datei `.shx` sowie alle sonstigen eventuell dazugehörigen Dateien – importiert werden.

Gehen Sie beispielsweise davon aus, dass eine Gruppe von ESRI-Formdarstellungen in Dateien mit dem Namen `Lakes.shp` und `Lakes.shx` gespeichert ist. Beim Importieren dieser Darstellungen würden Sie dem Parameter `fileName` nur den Namen `Lakes` zuordnen.

SDE-Transferdateien haben einen Namen, aber keine Dateierweiterung. Daher ordnen Sie dem Parameter `fileName` beim Importieren einer SDE-Transferdatei den Namen, aber keine Dateierweiterung zu.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der folgenden Berechtigungen haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, in die die importierten Formdaten geladen werden sollen.
- Die Berechtigung CONTROL auf diese Tabelle.

Eingabeparameter

Tabelle 22. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_import_shape`.

Name	Datentyp	Beschreibung
layerSchema	VARCHAR(30)	Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle oder Sicht gehört. Dieser Parameter kann Nullwerte enthalten. Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_import_shape</code> aufgerufen wurde.
layerTable	VARCHAR(128)	Name der Tabelle, in die die importierte Formdatei geladen werden sollen. Dieser Parameter kann keine Nullwerte enthalten.
layerColumn	VARCHAR(30)	Name der Spalte, die als die Schicht registriert wurde, in die die Formdaten geladen werden sollen. Dieser Parameter kann keine Nullwerte enthalten.
fileName	VARCHAR(128)	Name der zu importierenden Formdatei. Dieser Parameter kann keine Nullwerte enthalten.
exceptionFile	VARCHAR(128)	Pfad und Name der Datei, in der die Formen gespeichert werden sollen, die nicht importiert werden konnten. Diese ist eine neue Datei, die bei der Ausführung der gespeicherten Prozedur <code>db2gse.gse_import_shape</code> erstellt wird. Ordnen Sie dem Parameter <code>exceptionFile</code> einen Dateinamen, aber keine Dateierweiterung zu. Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 22. Eingabeparameter für die gespeicherte Prozedur *db2gse.gse_import_shape*. (Forts.)

Name	Datentyp	Beschreibung
srId	INTEGER	<p>Kennung des für die Schicht zu verwendenden räumlichen Bezugssystems, in das die Formdaten geladen werden sollen.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn diese Kennung nicht angegeben ist, wird die interne Umwandlung auf die maximal mögliche Auflösung für die Formdatei eingestellt.</p>
commitScope	INTEGER	<p>Anzahl der Datensätze pro Prüfpunkt.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p>

Ausgabeparameter

Tabelle 23. Ausgabeparameter für die gespeicherte Prozedur *db2gse.gse_import_shape*.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlermeldung, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_register_gc

Verwenden Sie diese gespeicherte Prozedur zum Registrieren eines anderen Geocodierers als des Standard-Codierers. Wenn Sie feststellen wollen, ob bereits ein Geocodierer registriert wurde, schlagen Sie in der Katalogsicht DB2GSE.SPATIAL_GEOCODER (siehe Abschnitt „DB2GSE.SPATIAL_GEOCODER“ auf Seite 145) nach.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der Berechtigungen SYSADM oder DBADM für die zu aktivierende Datenbank haben.

Eingabeparameter

Tabelle 24. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_register_gc.

Name	Datentyp	Beschreibung
gcId	INTEGER	Numerische Kennung des zu registrierenden Geocodierers. Dieser Parameter kann keine Nullwerte enthalten. Kommentar: Diese Kennung muss innerhalb der Datenbank eindeutig sein.
gcName	VARCHAR(64)	Kurzbeschreibung des zu registrierenden Geocodierers. Dieser Parameter kann keine Nullwerte enthalten. Kommentar: Diese Beschreibung muss eine innerhalb der Datenbank eindeutige Zeichenfolge sein.
vendorName	VARCHAR(64)	Name des Herstellers, der den zu registrierenden Geocodierer bereitgestellt hat. Dieser Parameter kann keine Nullwerte enthalten.
primaryUDF	VARCHAR(256)	Vollständig qualifizierter Name des zu registrierenden Geocodierers. Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 24. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_register_gc. (Forts.)

Name	Datentyp	Beschreibung
precisionLevel	INTEGER	Die Genauigkeit, mit der die Quelldaten den entsprechenden Bezugsdaten entsprechen müssen, damit der Geocoder die Quelldaten erfolgreich verarbeiten kann. Dieser Parameter kann keine Nullwerte enthalten. Kommentar: Die Genauigkeitsstufe kann von 1 bis 100 Prozent reichen.
vendorSpecific	VARCHAR(256)	Vom Hersteller bereitgestellte technische Informationen, beispielsweise der Pfad und Name einer Datei, über die Parameter eingestellt werden. Dieser Parameter kann Nullwerte enthalten.
geoArea	VARCHAR(256)	Der zu geocodierende geografische Bereich. Dieser Parameter kann Nullwerte enthalten.
Beschreibung	VARCHAR(256)	Anmerkungen des Herstellers. Dieser Parameter kann Nullwerte enthalten.

Ausgabeparameter

Tabelle 25. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_register_gc.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_register_layer

Verwenden Sie diese gespeicherte Prozedur zum Registrieren einer räumlichen Spalte als Schicht. Beim Verarbeiten dieser gespeicherten Prozedur werden der Katalogsicht DB2GSE.GEOMETRY_COLUMNS Informationen zu der zu registrierenden Schicht hinzugefügt. Informationen zu dieser Sicht finden Sie im Abschnitt „DB2GSE.GEOMETRY_COLUMNS“ auf Seite 144.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion gseRegisterLayer im Beispielprogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Diese gespeicherte Prozedur kann nicht für die folgenden Tabellentypen eingesetzt werden:

- A = Aliasname
- H = Hierarchietabelle
- N = Kurzname
- S = Übersichtstabelle
- U = Typisierte Tabelle
- W = Typisierte Sicht

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der folgenden Berechtigungen haben:

- Für eine Tabellenspalte:
 - SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, zu der diese Schicht gehört.
 - Die Berechtigung CONTROL oder ALTER auf diese Tabelle.
- Für eine Sichtstufe:
 - Die Berechtigung SELECT für die Basistabelle oder -tabellen, die (1) die zu geocodierenden Adressdaten für diese Schicht enthalten und (2) die räumlichen Daten enthalten, die aus der Geocodierung resultieren.

Eingabeparameter

Tabelle 26. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_register_layer`.

Name	Datentyp	Beschreibung
layerSchema	INTEGER(30)	<p>Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle oder Sicht gehört.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_register_layer</code> aufgerufen wurde.</p>
layerTable	VARCHAR(128)	<p>Der Name der Tabelle, die die Spalte enthält, die als Schicht registriert werden soll.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
layerColumn	VARCHAR(128)	<p>Name der als Schicht zu registrierenden Spalte. Wenn die Spalte in einer Tabelle nicht existiert, fügt der Spatial Extender sie mit der Anweisung ALTER hinzu. In einer Sicht muss die Spalte bereits existieren.</p> <p>Für den Parameter <code>layerColumn</code> kann nur eine Spalte angegeben werden. Wenn Sie mehrere Spalten einer Tabelle oder Sicht als Schichten registrieren, müssen Sie diese gespeicherte Prozedur daher für jede Spalte separat ausführen.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
layerTypeName	VARCHAR(64)	<p>Datentyp der als Schicht zu registrierenden Spalte. Nur vom Spatial Extender bereitgestellte Datentypen werden akzeptiert. Sie müssen den Datentyp in Großbuchstaben eingeben, z. B.: <code>ST_POINT</code></p> <p>Sie müssen keinen Schemennamen angeben, da er automatisch hinzugefügt wird.</p> <p>Dieser Parameter kann keine Nullwerte enthalten, wenn die Spalte eine Tabellenspalte ist, die bei der Verarbeitung dieser gespeicherten Prozedur erstellt werden soll. Andernfalls, sofern die Spalte eine vorhandene Spalte in einer Tabelle oder Sicht ist, kann dieser Parameter Nullwerte enthalten.</p>

Tabelle 26. Eingabeparameter für die gespeicherte Prozedur *db2gse.gse_register_layer*. (Forts.)

Name	Datentyp	Beschreibung
srId	INTEGER	<p>Kennung des für diese Schicht verwendeten räumlichen Bezugssystems.</p> <p>Dieser Parameter kann für eine Tabellenschicht keine Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtschicht.</p>
geoSchema	VARCHAR(30)	<p>Wird beim Registrieren einer Sichtspalte als Schicht verwendet. Der Parameter geoSchema ist das Schema der Tabelle, das der Sicht zugrunde liegt, zu der die Spalte gehört.</p> <p>Dieser Parameter kann beim Registrieren einer Sichtspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Tabellenspalte als Schicht.</p> <p>Sichten, die auf mehreren Basistabellen oder anderen Sichten basieren, werden von diesem Parameter nicht unterstützt.</p> <p>Kommentar: Wenn Sie keinen Wert für den Parameter geoSchema angeben, wird als Standardwert der Wert des Parameters layerSchema verwendet.</p>
geoTable	VARCHAR(128)	<p>Wird beim Registrieren einer Sichtspalte als Schicht verwendet. Der Parameter geoTable ist der Name der Tabelle, die der Sicht zugrunde liegt, zu der die Spalte gehört.</p> <p>Sichten, die auf mehreren Basistabellen oder anderen Sichten basieren, werden von diesem Parameter nicht unterstützt.</p> <p>Dieser Parameter kann beim Registrieren einer Sichtspalte als Schicht keine Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Tabellenspalte als Schicht.</p>

Tabelle 26. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_register_layer. (Forts.)

Name	Datentyp	Beschreibung
geoColumn	VARCHAR(128)	<p>Wird beim Registrieren einer Sichtspalte als Schicht verwendet. Der Parameter geoColumn ist der Name der Tabellenspalte, die dieser Sichtspalte zugrunde liegt.</p> <p>Sichten, die auf mehreren Basistabellen oder anderen Sichten basieren, werden von diesem Parameter nicht unterstützt.</p> <p>Dieser Parameter kann beim Registrieren einer Sichtspalte als Schicht keine Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Tabellenspalte als Schicht.</p>
nAttributes	SMALLINT	<p>Anzahl der Spalten, die für diese Schicht zu geocodierenden Quelldaten enthalten.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p>
attr1Name	VARCHAR(128)	<p>Name der ersten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Wenn Sie den Standard-Geocodierer verwenden wollen, müssen Sie die Straßennamen in der Spalte attr1Name speichern.</p>
attr2Name	VARCHAR(128)	<p>Name der zweiten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Wenn Sie den Standard-Geocodierer verwenden wollen, müssen Sie die Ortsnamen in der Spalte attr2Name speichern.</p>

Tabelle 26. Eingabeparameter für die gespeicherte Prozedur *db2gse.gse_register_layer*. (Forts.)

Name	Datentyp	Beschreibung
attr3Name	VARCHAR(128)	<p>Name der dritten Spalte, die zu geocodierende Quellendaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Wenn Sie den Standard-Geocodierer verwenden wollen, müssen Sie die Namen bzw. Abkürzungen der Bundesstaaten in der Spalte attr3Name speichern.</p>
attr4Name	VARCHAR(128)	<p>Name der vierten Spalte, die zu geocodierende Quellendaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Wenn Sie den Standard-Geocodierer verwenden wollen, müssen Sie die Postleitzahl in der Spalte attr4Name speichern.</p>
attr5Name	VARCHAR(128)	<p>Name der fünften Spalte, die zu geocodierende Quellendaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr5Name.</p>
attr6Name	VARCHAR(128)	<p>Name der sechsten Spalte, die zu geocodierende Quellendaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr6Name.</p>

Tabelle 26. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_register_layer. (Forts.)

Name	Datentyp	Beschreibung
attr7Name	VARCHAR(128)	<p>Name der siebten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr7Name.</p>
attr8Name	VARCHAR(128)	<p>Name der achten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr8Name.</p>
attr9Name	VARCHAR(128)	<p>Name der neunten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr9Name.</p>
attr10Name	VARCHAR(128)	<p>Name der zehnten Spalte, die zu geocodierende Quelldaten für diese Schicht enthält.</p> <p>Dieser Parameter kann beim Registrieren einer Tabellenspalte als Schicht Nullwerte enthalten. Der Spatial Extender ignoriert diesen Parameter beim Registrieren einer Sichtspalte als Schicht.</p> <p>Der Standard-Geocodierer ignoriert die Spalte Attr10Name.</p>

Ausgabeparameter

Tabelle 27. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_register_layer.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

Einschränkungen

- Wenn Sie eine Sichtspalte als Schicht registrieren, muss sie auf einer Tabellenspalte basieren, die bereits als Schicht registriert wurde.
- Die zu geocodierenden Daten für die zu registrierende Schicht können nicht auf mehr als zehn Attributspalten verteilt sein.

db2gse.gse_run_gc

Verwenden Sie diese gespeicherte Prozedur zum Ausführen eines Geocodierers im Stapelbetrieb. Informationen zu dieser Task finden Sie im Abschnitt „Den Geocodierer im Stapelbetrieb ausführen“ auf Seite 57.

Ein Beispiel für den Code zum Aufrufen dieser gespeicherten Prozedur finden Sie in der C-Funktion `gseRunGC` im Beispielpogramm. Informationen zu diesem Programm finden Sie in „Kapitel 8. Anwendungen für Spatial Extender schreiben“ auf Seite 73.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der folgenden Berechtigungen haben:

- SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Tabelle enthält, mit der der angegebene Geocodierer arbeiten soll.
- Die Berechtigung CONTROL oder UPDATE auf diese Tabelle.

Eingabeparameter

Tabelle 28. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_run_gc.

Name	Datentyp	Beschreibung
layerSchema	VARCHAR(30)	Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle oder Sicht gehört. Dieser Parameter kann Nullwerte enthalten. Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_run_gc</code> aufgerufen wurde.
layerTable	VARCHAR(128)	Der Name der Tabelle, die die Spalte enthält, in die die geocodierten Daten eingefügt werden sollen. Dieser Parameter kann keine Nullwerte enthalten.
layerColumn	VARCHAR(128)	Der Name der Spalte, in die die geocodierten Daten eingefügt werden sollen. Dieser Parameter kann keine Nullwerte enthalten.

Tabelle 28. Eingabeparameter für die gespeicherte Prozedur *db2gse.gse_run_gc*. (Forts.)

Name	Datentyp	Beschreibung
gcId	INTEGER	<p>Kennung des auszuführenden Geocodierers.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Zum Ermitteln der Kennungen der registrierten Geocodierer schlagen Sie in der Katalogsicht DB2GSE.SPATIAL_GEOCODER nach.</p>
precisionLevel	INTEGER	<p>Die Genauigkeit, mit der die Quelldaten den entsprechenden Bezugsdaten entsprechen müssen, damit der Geocodierer die Quelldaten erfolgreich verarbeiten kann.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Die Genauigkeitsstufe kann von 1 bis 100 Prozent reichen.</p>
vendorSpecific	VARCHAR(256)	<p>Vom Hersteller bereitgestellte technische Informationen, beispielsweise der Pfad und Name einer Datei, über die Parameter eingestellt werden.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p>
whereClause	VARCHAR(256)	<p>Der Hauptteil der WHERE-Klausel. Er definiert eine Einschränkung zu der Gruppe der zu geocodierenden Datensätze. Die Klausel kann auf eine Attributspalte in der Tabelle, mit der der Geocodierer arbeiten soll, verweisen.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p>
commitScope	INTEGER	<p>Anzahl der Datensätze pro Prüfpunkt.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p>

Ausgabeparameter

Tabelle 29. Ausgabeparameter für die gespeicherte Prozedur *db2gse.gse_run_gc*.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_unregist_gc

Verwenden Sie diese gespeicherte Prozedur, um die Registrierung eines anderen Geocodierers als des Standard-Codierers aufzuheben.

Wenn Sie Informationen über den Geocodierer suchen, dessen Registrierung aufgehoben werden soll, schlagen in der Katalogsicht DB2GSE.SPATIAL_GEOCODER nach; siehe hierzu den Abschnitt „DB2GSE.SPATIAL_GEOCODER“ auf Seite 145.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der Berechtigungen SYSADM oder DBADM für die Datenbank haben, die den Geocodierer enthält, dessen Berechtigung aufgehoben werden soll.

Eingabeparameter

Tabelle 30. Eingabeparameter für die gespeicherte Prozedur db2gse.gse_unregist_gc.

Name	Datentyp	Beschreibung
gcId	INTEGER	Die Kennung des Geocodierers, dessen Registrierung aufgehoben werden soll. Dieser Parameter kann keine Nullwerte enthalten.

Ausgabeparameter

Tabelle 31. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_unregist_gc.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlermeldung, wie auf dem Spatial Extender-Server konstruiert.

db2gse.gse_unregist_layer

Verwenden Sie diese gespeicherte Prozedur, um die Registrierung einer Schicht aufzuheben. Die gespeicherte Prozedur führt hierzu folgende Aktionen aus:

- Entfernen der Definition der Schicht aus den Spatial Extender-Katalogtabellen.
- Löschen der Prüfeinschränkung, die der Spatial Extender auf die Basistabelle dieser Schicht platziert hat, um sicherzustellen, dass die räumlichen Daten der Schicht den Anforderungen des räumlichen Bezugssystems der Schicht entsprechen.
- Freigeben der Auslöser, mit denen die räumliche Spalte beim Hinzufügen, Ändern oder Entfernen von Adressdaten aktualisiert wird.

Wenn Adressdaten in einer Tabellenzeile geocodiert werden, werden die resultierenden räumlichen Daten in dieselbe Zeile gestellt. Wenn die Zeile gelöscht wird, werden daher gleichzeitig auch die Adressdaten und die räumlichen Daten gelöscht. Auslöser führen nicht zum Löschen der räumlichen Daten.

Beim Verarbeiten dieser gespeicherten Prozedur werden Informationen über die Schicht aus der Katalogsicht DB2GSE.GEOMETRY_COLUMNS entfernt. Informationen zu dieser Sicht finden Sie im Abschnitt „DB2GSE.GEOMETRY_COLUMNS“ auf Seite 144.

Autorisierung

Die Benutzer-ID, unter der diese gespeicherte Prozedur aufgerufen wird, muss eine der folgenden Berechtigungen haben:

- Für eine Tabellenspalte:
 - SYSADM- oder DBADM-Berechtigung für die Datenbank, die die Basistabelle zu dieser Schicht enthält.
 - Die Berechtigung CONTROL oder ALTER auf diese Tabelle.
- Für eine Sichtstufe:
 - Die Berechtigung SELECT für die Basistabelle oder -tabellen, die (1) die zu geocodierenden Adressdaten für diese Schicht enthalten und (2) die räumlichen Daten enthalten, die aus der Geocodierung resultieren.

Eingabeparameter

Tabelle 32. Eingabeparameter für die gespeicherte Prozedur `db2gse.gse_unregister_layer`.

Name	Datentyp	Beschreibung
layerSchema	VARCHAR(30)	<p>Name des Schemas, zu dem die im Parameter <code>layerTable</code> angegebene Tabelle gehört.</p> <p>Dieser Parameter kann Nullwerte enthalten.</p> <p>Kommentar: Wenn Sie keinen Wert für den Parameter <code>layerSchema</code> angeben, wird als Standardwert die Benutzer-ID verwendet, unter der die gespeicherte Prozedur <code>db2gse.gse_unregister_layer</code> aufgerufen wurde.</p> <p>In Großbuchstaben müssen Sie alle Schemennamen, Tabellennamen, Sichtnamen, Spaltennamen oder Schichtnamen angeben, die Sie einem Parameter zuordnen.</p>
layerTable	VARCHAR(128)	<p>Der Name der Tabelle, die die im Parameter <code>layerColumn</code> angegebene Spalte enthält.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p>
layerColumn	VARCHAR(128)	<p>Name der räumlichen Spalte, die als die Schicht definiert wurde, deren Registrierung aufgehoben werden soll.</p> <p>Dieser Parameter kann keine Nullwerte enthalten.</p> <p>Kommentar: Für den Parameter <code>layerColumn</code> kann nur eine Schicht angegeben werden. Wenn Sie die Registrierung mehrerer Schichten einer Tabelle oder Sicht rückgängig machen, müssen Sie diese gespeicherte Prozedur daher für jede Schicht separat ausführen.</p>

Ausgabeparameter

Tabelle 33. Ausgabeparameter für die gespeicherte Prozedur db2gse.gse_unregist_layer.

Name	Datentyp	Beschreibung
msgCode	INTEGER	Code, der den Nachrichten zugeordnet ist, die der Aufrufer dieser gespeicherten Prozedur zurückgeben kann.
msgText	VARCHAR(1024)	Vollständige Fehlernachricht, wie auf dem Spatial Extender-Server konstruiert.

Einschränkung

Wenn eine als Sichtstufe definierte Sichtspalte auf einer Tabellenspalte basiert, die als Tabellenschicht definiert wurde, können Sie die Registrierung dieser Tabellenschicht nicht aufheben, bis die Registrierung der Sichtschicht aufgehoben wurde.

Kapitel 10. Nachrichten

DB2 Spatial Extender generiert Nachrichten, die

- von der Steuerzentrale
- von gespeicherten Prozeduren
- von räumlichen Funktionen

zurückgegeben werden.

Jede Nachricht weist eine aus Präfix und Nachrichtennummer bestehende Nachrichten-ID auf. Die Nachrichtennummer wird auch als *SQLCODE* bezeichnet.

Drei Arten von Nachrichten werden unterschieden: Fehlermeldungen, Warnungen und Informationsnachrichten. Nachrichten-IDs, die auf *E* enden, bezeichnen Fehlermeldungen. IDs, die auf *W* enden, bezeichnen Fehlermeldungen, und solche, die auf *I* enden, Informationsnachrichten.

Von der Steuerzentrale zurückgegebene Nachrichten

Die folgenden Nachrichten werden von der Steuerzentrale zurückgegeben. Ihr *SQLCODE* beginnt mit den Buchstaben "DBA".

DBA7200E More than 10 columns are selected as input to a geocoder

Erläuterung: Es können bis zu 10 Spalten als Eingabe für einen Geocoder ausgewählt werden.

Benutzeraktion: Verschieben Sie Spaltennamen aus dem Feld "Selected columns" in das Feld "Available columns", bis das Feld "Selected columns" maximal zehn Namen enthält.

DBA7201E The database is not enabled for Spatial Extender operation.

Erläuterung: Die Datenbank muss für Spatial Extender aktiviert werden, bevor Sie mit Spatial Extender arbeiten können.

Benutzeraktion: Klicken Sie mit der rechten Maustaste die Datenbank an, und wählen Sie Spatial Extender → Enable in den Menüs aus.

Von gespeicherten Prozeduren zurückgegebene Nachrichten

Die folgenden Nachrichten werden von gespeicherten Prozeduren zurückgegeben. Ihr *SQLCODE* beginnt mit den Buchstaben "GSE", gefolgt von Zahlen im Bereich 0000 bis 2035.

Achtung: Zwei Fehlermeldungen – GSE2022E und GSE2035E – gehören nicht zum Nachrichtenverzeichnis von DB2 Spatial Extender. Wenn die Fehler auftreten, auf die sich diese Nachrichten beziehen, werden Sie daher in der folgenden Nachricht online darüber informiert, dass sie nicht abgerufen werden können: SQL10007N Message "<SQLCODE -2022 or -2035>" could not be retrieved. Reason code: "4".

GSE0000I The operation is completed successfully.

GSE0001E Spatial Extender could not perform the requested operation ("<operation-name>") under user ID "<benutzer-ID>".

Erläuterung: Sie haben diese Operation mit einer Benutzer-ID angefordert, die keine ausreichende Berechtigung für diese Aktion hat.

Benutzeraktion: Schlagen Sie in der Dokumentation nach, um festzustellen, welche Berechtigung erforderlich ist, und fordern Sie diese Berechtigung bei Ihrem Spatial Extender Administrator an.

GSE0002E "<wert>" is not a valid value for the "<argumentname>" argument.

Erläuterung: Der eingegebene Wert war falsch oder falsch geschrieben.

Benutzeraktion: Schlagen Sie in der Dokumentation nach oder fragen Sie Ihren Spatial Extender Administrator, welcher Wert bzw. welcher Wertebereich angegeben werden muss.

GSE0003E Spatial Extender could not perform the requested operation because argument "<argumentname>" was not specified.

Erläuterung: Sie haben ein für diese Operation erforderliches Argument nicht angegeben.

Benutzeraktion: Geben Sie das Argument "<argument-name>" mit dem gewünschten Wert an, und fordern Sie die Operation erneut an.

GSE0004W The argument "<argumentname>" was not evaluated.

Erläuterung: Die angeforderte Operation verwendet das Argument "<argumentname>" nicht.

Benutzeraktion: Keine erforderlich.

GSE0005E Spatial Extender could not process your request to create an object named "<objektname>".

Erläuterung: Das Objekt "<objektname>" ist bereits vorhanden, oder Sie haben nicht die erforderliche Berechtigung, um es zu erstellen. Das Objekt kann eine Tabelle, eine Spalte, ein Auslöser, ein Index oder ein anderes Objekt sein.

Benutzeraktion: Wenn "<objektname>" das gewünschte Objekt ist, brauchen Sie keine Aktion auszuführen. Geben Sie andernfalls den Namen richtig ein, und stellen Sie sicher, dass Sie die Berechtigung zum Erstellen des Objekts haben.

GSE0006E Spatial Extender could not perform the requested operation on an enabled or registered object named "<objektname>".

Erläuterung: Das Objekt "<objektname>" wurde bereits aktiviert bzw. registriert, oder es ist bereits vorhanden. Das Objekt kann eine Schicht, ein Index, ein räumliches Bezugssystem, ein Koordinatensystem oder ein anderes Objekt sein.

Benutzeraktion: Stellen Sie sicher, dass das Objekt "<objektname>" vorhanden ist, und geben Sie Ihre Anforderung erneut ein.

GSE0007E Spatial Extender could not perform the requested operation on "<objektname>", an object that has not yet been enabled or registered.

Erläuterung: Das Objekt "<objektname>" wurde nicht aktiviert oder registriert. Das Objekt kann eine Schicht, ein Index, ein räumliches Bezugssystem, ein räumliches Koordinatensystem, ein Geocodierer oder ein anderes Objekt sein.

Benutzeraktion: Aktivieren oder registrieren Sie das Objekt "<objektname>". Übergeben Sie Ihre Anforderung anschließend erneut.

GSE0008E An unexpected SQL error ("<sql-fehlernachricht>") has occurred.

Benutzeraktion: Schlagen Sie die ausführliche Nachricht zum SQLCODE in der SQL-Fehlernachricht "<sql-fehlernachricht>" nach. Wenden Sie sich ggf. an Ihren IBM Ansprechpartner.

GSE0009E The requested operation could not be performed on an object named "<objektname>" that already exists.

Erläuterung: "<objektname>" ist in der Datenbank bzw. dem Betriebssystem bereits vorhanden. Das Objekt kann eine Datei, eine Tabelle, eine Spalte, ein Index, ein Auslöser oder ein anderes Objekt sein.

Benutzeraktion: Stellen Sie sicher, dass Sie das Objekt bei dem Versuch, darauf zuzugreifen, richtig angeben. Wenn nötig, löschen Sie das Objekt.

GSE0010E The requested operation could not be performed on an object named "<objektname>" that might not exist.

Erläuterung: "<objektname>" ist in der Datenbank oder im Betriebssystem nicht vorhanden. Das Objekt kann eine Datei, eine Tabelle, eine Spalte, ein Index, ein Auslöser oder ein anderes Objekt sein.

Benutzeraktion: Stellen Sie sicher, dass Sie eine ausreichende Berechtigung zum Zugreifen auf das Objekt haben. Wenn Sie diese Berechtigung haben und das Objekt nicht vorhanden ist, müssen Sie es erstellen.

GSE0011E Spatial Extender could not disable or unregister object "<objektname>".

Erläuterung: "<objektname>" hängt von einem anderen Objekt ab. "<objektname>" kann ein räumliches Bezugssystem, eine Schicht, ein Geocodierer oder ein anderes Objekt sein.

Benutzeraktion: Schlagen Sie in der Dokumentation nach, um festzustellen, von welcher Art von Objekt "<objektname>" abhängen kann. Entfernen Sie anschließend das spezifische Objekt, von dem "<objektname>" abhängt.

GSE0012E Spatial Extender could not process your request because the fully qualified spatial column "<schichtschema.schichtname.schichtspalte>" is not registered as a table layer.

Erläuterung: Die vollständig qualifizierte räumliche Spalte "<schichtschema.schichtname.schichtspalte>" muss als Tabellenschicht registriert werden, bevor Sie bestimmte ihr zugeordnete Operationen ausführen können (z. B. das Aktivieren ihres Index, das Aktivieren eines Geocodierers zum Ausfüllen im Stapelbetrieb oder zum automatischen Aktualisieren).

Benutzeraktion: Stellen Sie sicher, dass die vollständig qualifizierte räumliche Spalte "<schichtschema.schichtname.schichtspalte>" als Tabellenschicht registriert ist. Überprüfen Sie hierzu die Sicht DB2GSE.GEOMETRY_COLUMNS im Spatial Extender-Katalog. Stellen Sie außerdem sicher, dass die Tabelle, die diese Spalte enthält, entsprechende gültige Attributspalten enthält.

GSE0013E The database is not enabled for spatial operations.

Erläuterung: Die Datenbank wurde nicht für räumliche Operationen aktiviert. Der Spatial

Extender-Katalog ist deshalb nicht vorhanden.

Benutzeraktion: Aktivieren Sie die Datenbank für räumliche Operationen.

GSE0014E The database has already been enabled for spatial operations.

Erläuterung: Die Datenbank wurde bereits für räumliche Operationen aktiviert.

Benutzeraktion: Stellen Sie sicher, dass die Datenbank wie erwartet aktiviert wurde. Inaktivieren Sie die Datenbank gegebenenfalls.

GSE0498E The following error occurred:
"<fehlernachricht>".

GSE0499W Spatial Extender issued the following warning: "<warnung>".

GSE0500E The operation mode that you specified ("<betriebsmodus>") is invalid.

Erläuterung: Der angegebene Modus wird von der angeforderten Operation nicht unterstützt.

Benutzeraktion: Schlagen Sie in der Dokumentation nach, um festzustellen, welche Modi von der Operation unterstützt werden.

GSE1001E Spatial Extender was unable to register a view layer that is named "<schemaname.sichtname.spaltenname>" and that is based on spatial column "<schemaname.tabellenname.spaltenname>".

Erläuterung: Die angegebene räumliche Spalte ("<schemaname.tabellenname.spaltenname>") wurde nicht als Tabellenschicht registriert.

Benutzeraktion: Registrieren Sie die Spalte "<schemaname.tabellenname.spaltenname>" als Tabellenschicht.

GSE1002E Spatial Extender was unable to register a view layer that is named "<schemaname.sichtname.spaltenname>" and that is based on table "<schemaname.tabellenname>".

Erläuterung: Die Tabelle, die Sie in ("<schemaname.tabellenname>") angegeben haben, liegt der Sicht "<schemaname.sichtname.spaltenname>" weder direkt noch indirekt zugrunde.

Benutzeraktion: Stellen Sie fest, welches die Basistabelle für die Sicht "<schemaname.sichtname.spaltenname>" ist, und geben Sie diese Tabelle an.

GSE1003E Spatial Extender was unable to access a column named "<spaltenname>" in a table or view named "<schemaname.objektname>".

Erläuterung: Die Tabelle oder Sicht "<schemaname.objektname>" enthält keine Spalte mit dem Namen "<spaltenname>".

Benutzeraktion: Überprüfen Sie Definition der Tabelle oder Sicht "<schemaname.objektname>", um den richtigen Namen der gewünschten Spalte festzustellen.

GSE1004E Spatial Extender was unable to register the fully qualified spatial column "<schemaname.tabellenname.spaltenname>" as a table layer.

Erläuterung: Die Spalte "<schemaname.tabellenname.spaltenname>" hat keinen Typ räumlicher Daten oder ist nicht einer Basistabelle zugeordnet.

Benutzeraktion: Definieren Sie einen Typ räumlicher Daten für die Spalte "<schemaname.tabellenname.spaltenname>", oder stellen Sie sicher, dass diese Spalte Teil einer lokalen Basistabelle ist.

GSE1005E The spatial reference system ("`<sichtschicht-räumliche-bezugs-ID>`") that you specified for a view layer differs from the spatial reference system ("`<tabellenschicht-räumliche-bezugs-ID>`") that is used for this layer's underlying table layer.

Erläuterung: Ein räumliches Bezugssystem einer Sichtschicht muss dasselbe sein wie das der Tabelle zugrundeliegende räumliche Bezugssystem.

Benutzeraktion: Geben Sie das der Tabellenschicht zugrundeliegende räumliche Bezugssystem für die Sichtschicht an.

GSE1006E Because "`<räumliche-bezugs-ID>`" is an invalid spatial reference system ID, Spatial Extender was unable to register the layer that you requested.

Erläuterung: Das angegebene räumliche Bezugssystem ("`<räumliche-bezugs-ID>`") wurde nicht aktiviert oder registriert.

Benutzeraktion: Aktivieren oder registrieren Sie das räumliche Bezugssystem. Übergeben Sie anschließend Ihre Anforderung zur Registrierung der Schicht erneut.

GSE1007E An SQL error (SQLSTATE "`<sql-status>`") might have occurred when Spatial Extender tried unsuccessfully to add a spatial column ("`<spaltenname>`") to table "`<schemaname.tabellename>`".

Benutzeraktion: Schlagen Sie die Nachricht zu SQLSTATE "`<sqlstatus>`" nach.

GSE1008E Spatial Extender was unable to register a view layer "`<schicht-schema.schichtname.schichtspalte>`" because the spatial data type "`<schichtspaltentyp>`" of the view layer does not match the spatial data type "`<geo-spaltentyp>`" of the underlying table layer "`<geo-schema.geo-name.geo-spalte>`".

Erläuterung: Der Typ von räumlichen Daten einer Sichtspalte "`<schichtschema.schichtname.schichtspalte>`" muss dem Typ räumlicher Daten der Tabellenschicht "`<geo-schema.geo-name.geo-spalte>`" entsprechen, die der Schicht zugrundeliegt. Die Inkonsistenz zwischen diesen beiden Datentypen bewirkt eine Doppeldeutigkeit bei der Verarbeitung räumlicher Daten.

Benutzeraktion: Stellen Sie sicher, dass die Typen von räumlichen Daten der Sichtschicht und der ihr zugrundeliegende Tabellenschicht identisch sind.

GSE1020E "`<'räumliche-bezugs-ID>`" is an invalid spatial reference system ID.

Erläuterung: Ein räumliches Bezugssystem mit der Kennung "`<'räumliche-bezugs-ID>`" wurde nicht aktiviert.

Benutzeraktion: Stellen Sie sicher, dass der angegebene räumliche Bezug aktiviert wurde.

GSE1021E Spatial Extender could not enable spatial reference system "`<räumliche-bezugs-ID>`" because the corresponding spatial coordinate system ID "`<räumliche-koordinaten-ID>`" is invalid.

Erläuterung: Ein Koordinatensystem mit der Kennung "`<räumliche-koordinaten-ID>`" ist im Spatial Extender-Katalog nicht definiert.

Benutzeraktion: Überprüfen Sie die Kennung des Koordinatensystems "`<räumliche-koordinaten-ID>`". Überprüfen Sie hierzu die Sicht DB2GSE.COORD_REF_SYS im Spatial Extender-Katalog.

GSE1030E Because "`<schemaname.tabellenname>`" is not a base table, Spatial Extender could not enable a geocoder for it.

Erläuterung: Das Objekt, das die zu geocodierenden Quellendaten enthält, muss eine Basistabelle sein.

Benutzeraktion: Stellen Sie sicher, dass die Spalten mit den zu geocodierenden Daten Teil einer Basistabelle sind.

GSE1031E Spatial Extender could not enable geocoder "`<geocodierer-ID>`" to operate automatically in create mode for layer "`<schichtschema.schichtname.schichtspalte>`".

Erläuterung: Mögliche Erklärungen sind:

- Der Geocodierer ist bereits aktiviert für eine automatische Aktualisierung der Schicht "`<schichtschema.schichtname.schichtspalte>`".
- Der Geocodierer wurde für diese Schicht vorübergehend ungültig gemacht.
- Es wurden keine Spalten für Quellendaten für diese Schicht definiert.

Benutzeraktion: Wenn der Geocodierer vorübergehend ungültig gemacht wurde, aktivieren Sie ihn, so dass er automatisch im Modus "Recreate" arbeitet.

GSE1032E Spatial Extender could not enable geocoder "`<geocodierer-ID>`" to operate automatically in recreate mode for layer "`<schichtschema.schichtname.schichtspalte>`".

Erläuterung: Mögliche Erklärungen sind:

- Der Geocodierer ist bereits aktiviert für eine automatische Aktualisierung der Schicht "`<schichtschema.schichtname.schichtspalte>`".
- Der Geocodierer wurde für diese Schicht nicht zuvor ungültig gemacht.
- Es wurden keine Spalten für Quellendaten für diese Schicht definiert.

Benutzeraktion: Wenn der Geocodierer zuvor im Freigabemodus inaktiviert wurde oder über-

haupt nicht für diese Schicht definiert wurde, aktivieren Sie ihn, so dass er automatisch im Modus "Create" arbeitet.

GSE1033E An SQL error occurred when Spatial Extender tried to add triggers to a table that contains the column for layer "`<schichtschema.schichtname.schichtspalte>`" (SQLSTATE "`<sqlstatus>`").

Erläuterung: Der Zweck der Auslöser ist das Aufrechterhalten der Datenintegrität zwischen den Attributspalten, aus denen die Eingabe des Geocodierers stammt, und der räumlichen Spalte, in die seine Ausgabe geht. Der SQL-Fehler trat auf, als DB2 versuchte, diese Auslöser zu erstellen.

Benutzeraktion: Schlagen Sie die Nachricht zu SQLSTATE "`<sqlstatus>`" nach.

GSE1034E Spatial Extender could not disable geocoder "`<geocodierer-ID>`" in drop mode for layer "`<schichtschema.schichtname.schichtspalte>`".

Erläuterung: Mögliche Erklärungen sind:

- Der Geocodierer wurde nicht aktiviert für eine automatische Aktualisierung der Schicht "`<schichtschema.schichtname.schichtspalte>`".
- Der Geocodierer wurde im Freigabemodus inaktiviert.

Benutzeraktion: Ermitteln Sie den Status des Geocodierers, bevor Sie versuchen, ihn zu inaktivieren. Wurde er registriert? War er aktiviert? Entscheiden Sie anschließend, ob er im Freigabemodus inaktiviert werden muss. Wenn er beispielsweise überhaupt nie aktiviert wurde, muss er auch nicht inaktiviert werden.

GSE1035E Spatial Extender could not disable geocoder "`<geocodierer-ID>`" in invalidate mode for layer "`<schichtschema.schichtname.schichtspalte>`".

Erläuterung: Mögliche Erklärungen sind:

- Der Geocodierer wurde nicht aktiviert für eine automatische Aktualisierung der Schicht "`<schichtschema.schichtname.schichtspalte>`".
- Der Geocodierer wurde im ungültigen Modus oder im Freigabemodus inaktiviert.

Benutzeraktion: Ermitteln Sie den Status des Geocodierers, bevor Sie versuchen, ihn zu inaktivieren. Wurde er registriert? War er aktiviert? Entscheiden Sie anschließend, ob er im ungültigen Modus inaktiviert werden muss. Wenn er beispielsweise bereits im ungültigen Modus inaktiviert wurde, muss er in diesem Modus nicht ein zweites Mal inaktiviert werden.

GSE1036E An SQL error occurred when Spatial Extender tried to drop triggers from a table that contains the column for layer "`<schichtschema.schichtname.schichtspalte>`" (SQLSTATE "`<sqlstatus>`").

Erläuterung: Die Auslöser wurden erstellt, um die Datenintegrität zwischen den Attributspalten, aus denen die Eingabe des Geocodierers stammt, und der räumlichen Spalte, in die seine Ausgabe geht, aufrechtzuerhalten. Der SQL-Fehler trat auf, als DB2 versuchte, diese Auslöser freizugeben.

Benutzeraktion: Schlagen Sie die Nachricht zu SQLSTATE "`<sqlstatus>`" nach.

GSE1037E Spatial Extender could not geocode source data for table layer "`<schichtschema.schichtname.schichtspalte>`", possibly because an incorrect value "`<anzahl-der-attribut>`" was assigned to the argument that specifies how many attribute columns are to provide source data for this layer.

Erläuterung: Die Anzahl der Attributspalten zu dieser Schicht wurde falsch angegeben, oder der Name einer oder mehrerer dieser Spalten wurde falsch angegeben.

Benutzeraktion: Stellen Sie sicher, dass diese Schicht mit der richtigen Anzahl und den richtigen Namen der zugeordneten Attributspalten registriert wurde, oder überprüfen Sie die Rich-

tigkeit der Ein- und Ausgabedaten für den Geocodierer.

GSE1038E An SQL error occurred when Spatial Extender tried to geocode source data for table layer "`<schichtschema.schichtname.schichtspalte>`" in batch mode (SQLSTATE "`<sqlstatus>`").

Benutzeraktion:

- Schlagen Sie die Nachricht zu SQLSTATE "`<sqlstatus>`" nach.
- Stellen Sie sicher, dass der Inhalt und das Argument `primaryUDF` dieser Schicht richtig definiert wurden.

GSE1050E The grid size that you specified ("`<gittergröße>`") is invalid for the first grid level.

Erläuterung: Sie haben Null oder eine negative Zahl als Gittergröße für die erste Gitterstufe angegeben.

Benutzeraktion: Geben Sie eine positive Zahl für die Gittergröße an.

GSE1051E The grid size that you specified ("`<gittergröße>`") is invalid for the second and third grid levels.

Erläuterung: Sie haben eine negative Zahl als Gittergröße für die zweite Gitterstufe angegeben.

Benutzeraktion: Geben Sie Null oder eine positive Zahl für die Gittergröße an.

GSE1052E An SQL error occurred when the Spatial Extender tried to create spatial index "`<indexschema.indexspalte>`" for a table layer "`<schichtschema.schichtname.schichtspalte>`" (SQLSTATE "`<sqlstatus>`").

Benutzeraktion:

- Stellen Sie sicher, dass der räumliche Index richtig angegeben wurde und dass der räumlichen Spalte kein Index zugeordnet ist.

- Schlagen Sie die SQLSTATE zugeordnete Nachricht “<sqlstatus>” nach.

GSE1500I Source record “<datensatznummer>” was successfully geocoded.

Erläuterung: Ein Datensatz mit attributiven Daten wurde erfolgreich geocodiert.

GSE1501W Source record “<datensatznummer>” was not geocoded.

Erläuterung: Die Genauigkeitsstufe war zu hoch.

Benutzeraktion: Führen Sie eine Geocodierung mit einer niedrigeren Genauigkeitsstufe durch.

GSE1502W Source record “<datensatznummer>” was not found.

Benutzeraktion: Stellen Sie fest, ob der Datensatz in der Datenbank vorhanden ist.

GSE2001E Spatial Extender could not perform the operation that you requested.

Benutzeraktion: Wenden Sie sich an Ihren Datenbankadministrator.

GSE2002E A database management system error has occurred.

Benutzeraktion: Wenden Sie sich an Ihren Datenbankadministrator.

GSE2003E The stored procedure that you invoked cannot connect to your workstation.

Erläuterung: Die gespeicherte Prozedur kann nicht auf die Informationen zugreifen, die Ihre Workstation identifizieren.

Benutzeraktion: Wenden Sie sich an Ihren Datenbankadministrator.

GSE2004E Spatial Extender cannot validate the coordinate system identifier specified in the SDE transfer file that you are importing.

Benutzeraktion: Versuchen Sie eine oder mehrere der folgenden Methoden:

- Stellen Sie sicher, dass die Kennung des räumlichen Bezugssystems in der SDE-Transferdatei auf die Kennung des richtigen Koordinatensystems zeigt.
- Stellen Sie fest, ob die Kennung des richtigen Koordinatensystems in der Katalogsicht DB2GSE.COORD_REF_SYS aufgeführt ist. Informieren Sie Ihren Datenbankadministrator, wenn dies nicht der Fall ist.
- Stellen Sie fest, ob die SDE-Transferdatei beschädigt ist. Wenn dies der Fall ist, versuchen Sie, eine intakte Kopie dieser Datei zu beschaffen und zu importieren.

GSE2005E Spatial Extender cannot validate the file that you want to export.

Erläuterung: Für diesen Fehler kann es einen oder mehrere Gründe geben. So sind Sie beispielsweise u. U. nicht berechtigt, auf die Datei zuzugreifen. Eventuell kann auch der Spatial Extender die Datei nicht finden oder lesen oder er kann die Datentypen von Daten in dieser Datei nicht erkennen.

Benutzeraktion: Achten Sie darauf, dass Sie den vollständig qualifizierten Pfad der Datei angeben. Achten Sie weiter darauf, dass die Benutzer-ID, unter der Sie die gespeicherte Prozedur db2gse.gse_export_shape ausführen, für jedes Verzeichnis des Pfads lese- und schreibberechtigt ist. Stellen Sie sicher, dass die Platte, die diese Verzeichnisse enthält, an denselben Knoten angehängt ist, auf dem DB2 läuft, und dass die Platte den im Pfad angegebenen Mount-Punkt verwendet. Überprüfen Sie außerdem, ob der Spatial Extender die Datentypen der Daten in der Datei erkennt.

Wenn der Fehler erneut auftritt, versuchen Sie festzustellen, ob die Datei beschädigt ist. Wenn dies der Fall ist, versuchen Sie, eine intakte

Kopie dieser Datei zu beschaffen und zu exportieren.

GSE2006E An I/O error for a file named "`<dateiname>`" has occurred.

Benutzeraktion: Stellen Sie sicher, dass die Datei vorhanden ist, dass Sie eine ausreichende Zugriffsberechtigung auf die Datei haben und dass die Datei nicht von einem anderen Benutzer verwendet wird.

GSE2007E Spatial Extender cannot validate the layer to which you want to import data.

Erläuterung: Möglicherweise wurde der Name der Spalte, für die diese Schicht definiert ist, falsch angegeben, oder er entspricht nicht den Standardnamenskonventionen. Eventuell wurde auch der Name der Tabelle, zu der die Spalte gehört, falsch angegeben, oder er entspricht nicht den Standardnamenskonventionen.

Benutzeraktion: Stellen Sie sicher, dass die Schicht in der Katalogsicht DB2GSE.GEOMETRY_COLUMNS aufgeführt ist, dass der Name der Spalte und der Name der Tabelle, zu der die Spalte gehört, korrekt angegeben sind und dass diese Namen den Standardnamenskonventionen entsprechen.

GSE2008E Spatial Extender attempted to insert a null into a layer that has a NOT NULL constraint.

Benutzeraktion: Importieren Sie entweder die Spalte, die Nullen enthält, in eine Schicht, die Nullen akzeptiert, oder bitten Sie Ihren Datenbankadministrator, die NOT NULL-Integritätsbedingung zu entfernen.

GSE2012E Spatial Extender was unable to access the layer that you want to import data to.

Erläuterung: Die Benutzer-ID, unter der Sie auf die Schicht zugreifen wollen, ist nicht zur Änderung der Spalte berechtigt, für die diese Schicht definiert ist.

Benutzeraktion: Bitten Sie Ihren Datenbankadministrator, Ihnen die erforderliche Berechtigung zu erteilen (z. B. benötigen Sie u. U. ein INSERT- oder SELECT-Zugriffsrecht für die Tabelle, zu der die Spalte gehört).

GSE2014E Spatial Extender was unable to import data to, or export data from, the specified layer.

Erläuterung: Der Spatial Extender konnte die Schicht nicht finden, in die bzw. aus der Sie Daten importieren oder exportieren wollen.

Benutzeraktion: Stellen Sie fest, ob die Schicht in der Sicht DB2GSE.GEOMETRY_COLUMNS aufgeführt wird. Wenn dies nicht der Fall ist, registrieren Sie die Schicht mit der gespeicherten Prozedur `db2gse.gse_register_layer` oder im Fenster **Create Layer** der Steuerzentrale. Wenn die Schicht in DB2GSE.GEOMETRY_COLUMNS aufgeführt ist, informieren Sie Ihren Datenbankadministrator über das Problem.

GSE2016E Spatial Extender was unable to import the shape file that you requested to the layer that you specified.

Erläuterung: Der Datentyp der räumlichen Daten, die Sie importieren wollen, ist nicht mit dem Datentyp der Schicht kompatibel, für die diese räumlichen Daten bestimmt sind.

Benutzeraktion: Erstellen Sie eine neue Schicht, deren Datentyp mit dem Datentyp der räumlichen Daten kompatibel ist, die Sie importieren wollen. Importieren Sie anschließend die Daten in diese neue Schicht. Importieren Sie alternativ eine andere Formdatei, deren räumliche Daten mit der Schicht, die Sie auffüllen wollen, kompatibel sind.

GSE2021E Spatial Extender was unable to access the shape file that you want to import.

Erläuterung: Für diesen Fehler gibt es mehrere mögliche Gründe. Z. B. kennt der Spatial Extender möglicherweise nicht den vollständigen Pfad der Formdatei, oder er erkennt das Dateiformat

nicht, oder die Platte, auf der sich die Datei befindet, ist nicht ordnungsgemäß angehängt.

Benutzeraktion: Achten Sie darauf, dass Sie den vollständigen Pfad der Datei angeben. Wenn der Fehler dennoch erneut auftritt, prüfen Sie, ob die Datei tatsächlich eine Formdatei ist und nicht eine andere Art von Datei, die versehentlich als Formdatei angegeben wurde. Wenn die Datei eine Formdatei ist, versuchen Sie eine der folgenden Lösungsmöglichkeiten:

- Stellen Sie fest, ob die Datei beschädigt ist. Wenn dies der Fall ist, versuchen Sie, eine intakte Kopie der Datei zu beschaffen und zu importieren.
- Wenn Sie von einer anderen Workstation aus auf die Datei zugreifen, stellen Sie sicher, dass folgende Bedingungen erfüllt sind:
 - Die Platte, auf der sich die Datei befindet, ist angehängt.
 - Die Platte verwendet den im Dateipfadnamen angegebenen Mount-Punkt.
 - Die Benutzer-ID, die Sie an der anderen Workstation verwenden, ist für die die Datei leseberechtigt.

GSE2022E The specified spatial reference system identifier does not exist.

Erläuterung: Die Kennung des räumlichen Bezugssystems (SRID), die Sie für die Formdatei angegeben haben, die Sie importieren wollen, ist nicht im Katalog des Spatial Extender aufgeführt.

Benutzeraktion: Führen Sie eine der folgenden Aktionen aus:

- Ordnen Sie die Formdatei einem kompatiblen räumlichen Bezugssystem zu, dessen SRID bereits in der Katalogsicht DB2GSE.SPATIAL_REF_SYS verzeichnet ist.
- Erstellen Sie alternativ ein neues räumliches Bezugssystem für die Formdatei. Prüfen Sie dann, ob die SRID für dieses System in den Katalog des Spatial Extender eingetragen wurde.

Diese Nachricht gehört nicht zum Nachrichten-katalog von DB2 Spatial Extender. Wenn die Fehler auftreten, auf die sich diese Nachricht bezieht,

werden Sie online darüber informiert, dass sie nicht abgerufen werden kann.

GSE2023E Spatial Extender was unable to import attribute data from the shape file that you specified.

Erläuterung: Die Definition einer Attributspalte in der Formdatei konnte nicht in eine Definition für eine entsprechende Spalte in der Tabelle umgewandelt werden, in die Sie Daten importieren wollen.

Benutzeraktion: Stellen Sie sicher, dass der Datentyp, die maximale Länge sowie andere Merkmale dieser Attributspalte in Äquivalente oder Entsprechungen für die Attributspalte umgewandelt werden können, in die Sie Ihre Daten importieren.

GSE2026E Spatial Extender was unable to create a file to contain the data that it could not import.

Erläuterung: Wenn Sie eine Formdatei importieren, sammelt der Spatial Extender alle Datensätze in dieser Datei, deren Import fehlschlägt, damit sie zu einem späteren Zeitpunkt korrigiert und importiert werden können. In diesem Fall verfügte der Spatial Extender nicht über genügend Informationen oder eine entsprechende Berechtigung, um eine Datei zur Aufnahme der zurückgewiesenen Datensätze zu erstellen.

Benutzeraktion: Geben Sie einen vollständig qualifizierten Pfad für die Datei an, die der Spatial Extender für die zurückgewiesenen Datensätze erstellt. Stellen Sie sicher, dass nicht bereits eine Datei mit demselben Pfad und Namen existiert. Achten Sie weiter darauf, dass die Benutzer-ID, unter der Sie die gespeicherte Prozedur db2gse.gse_import_shape ausführen, für jedes Verzeichnis des Pfads lese- und schreibberechtigt ist. Stellen Sie sicher, dass die Platte, die diese Verzeichnisse enthält, an denselben Knoten angehängt ist, auf dem DB2 läuft, und dass die Platte den im Pfad angegebenen Mount-Punkt verwendet.

GSE2027E Spatial Extender could not perform the import or export operation that you requested.

Erläuterung: Der verfügbare Speicherplatz reicht nicht aus, um diese Operation durchzuführen. Die Datei, die Sie importieren oder exportieren, ist möglicherweise beschädigt und benötigt daher zu viel Speicherplatz.

Benutzeraktion: Versuchen Sie, die Datei erneut zu importieren oder exportieren. Wenn der Fehler erneut auftritt, versuchen Sie festzustellen, ob die Datei beschädigt ist. Wenn dies der Fall ist, versuchen Sie, eine intakte Kopie der Datei zu beschaffen, und importieren oder exportieren Sie diese Kopie. Wenn das Problem nicht behoben werden kann, benachrichtigen Sie Ihren Datenbankadministrator.

GSE2030E Spatial Extender was unable to import data to the column that you specified.

Erläuterung: Die angegebene Spalte wurde nicht als Schicht registriert.

Benutzeraktion: Wenn Sie SDE-Daten importieren wollen, registrieren Sie die Spalte über die DB2-Steuerzentrale oder mit der gespeicherten Prozedur `db2gse.gse_import_sde` als Schicht und importieren Sie die Daten. Wenn Sie Formdaten importieren wollen, registrieren Sie die Spalte über die DB2-Steuerzentrale oder mit der gespeicherten Prozedur `db2gse.gse_import_shape` als Schicht und importieren Sie die Daten.

GSE2031E Spatial Extender was unable to import data to the layer that you specified.

Erläuterung: Die Tabelle, für die die Schicht definiert wurde, existiert nicht mehr.

Benutzeraktion: Wenn Sie SDE-Daten importieren wollen, erstellen Sie die Tabelle über die DB2-Steuerzentrale oder mit der gespeicherten Prozedur `db2gse.gse_import_sde` erneut und importieren Sie die Daten. Wenn Sie Formdaten importieren wollen, erstellen Sie die Tabelle über die DB2-Steuerzentrale oder mit der gespeicher-

ten Prozedur `db2gse.gse_import_shape` erneut und importieren Sie die Daten.

GSE2032E Spatial Extender was unable to import or export attribute data.

Erläuterung: Eine mögliche Erklärung für diesen Fehler ist die, dass Sie versuchen, Attributdaten in eine Tabelle zu importieren, eine oder mehrere der in der Importdatei angegebenen Attributspalten jedoch keine Entsprechungen in dieser Tabelle haben. Eine andere Erklärung ist die, dass Sie versuchen, Daten aus einer Tabelle oder Sicht zu exportieren, eine oder mehrere der Attributspalten in dieser Tabelle oder Sicht jedoch keine Entsprechungen in der Datei haben, in die Sie die Daten exportieren wollen.

Benutzeraktion: Wenn Sie Attributdaten importieren, geben Sie die Spalte (oder jede der Spalten) in der Importdatei an, die keine Entsprechung in der Tabelle hat, in die die Datei geladen wird. Geben Sie dann die Tabelle mit der fehlenden Spalte (bzw. den fehlenden Spalten) an. Alternativ können Sie das Ziel des Importvorgangs in eine andere als die ursprünglich beabsichtigte Schicht und Gruppe von Attributspalten ändern.

Wenn Sie Attributdaten aus einer Tabelle oder Sicht exportieren, geben Sie jede Spalte der Tabelle oder Sicht an, die keine Entsprechung in der Exportdatei hat. Geben Sie dann entweder die Exportdatei mit der fehlenden Spalte (bzw. den fehlenden Spalten) oder eine neue Exportdatei an – eine Datei, die eine Spalte für jede Datenspalte enthält, die Sie exportieren wollen.

GSE2033E Spatial Extender could not read the complete file that you want to import.

Erläuterung: Die Datei ist eventuell beschädigt oder abgeschnitten.

Benutzeraktion: Versuchen Sie, die Datei erneut zu importieren. Wenn der Fehler erneut auftritt, versuchen Sie, eine intakte Kopie der Datei zu beschaffen und zu importieren.

GSE2034E Spatial Extender could not import the SDE transfer file that you requested.

Erläuterung: Der Datentyp der räumlichen Daten, die Sie importieren wollen, ist nicht mit dem Datentyp der Schicht kompatibel, für die diese räumlichen Daten bestimmt sind.

Benutzeraktion: Erstellen Sie eine neue Schicht, deren Datentyp mit dem Datentyp der räumlichen Daten kompatibel ist, die Sie importieren wollen. Importieren Sie anschließend die Daten in diese neue Schicht. Importieren Sie alternativ eine andere SDE-Transferdatei, deren räumliche Daten mit der Schicht, die Sie auffüllen wollen, kompatibel sind.

GSE2035E The specified coordinate is out of bounds.

Erläuterung: Der Spatial Extender hat eine Koordinate gefunden, die entweder zu groß oder zu klein ist und daher nicht in die Grenzen des Koordinatensystems passt, das der angegebenen Schicht zugeordnet ist. Dieser Fehler kann auftreten,

wenn dieses Koordinatensystem mit den Schichtdaten oder mit dem zugeordneten räumlichen Bezugssystem nicht kompatibel ist. Alternativ kann der Fehler auch auf eine beschädigte Form- oder SDE-Transferdatei zurückzuführen sein oder durch anomale Daten hervorgerufen werden, die versehentlich in die Schicht eingefügt wurden.

Benutzeraktion: Prüfen Sie, ob Sie die richtige Kennung für das räumliche Bezugssystem angegeben haben, das die angegebene Schicht verwendet. Wenn dies der Fall ist, ist das System u. U. mit Schichtdaten oder mit dem zugrundeliegenden Koordinatensystem nicht kompatibel. Wählen oder erstellen Sie in Anbetracht dieser Möglichkeiten ein anderes räumliches Bezugssystem für die Schicht. Wenn der Fehler erneut auftritt, benachrichtigen Sie Ihren Datenbankadministrator.

Diese Nachricht gehört nicht zum Nachrichten-katalog von DB2 Spatial Extender. Wenn die Fehler auftreten, auf die sich diese Nachricht bezieht, werden Sie online darüber informiert, dass sie nicht abgerufen werden kann.

Von räumlichen Funktionen zurückgegebene Nachrichten

Der SQLCODE der Nachrichten, die von räumlichen Funktionen zurückgegeben werden, beginnt mit den Buchstaben "GSE", gefolgt von Zahlen im Bereich 3001 bis 3042.

Wenn eine räumliche Funktion eine Nachricht zurückgibt, gibt sie auch den zugehörigen SQLSTATE-Wert zurück, nicht jedoch den SQLCODE.

GSE3001E Unknown system failure.

Erläuterung: Ein unerwarteter Systemfehler ist aufgetreten.

Benutzeraktion: Korrigieren Sie die Syntax, und rufen Sie die Funktion dann erneut auf. Wenn der Fehler erneut auftritt, wenden Sie sich an den technischen Dienst.

Benutzeraktion: Korrigieren Sie die Textfolge, und rufen Sie die Funktion erneut auf. Informationen zur Ermittlung des richtigen Formats für bekannte Textfolgen finden Sie im Handbuch DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3002E Invalid Well-Known Text string

Erläuterung: Eine ungültige bekannte Textfolge (WKT-Folge) wurde als Eingabe für die Funktion verwendet, die Sie aufgerufen haben.

GSE3003E Invalid SRID

Erläuterung: Die Kennung des räumlichen Bezugssystems (SRID), die Sie an diese Funktion übergeben wollten, ist im Systemkatalog von DB2 Spatial Extender nicht aufgeführt.

Benutzeraktion: Geben Sie entweder eine SRID

an, die in der Katalogsicht DB2GSE.SPATIAL_REF_SYS aufgeführt ist, oder erstellen Sie ein räumliches Bezugssystem mit der von Ihnen gewünschten SRID.

GSE3004E Insufficient Memory

Erläuterung: Es war nicht genügend Speicherplatz verfügbar. DB2 Spatial Extender benötigt bis zu einem Megabyte Speicherplatz.

Benutzeraktion: Führen Sie eine Neuordnung des Speichers durch, um DB2 Spatial Extender mehr Speicherkapazität zur Verfügung zu stellen. Wenn eine Neuordnung des Speichers nicht möglich ist, fügen Sie mehr physischen Speicher hinzu.

GSE3005E Geometries' SRIDs differ.

Erläuterung: Geometrien, die an eine Funktion von DB2 Spatial Extender übergeben werden, müssen dieselbe Kennung des räumlichen Bezugssystems (SRID) aufweisen.

Benutzeraktion: Erstellen Sie eine der Geometrien erneut, so dass ihr räumliches Bezugssystem mit dem der anderen übereinstimmt.

GSE3006E Invalid binary string.

Erläuterung: Eine nicht ordnungsgemäß konstruierte bekannte Binärzeichenfolge oder ESRI-Binärzeichenfolge wurde als Eingabe für die Funktion verwendet, die Sie aufgerufen haben.

Benutzeraktion: Konstruieren Sie die Zeichenfolge erneut im richtigen Format. Informationen zur Ermittlung des richtigen Formats finden Sie im Handbuch DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3007E Valid geometry not specified.

Erläuterung: Es wurde kein gültiger Geometrietyp an die Funktion übermittelt, die Sie aufgerufen haben. Gültige Typen sind Geometrie, Punkt, Linienfolge, Polygon, Mehrpunktangabe, Mehrlinienfolge und Multipolygon.

Benutzeraktion: Wiederholen Sie die SQL-Anweisung mit einem gültigen Geometrietyp.

GSE3008E Parenthesis not balanced.

Erläuterung: Die Anzahl der linken runden Klammern in der Zeichenfolge der bekannten Textdarstellung stimmt nicht mit der Anzahl der rechten runden Klammern überein.

Benutzeraktion: Geben Sie die Zeichenfolge erneut ein, und geben Sie dabei für jede linke runde Klammer eine entsprechende rechte runde Klammer an.

GSE3009E Too many parts specified.

Erläuterung: Die Anzahl der in der Binärzeichen- oder Textfolge angegebenen Teile ist größer als die tatsächliche Anzahl der bereitgestellten Teile.

Benutzeraktion: Geben Sie die Zeichenfolge unter Angabe der richtigen Teileanzahl erneut ein.

GSE3010E Incorrect geometry type.

Erläuterung: Es wurde ein ungültiger Geometrietyp an die Funktion übermittelt, die Sie aufgerufen haben. So wurde möglicherweise eine Linienfolge an eine Funktion übermittelt, die nur Polygone als Eingabe akzeptiert.

Benutzeraktion: Übermitteln Sie entweder einen Geometrietyp an die Funktion, der von der Funktion verarbeitet werden kann, oder verwenden Sie eine Funktion, die den Geometrietyp akzeptiert, den Sie übermitteln wollen.

GSE3011E Text string is too long.

Erläuterung: Die Länge der Geometrietextfolge übersteigt die maximal zulässige Länge von 4000 Zeichen.

Benutzeraktion: Die Geometrie enthält zu viele Details und kann daher nicht in eine Textfolge umgewandelt werden. Sie können die Geometrie jedoch stattdessen in das WKB-Format oder in das binäre ESRI-Formformat umwandeln.

GSE3012E Invalid parameter value.

Erläuterung: Ein ungültiger Parameter wurde an die Funktion übermittelt.

Benutzeraktion: Vergleichen Sie die Syntax der Funktion mit der Syntax, die im DB2 Spatial Extender Benutzer- und Referenzhandbuch angegeben ist. Korrigieren Sie den ungültigen Parameter, und übergeben Sie die Funktion dann erneut.

GSE3013E Invalid grid size.

Erläuterung: Es wurde eine der folgenden ungültigen Angaben gemacht:

- Eine negative Zahl wurde als Gittergröße für die erste, zweite oder dritte Gitterstufe angegeben.
- Null wurde als Gittergröße für die erste Gitterstufe angegeben.
- Für die zweite Gitterstufe wurde eine geringere Gittergröße angegeben als die der ersten Gitterstufe.
- Für die dritte Gitterstufe wurde eine geringere Gittergröße angegeben als die der zweiten Gitterstufe.

Benutzeraktion: Geben Sie im Fenster **Create Index** oder mit der gespeicherten Funktion `db2gse.gse_enable_idx` eine gültige Gittergröße an. Informationen über gültige Gittergrößen finden Sie im DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3014E Grid size too small.

Erläuterung: Die angegebene Gittergröße führt zu mehr als 1000 Gitterzellen pro Geometrie.

Benutzeraktion: Verwenden Sie das Fenster **Create Index** oder die gespeicherte Funktion `db2gse.gse_enable_idx`, um die Gittergröße zu erhöhen oder eine weitere Gitterstufe hinzuzufügen.

GSE3015E Invalid geometry produced.

Erläuterung: Die eingegebenen Parameter haben eine ungültige Geometrie erzeugt. Eine ungültige

Geometrie wird z. B. von den Parametern erzeugt, die mit der Funktion `LineFromShape` eingegeben werden. Eine Geometrie ist dann ungültig, wenn sie ein Geometriemerkmal verletzt.

Benutzeraktion: Korrigieren Sie den Parameter, und übergeben Sie die Geometrie dann erneut.

GSE3016E Wrong geometries submitted.

Erläuterung: Die Funktion erwartete zwei Geometrien eines bestimmten Typs, die sie jedoch nicht empfing. Die Funktion `ST_Union` z. B. erwartet zwei Geometrien mit derselben Dimension und empfing einen Punkt und eine Linienfolge, die eine unterschiedliche Dimension aufweisen.

Benutzeraktion: Geben Sie Geometrien an, die die Funktion als gültige Eingaben akzeptiert. Informationen zur Ermittlung der gültigen Geometrietyden für diese Funktion finden Sie im Handbuch DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3017E Geometry integrity error.

Erläuterung: Die Funktion kann die übermittelte Geometrie nicht verarbeiten, da ein oder mehrere Merkmale der Geometrie eine Integritätsbedingung verletzen.

Benutzeraktion: Übergeben Sie die Geometrie mit korrekt definierten Merkmalen erneut. Informationen über die Merkmale von Geometrien finden Sie im DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3018E Too many points.

Erläuterung: Die Konstruktion einer Geometrie überstieg die Speichergrenze von 1 MB; die Geometrie hat zu viele Punkte.

Benutzeraktion: Entfernen Sie unnötige Punkte. Im Hinblick auf Leistung und Speicherkapazität sollten Sie nur die Punkte aufnehmen, die zur Darstellung einer Geometrie erforderlich sind. Auf alle nicht unbedingt erforderlichen Punkte sollte verzichtet werden.

GSE3019E Geometry is too small.

Erläuterung: Die von der Funktion `ST_Difference`, `ST_Intersection`, `ST_SymmetricDiff` oder `ST_Union` zurückgegebene Geometrie ist zu klein, so dass sie nicht mit Werten des aktuellen Koordinatensystems dargestellt werden kann.

Benutzeraktion: Wenn ein Ergebnis benötigt wird, erhöhen Sie mit der gespeicherten Funktion `db2gse.gse_enable_sref` den Parameter `xyunits` des räumlichen Bezugssystems der Quellengeometrie. Erstellen Sie dann erneut die Tabelle, in der die Quellengeometrie gespeichert wird.

GSE3020E Buffer out of bounds.

Erläuterung: Die Funktion `buffer` hat einen Puffer außerhalb des Koordinatensystems erstellt.

Benutzeraktion: Reduzieren Sie entweder den Pufferabstand oder ändern Sie das Koordinatensystem der Quellengeometrie. In den meisten Fällen erfordert eine Änderung des Koordinatensystems, dass das räumliche System erneut geladen wird.

GSE3021E Invalid scale factor.

Erläuterung: Ein Maßstabsfaktor (eine XY-Einheit, eine Z-Einheit oder eine M-Einheit) kann nicht kleiner als 1 sein.

Benutzeraktion: Korrigieren Sie mit der gespeicherten Prozedur `db2gse.gse_enable_sref` alle Maßstabsfaktoren in der Katalogsicht `DB2GSE.SPATIAL_REF_SYS`, die kleiner als 1 sind.

GSE3022E Coordinate out of bounds.

Erläuterung: Eine Koordinate ist zu groß oder zu klein, und passt daher nicht in die Grenzen des Koordinatensystems.

Benutzeraktion: Stellen Sie fest, ob die Koordinate korrekt ist. Wenn dies der Fall ist, stellen Sie fest, ob sie in die Grenzen des Koordinatensystems passt, das Sie verwenden. Informationen über dieses Koordinatensystem finden Sie in der Katalogsicht `DB2GSE.COORD_REF_SYS`.

GSE3023E Invalid coordinate system ID.

Erläuterung: Der Spatial Extender kann die angegebene Koordinatensystemkennung nicht auswerten.

Benutzeraktion: Stellen Sie fest, ob die Kennung in der Katalogsicht `DB2GSE.COORD_REF_SYS` aufgeführt ist. Wenn dies nicht der Fall ist, prüfen Sie, ob die Kennung korrekt ist, und bitten Sie Ihren Datenbankadministrator, die Kennung in den Systemkatalog des Spatial Extender einzutragen.

GSE3024E Invalid annotation text.

Erläuterung: Der Anmerkungs-text, der das angegebene Koordinatensystem definiert, kann nicht in eine gültige Projektion umgewandelt werden.

Benutzeraktion: Suchen Sie den Anmerkungs-text für dieses Koordinatensystem in der Katalogsicht `DB2GSE.COORD_REF_SYS`. Stellen Sie fest, ob der Text das System angemessen definiert. Hilfreiche Informationen finden Sie im Kapitel über Koordinatensysteme im DB2 Spatial Extender Benutzer- und Referenzhandbuch.

GSE3025E Projection Error

Erläuterung: Bei dem Versuch, eine Geometrie zu projizieren, ist ein Fehler aufgetreten.

Benutzeraktion: Stellen Sie sicher, dass sich die Geometrie im zulässigen Projektionsbereich befindet.

GSE3026E Polygon rings overlap.

Erläuterung: Die Ringe eines Polygons dürfen sich nicht überlappen, können sich jedoch in einem Tangentialpunkt schneiden.

Benutzeraktion: Korrigieren Sie die Koordinaten des Polygons und übergeben Sie es dann erneut.

GSE3027E Too few points.

Erläuterung: Linienfolgen müssen aus mindestens zwei Punkten, Polygone aus mindestens vier Punkten bestehen.

Benutzeraktion: Übergeben Sie die Geometrie mit der richtigen Punktanzahl erneut.

GSE3028E Polygon is not closed.

Erläuterung: Die Koordinaten des Start- und Endpunkts des Polygons stimmen nicht überein.

Benutzeraktion: Editieren Sie die Koordinatenliste des Polygons und stellen Sie sicher, dass Start- und Endpunkt übereinstimmen. Übergeben Sie die Liste dann erneut.

GSE3029E Exterior ring is invalid.

Erläuterung: Der äußere Ring umschließt den inneren Ring nicht. Der gesamte innere Ring befindet sich außerhalb des äußeren Rings und es gibt keine Überlappung.

Benutzeraktion: Stellen Sie sicher, dass alle Koordinaten des inneren Rings innerhalb des äußeren Rings liegen. Wenn der innere Ring eigentlich den äußeren Ring eines anderen Polygons darstellt, geben Sie die Geometrie als Multipolygon ein.

GSE3030E Polygon has no area.

Erläuterung: Eine Geometrie ist nur dann ein Polygon, wenn ihre Koordinaten zwei Dimensionen im Raum abdecken.

Benutzeraktion: Editieren Sie die Koordinaten des Polygons so, dass sie eine Fläche umschließen, und übergeben Sie das Polygon dann erneut. Oder übergeben Sie ggf. eine Linienfolge.

GSE3031E Polygon contains a spike.

Erläuterung: Nur End- und Startpunkt eines Polygons dürfen übereinstimmen. Alle sonstigen Koordinaten eines Polygonrings müssen sich voneinander unterscheiden und gemeinsam eine Fläche umschließen.

Benutzeraktion: Suchen Sie die Koordinatenpaare mit identischen X- und Y-Werten. Editieren Sie diese Punkte so, dass das Polygon eine einzige Fläche umschließt, und übergeben Sie das Polygon dann erneut.

GSE3032E Exterior rings overlap.

Erläuterung: Die äußeren Ringe eines Multipolygons können sich in einem Tangentialpunkt schneiden. Sie dürfen sich jedoch nicht überlappen.

Benutzeraktion: Editieren Sie die Koordinaten der äußeren Ringe so, dass sie sich nicht überlappen, und übergeben Sie das Multipolygon dann erneut.

GSE3033E Polygon intersects itself.

Erläuterung: Der Polygonring darf sich nicht selbst schneiden.

Benutzeraktion: Editieren Sie die Koordinaten des Rings, der sich selbst schneidet, und übergeben Sie das Polygon dann erneut.

GSE3034E Invalid number of measures.

Erläuterung: Der Parameter *number of measures* der Binärzeichenfolge enthält eine Anzahl von Maßen, die sich von der Anzahl der Maße unterscheidet, die in der Zeichenfolge angegeben ist.

Benutzeraktion: Editieren Sie den Parameter *number of measures* so, dass er mit der Anzahl in der Binärzeichenfolge übereinstimmt.

GSE3035E Invalid number of parts.

Erläuterung: Der Parameter *number of parts* der Binärzeichenfolge gab eine Anzahl von Teilen an, die sich von der Anzahl der Teile unterscheidet, die in der Zeichenfolge angegeben wurde.

Benutzeraktion: Editieren Sie den Parameter *number of parts* so, dass er mit der Anzahl in der Binärzeichenfolge übereinstimmt.

GSE3036E Invalid part offset.

Erläuterung: Der Parameter *part offset* der Binärzeichenfolge gab einen Teileabstand an, der sich von dem in der Zeichenfolge angegebenen Abstand unterscheidet.

Benutzeraktion: Editieren Sie den Parameter *part offset* so, dass er mit dem in der Binärzeichenfolge angegebenen Teileabstand übereinstimmt.

GSE3037E Projection error.

Erläuterung: Es wurde eine unzulässige Geometrie gefunden. Der Teileseparator ist ungültig.

Benutzeraktion: Wenden Sie sich an Ihren IBM Ansprechpartner.

GSE3038E BLOB too small.

Erläuterung: Die Byteanzahl des angegebenen großen Binärobjekts (BLOB) ist kleiner als die Byteanzahl im bereitgestellten BLOB.

Benutzeraktion: Gleichen Sie die BLOB-Länge und die Byteanzahl des BLOB ab, und übergeben Sie die Funktion dann erneut.

GSE3039E Invalid entity type.

Erläuterung: Es wurde eine unzulässige Geometrie gefunden. Die zugehörige Entitätsart ist ungültig.

Benutzeraktion: Wenden Sie sich an Ihren IBM Ansprechpartner.

GSE3040E Invalid byte order

Erläuterung: Die Byteanordnung muss 0 oder 1 sein.

Benutzeraktion: Editieren Sie die Byteanordnung so, dass sie 0 für Little Endian oder 1 für Big Endian ist.

GSE3041E Invalid part.

Erläuterung: Ein Funktionsparameter hat ein Teil indexiert, das nicht existiert. Dieser Fehler

würde z. B. auftreten, wenn der Funktion ST_GeometryN eine 3 zur Rückgabe eines dritten Punktes in einer Multipunktangabe übergeben würde, obwohl die Mehrpunktangabe nur zwei Punkte umfasst.

Benutzeraktion: Editieren Sie den Parameter, und übergeben Sie die Funktion dann erneut.

GSE3042E Empty geometry.

Erläuterung: Der Funktion ST_AsBinary wurde eine leere Geometrie übergeben, obwohl eine solche Eingabe für diese Funktion nicht zulässig ist.

Benutzeraktion: Editieren Sie die übergebene SQL-Anweisung so, dass nur eine nicht leere Geometrie an die Funktion ST_AsBinary übergeben wird. Sie können z. B. eine WHERE-Klausel verwenden, um leere Geometrien mit der Funktion ST_IsEmpty zu disqualifizieren.

Wenn eine räumliche Funktion eine Nachricht zurückgibt, zeigt DB2 die Kurzform dieser Nachricht und den entsprechenden SQLSTATE-Wert in der Nachricht SQL0443N an.

Zum Beispiel:

```
DB21034E The command was processed as an SQL statement because it was not a
valid Command Line Processor command. During SQL processing it returned:
SQL0443N Routine "DB2GSE.ST_POINTFROMTEX" (specific name
"SQL000503150228187") has returned an error SQLSTATE with diagnostic text
"Invalid SRID". SQLSTATE=38601
```

In Tabelle 34 können Sie den SQLCODE ermitteln, der dem in Nachricht SQL0443N zurückgegebenen SQLSTATE-Wert zugeordnet ist. Den vollständigen Text zu dem SQLCODE finden Sie in diesem Kapitel. Alternativ können Sie den folgenden Befehl eingeben:

```
DB2 ? [SQLCODE]
```

Tabelle 34. SQLSTATE-Werte und SQLCODE-Werte der Nachrichten, die von räumlichen Funktionen zurückgegeben werden

Bei diesem SQLSTATE-Wert:	. . . ist der SQLCODE:
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

Tabelle 34. SQLSTATE-Werte und SQLCODE-Werte der Nachrichten, die von räumlichen Funktionen zurückgegeben werden (Forts.)

Bei diesem SQLSTATE-Wert:	. . . ist der SQLCODE:
38808	GSE3023E
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

Kapitel 11. Katalogsichten

Die Katalogsichten des Spatial Extender enthalten Metadaten zu folgenden Informationen:

- Koordinatensysteme, die verwendet werden können. Informationen über Kennungen und Anmerkungstexten zu diesen Systemen finden Sie im Abschnitt „DB2GSE.COORD_REF_SYS“.
- Räumliche Spalten, die als Schichten registriert wurden. Informationen zu den Namen dieser Spalten, den Datentypen und den zugeordneten räumlichen Bezugssystemen finden Sie im Abschnitt „DB2GSE.GEOMETRY_COLUMNS“ auf Seite 144.
- Geocodierer, die Sie verwenden können. Informationen zu den Kennungen dieser Geocodierer und den Regionen, die die von diesen Geocodierern verarbeiteten Standorte enthalten, finden Sie im Abschnitt „DB2GSE.SPATIAL_GEOCODER“ auf Seite 145.
- Räumliche Bezugssysteme, die Sie verwenden können. Informationen zu den Kennungen und ihren Beschreibungen finden Sie im Abschnitt „DB2GSE.SPATIAL_REF_SYS“ auf Seite 145.

DB2GSE.COORD_REF_SYS

Wenn Sie eine Datenbank für räumliche Operationen aktivieren, registriert der Spatial Extender die zur Verfügung stehenden Koordinatensysteme in einer Katalogtabelle. Ausgewählte Spalten aus dieser Tabelle bilden die Katalogsicht DB2GSE.COORD_REF_SYS, die in Tabelle 35 beschrieben ist.

Tabelle 35. Spalten in der Katalogsicht DB2GSE.COORD_REF_SYS

Name	Datentyp	Null zulässig?	Inhalt
CSID	INTEGER	Nein	Eindeutige numerische Kennung für dieses Koordinatensystem.
CS_NAME	VARCHAR(64)	Nein	Name dieses Koordinatensystems
AUTH_NAME	VARCHAR(256)	Ja	Name der Organisation, die mit diesem Koordinatensystem arbeitet, z. B. die European Petroleum Survey Group (EPSG).
AUTH_SRID	INTEGER	Ja	Eine numerische Kennung, die diesem Koordinatensystem von der in der Spalte AUTH_NAME angegebenen Organisation zugeordnet wurde.
DESC	VARCHAR(256)	Ja	Beschreibung dieses Koordinatensystems.

Tabelle 35. Spalten in der Katalogsicht DB2GSE.COORD_REF_SYS (Forts.)

Name	Datentyp	Null zulässig?	Inhalt
SRTEXT	VARCHAR(2048)	Nein	Anmerkungstext für dieses Koordinatensystem.

DB2GSE.GEOMETRY_COLUMNS

Wenn Sie eine Schicht erstellen, registriert der Spatial Extender diese durch Aufzeichnen ihrer Kennung und ihrer Informationen in einer Katalogtabelle. Ausgewählte Spalten aus dieser Tabelle bilden die Katalogsicht DB2GSE.GEOMETRY_COLUMNS, die in Tabelle 36 beschrieben ist.

Tabelle 36. Spalten in der Katalogsicht DB2GSE.GEOMETRY_COLUMNS

Name	Datentyp	Null zulässig?	Inhalt
LAYER_CATALOG	VARCHAR(30)	Ja	NULL. Ein LAYER_CATALOG-Konzept existiert im Spatial Extender nicht.
LAYER_SCHEMA	VARCHAR(30)	Nein	Schema der Tabelle oder Sicht, die die als diese Schicht registrierte Spalte enthält.
LAYER_TABLE	VARCHAR(128)	Nein	Name der Tabelle oder Sicht, die die als diese Schicht registrierte Spalte enthält.
LAYER_COLUMN	VARCHAR(128)	Nein	Name der Spalte, die als diese Schicht registriert wurde.
GEOMETRY_TYPE	INTEGER	Nein	Datentyp der Spalte, die als diese Schicht registriert wurde.
SRID	INTEGER	Nein	Kennung des räumlichen Bezugssystems, das für die Werte in der Spalte, die als diese Schicht registriert wurde, verwendet wird.
STORAGE_TYPE	INTEGER	Ja	Informationen dazu, wie DB2 die Werte in der Spalte speichert, die als diese Schicht registriert wurde. Daten in der Spalte STORAGE_TYPE können z. B. darauf hinweisen, dass die Werte als große Objekte (LOBs) gespeichert wurden.

DB2GSE.SPATIAL_GEOCODER

Die verfügbaren Geocodierer sind in einer Katalogtabelle registriert. Ausgewählte Spalten aus dieser Tabelle bilden die Katalogsicht DB2GSE.SPATIAL_GEOCODER, die in Tabelle 37 beschrieben ist.

Tabelle 37. Spalten in der Katalogsicht DB2GSE.SPATIAL_GEOCODER

Name	Datentyp	Null zulässig?	Inhalt
GCID	INTEGER	Nein	Numerische Kennung des Geocodierers.
GC_NAME	VARCHAR(64)	Nein	Namenskennung des Geocodierers.
VENDOR_NAME	VARCHAR(128)	Nein	Name des Lieferanten, von dem der Geocodierer bezogen wurde.
PRIMARY_UDF	VARCHAR(256)	Nein	Vollständig qualifizierter Name des Geocodierers.
PRECISION_LEVEL	INTEGER	Nein	Die Genauigkeit, mit der die Quelldaten den entsprechenden Bezugsdaten entsprechen müssen, damit sie vom Geocodierer erfolgreich verarbeitet werden können.
VENDOR_SPECIFIC	VARCHAR(256)	Ja	Pfad und Name einer Datei, die ein Lieferant verwenden kann, um spezielle Parameter festzulegen, die der Geocodierer unterstützt.
GEO_AREA	VARCHAR(256)	Ja	Geografischer Bereich, der die zu geocodierenden Standorte enthält.
DESCRIPTION	VARCHAR(256)	Ja	Beschreibung des Geocodierers.

DB2GSE.SPATIAL_REF_SYS

Wenn Sie ein räumliches Bezugssystem erstellen, registriert der Spatial Extender dieses durch Aufzeichnen seiner Kennung und seiner Informationen in einer Katalogtabelle. Ausgewählte Spalten aus dieser Tabelle bilden die Katalogsicht DB2GSE.SPATIAL_REF_SYS, die in Tabelle 38 beschrieben ist.

Tabelle 38. Spalten in der Katalogsicht DB2GSE.SPATIAL_REF_SYS

Name	Datentyp	Null zulässig?	Inhalt
SRID	INTEGER	Nein	Benutzerdefinierte Kennung für dieses räumliche Bezugssystem.
SR_NAME	VARCHAR(64)	Nein	Name dieses räumlichen Bezugssystems.
CSID	INTEGER	Nein	Numerische Kennung für das Koordinatensystem, das diesem räumlichen Bezugssystem zugrundeliegt.

Tabelle 38. Spalten in der Katalogsicht DB2GSE.SPATIAL_REF_SYS (Forts.)

Name	Datentyp	Null zulässig?	Inhalt
CS_NAME	VARCHAR(64)	Nein	Name des Koordinatensystems, das diesem räumlichen Bezugssystem zugrundeliegt.
AUTH_NAME	VARCHAR(256)	Ja	Name der Organisation, die die Standards für dieses räumliche Bezugssystem festlegt.
AUTH_SRID	INTEGER	Ja	Die Kennung, die die in der Spalte AUTH_NAME angegebene Organisation diesem räumlichen Bezugssystem zuordnet.
SRTEXT	VARCHAR(2048)	Nein	Anmerkungstext für dieses räumliche Bezugssystem.
FALSEX	FLOAT	Nein	Eine Zahl, die von einem negativen X-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).
FALSEY	FLOAT	Nein	Eine Zahl, die von einem negativen Y-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).
XYUNITS	FLOAT	Nein	Eine Zahl, die bei Multiplikation mit einer dezimalen X-Koordinate oder einer dezimalen Y-Koordinate eine ganze Zahl ergibt, die als 32-Bit-Datenelement gespeichert werden kann.
FALSEZ	FLOAT	Nein	Eine Zahl, die von einem negativen Z-Koordinatenwert abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).
ZUNITS	FLOAT	Nein	Eine Zahl, die bei Multiplikation mit einer dezimalen Z-Koordinate eine ganze Zahl ergibt, die als 32-Bit-Datenelement gespeichert werden kann.
FALSEM	FLOAT	Nein	Eine Zahl, die von einem negativen Maß abgezogen wird, um auf diese Weise eine nicht negative Zahl zu erhalten (eine positive Zahl oder Null).
MUNITS	FLOAT	Nein	Eine Zahl, die bei Multiplikation mit einem dezimalen Maß eine ganze Zahl ergibt, die als 32-Bit-Datenelement gespeichert werden kann.

Kapitel 12. Räumliche Indizes

Da räumliche Spalten zweidimensionale geografische Daten enthalten, erfordern Anwendungen, die diese Spalten abfragen, eine Indexstrategie, die alle Geometrien innerhalb eines bestimmten Bereichs schnell identifiziert. Aus diesem Grund bietet der Spatial Extender den dreistufigen räumlichen Index auf der Basis eines Gitters.

Dieses Kapitel beschreibt diese Art von Index und definiert Richtlinien zur Verwendung dieses Index. Folgende Themen werden hierbei behandelt:

- „Fragment eines Beispielprogramms“
- „B-Baumstruktur-Indizes“ auf Seite 148
- „Möglichkeiten zum Erstellen eines räumlichen Index“ auf Seite 149
- „Wie ein räumlicher Index erstellt wird“ auf Seite 149
- „Richtlinien zur Verwendung eines räumlichen Index“ auf Seite 153

Fragment eines Beispielprogramms

Das folgende Beispiel zeigt, wie ein Index in SQL erstellt und verwendet wird. Sie können weitere Informationen zu den Befehlen CREATE INDEX und CREATE INDEX EXTENSION im Handbuch *SQL Reference* nachschlagen. Beachten Sie, dass Sie nach dem Erstellen des Index Standard-DDL- und DML-Anweisungen verwenden können, die mit den räumlichen Funktionen und Prädikaten arbeiten.

```
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)

```

B-Baumstruktur-Indizes

Die Technologie der räumlichen Indexierung basiert auf dem herkömmlichen hierarchischen B-Baumstruktur-Index, weist jedoch erhebliche Unterschiede auf. Der räumliche Index verwendet eine *Gitterindexierung* zum Indexieren zweidimensionaler räumlicher Spalten. Der B-Baumstruktur-Index kann nur eindimensionale Daten verarbeiten und nicht mit GIS-Informationen verwendet werden. Dieser Abschnitt beschreibt, wie ein B-Baumstruktur-Index strukturiert und verwendet wird.

Die oberste Ebene eines B-Baumstruktur-Index, die als Root-Knoten bezeichnet wird, enthält einen Schlüssel für jeden Knoten auf der nächsten Ebene. Der Wert jedes Schlüssels ist der größte vorhandene Schlüsselwert für den entsprechenden Knoten auf der nächsten Ebene. Je nach der Anzahl der Werte in der Basistabelle sind eventuell mehrere Zwischenknoten erforderlich. Diese Knoten bilden eine Brücke zwischen dem Root-Knoten und den "Blattknoten", die die eigentlichen Zeilen-IDs der Basistabelle enthalten.

Der Datenbank-Manager durchsucht einen B-Baumstruktur-Index ab dem Root-Knoten. Anschließend fährt er mit den Zwischenknoten fort, bis er den Blattknoten mit der Zeilen-ID der Basistabelle erreicht.

Der B-Baumstruktur-Index kann nicht auf eine räumliche Spalte angewandt werden, da die Zweidimensionalität der räumlichen Spalte die Struktur eines räumlichen Index erfordert. Aus dem gleichen Grund können Sie keinen räumlichen Index auf eine nicht räumliche Spalte anwenden. Darüber hinaus kann ein räumlicher Index nicht auf der Basis mehrerer Spalten erstellt werden.

Möglichkeiten zum Erstellen eines räumlichen Index

Ein räumlicher Index kann auf verschiedene Arten erstellt werden:

- Durch Definieren im Fenster **Create Spatial Index**. Anleitungen hierzu finden Sie in „Kapitel 6. Räumliche Indizes erstellen“ auf Seite 67.
- Durch Aufrufen der gespeicherten Prozedur `db2gse.gse_enable_idx` in einem Anwendungsprogramm. Informationen zu dieser gespeicherten Prozedur finden Sie in „Kapitel 9. Gespeicherte Prozeduren“ auf Seite 83.
- Durch Eingeben des Befehls **db2 create index** mit der Funktion **spatial_index** in der USING-Klausel. Zum Beispiel:

```
create index storesx1 on customers (loc) using db2gse.spatial_index(10e0,  
100e0, 1000e0)
```

Zur Verwendung räumlicher Daten muss der Datenbank-Designer ihre relative Größenverteilung verstehen. Der Designer muss die optimale Größe und die Anzahl der Gitterstufen festlegen, mit denen der räumliche Index erstellt werden soll.

Die Gitterstufen <Gitterstufe 1>, <Gitterstufe 2> und <Gitterstufe 3> werden durch Vergrößern der Zellengröße eingegeben. Somit muss die zweite Schicht eine höhere Zellengröße haben als die erste und die dritte Schicht eine höhere als die zweite. Die erste Gitterstufe ist verbindlich; Sie können die zweite und dritte mit einem Nullwert doppelter Genauigkeit (0.0e0) inaktivieren.

Wie ein räumlicher Index erstellt wird

Ein räumlicher Index wird durch Verwendung von *Umschlägen* erstellt. Der Umschlag ist selbst eine Geometrie und stellt den Minimal- und Maximalbereich für X und Y einer Geometrie dar. Für die meisten Geometrien ist der Umschlag ein Kasten; für horizontale und vertikale Linienfolgen ist es jedoch eine Zwei-Punkt-Linienfolge. Für Punkte ist der Umschlag selbst ein Punkt. Weitere Informationen zu Umschlägen finden Sie im Abschnitt „Umschlag“ auf Seite 163.

Der räumliche Index wird in einer räumlichen Spalte konstruiert; hierzu werden ein oder mehrere Einträge für die Schnittpunkte der Umschläge der einzelnen Geometrien mit dem Gitter vorgenommen. Ein Schnittpunkt wird als interne ID der Geometrie und als Mindestkoordinaten für X und Y der geschnittenen Gitterzellen aufgezeichnet. Das Polygon in Abb. 7 auf Seite 150 schneidet beispielsweise das Gitter an den Koordinaten (20,30), (30,30), (40,30), (20,40), (30,40), (40,40), (20,50), (30,50) und (40,50). Siehe Tabelle 39 auf Seite 151 für Mindestwerte der X- und Y-Koordinaten für alle Geometrien in Abb. 7 auf Seite 150.

Wenn mehrere Gitterstufen vorhanden sind, versucht der Spatial Extender, die niedrigste mögliche Gitterstufe zu verwenden. Wenn eine Geometrie vier oder mehr Gitterzellen auf einer angegebenen Ebene geschnitten hat, wird sie auf die nächsthöhere Stufe hochgestuft. Bei einem räumlichen Index mit den drei Gitterstufen 10.0e0, 100.0e0 und 1000.0e0 schneidet der Spatial Extender daher zunächst jede Geometrie mit dem Gitter der Ebene 10.0e0. Wenn eine Geometrie vier oder mehr 10.0e0-Gitterzellen schneidet, wird sie hochgestuft und mit dem Gitter der Ebene 100.0e0 geschnitten. Wenn sich vier oder mehr Schnittpunkte auf der Ebene 100.0e0 ergeben, wird die Geometrie auf die Ebene 1000.0e0 hochgestuft. Auf der Ebene 1000.0e0 müssen die Schnittpunkte im räumlichen Index eingegeben werden, da dies die höchste mögliche Ebene ist. Abb. 7 verdeutlicht, wie vier verschiedene Arten von Geometrien ein 10.0e-Gitter schneiden. Alle 23 Schnittpunkte für die vier Geometrien werden im räumlichen Index aufgezeichnet.

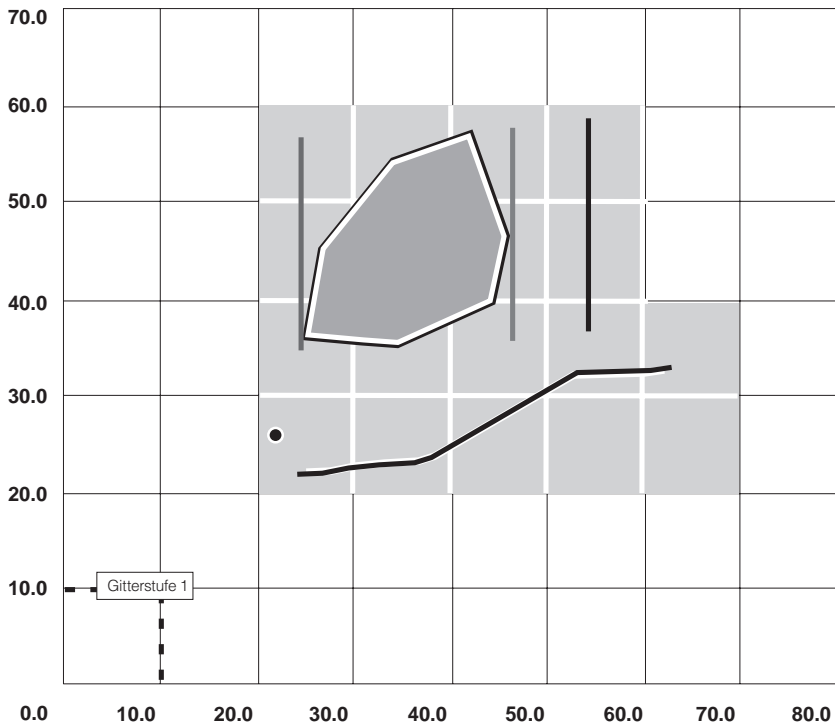


Abbildung 7. Anwendung einer 10.0e0-Gitterstufe

Tabelle 39 auf Seite 151 listet die Geometrien und ihre entsprechenden Gitterschnittpunkte auf. Die Umschläge der vier verschiedenen Geometrietypen schneiden das 10.0e-Gitter. Die Mindestkoordinaten für X und Y jeder geschnittenen Gitterzelle werden in dem räumlichen Index eingegeben.

Tabelle 39. Die 10.0e0-Gitterzelleneinträge für die Beispielgeometrien

Geometrie	Gitter X	Gitter Y
Polygon	20.0	30.0
Polygon	30.0	30.0
Polygon	40.0	30.0
Polygon	20.0	40.0
Polygon	30.0	40.0
Polygon	40.0	40.0
Polygon	20.0	50.0
Polygon	30.0	50.0
Polygon	40.0	50.0
Vertikale Linienfolge	50.0	30.0
Vertikale Linienfolge	50.0	40.0
Vertikale Linienfolge	50.0	50.0
Punkt (Point)	20.0	20.0
Horizontale Linienfolge	20.0	20.0
Horizontale Linienfolge	30.0	20.0
Horizontale Linienfolge	40.0	20.0
Horizontale Linienfolge	50.0	20.0
Horizontale Linienfolge	60.0	20.0
Horizontale Linienfolge	20.0	30.0
Horizontale Linienfolge	30.0	30.0
Horizontale Linienfolge	40.0	30.0
Horizontale Linienfolge	50.0	30.0
Horizontale Linienfolge	60.0	30.0

Abb. 8 auf Seite 152 zeigt, wie die Anzahl der Schnittpunkte erheblich auf acht reduziert werden kann durch Hinzufügen der Gitterstufen 30.0e0 und 60.0e0. In diesem Fall werden das als Geometrie 1 gekennzeichnete Polygon auf die Gitterstufe 30.0e0 und die als Geometrie 4 gekennzeichnete Linienfolge auf die Gitterstufe 60.0e0 hochgestuft. Statt der neun und zehn Schnittpunkte, die die Geometrien auf der Stufe 10.0e0 hatten, haben sie nach dem Hochstufen nur noch zwei Schnittpunkte.

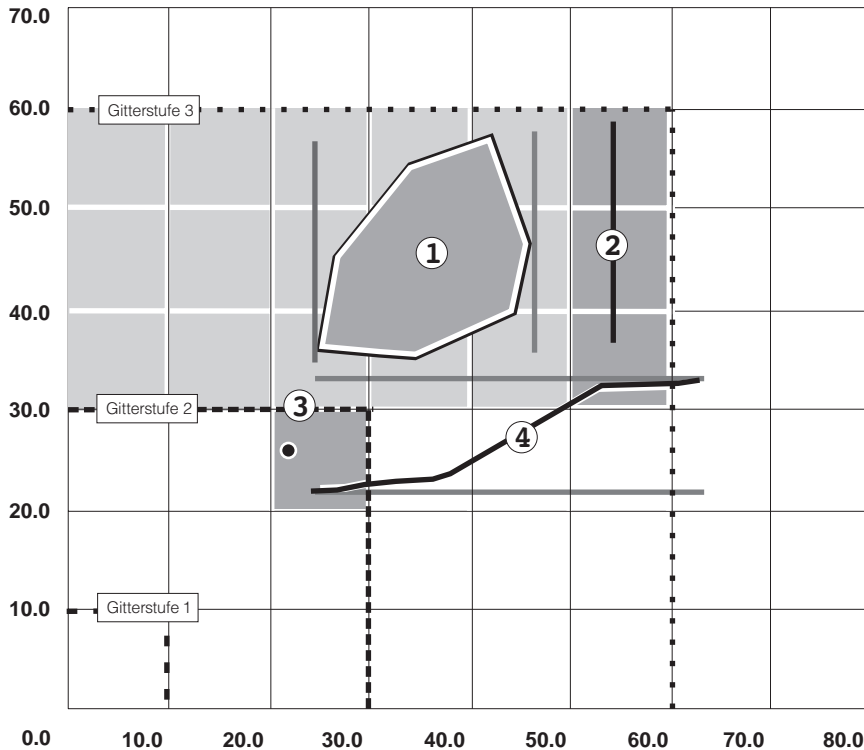


Abbildung 8. Auswirkung beim Hinzufügen der Gitterstufen 30.0e0 und 60.0e0. Der Umschlag des als Geometrie 1 gekennzeichneten Umschlags schneidet neun Gitterzellen. Der Umschlag der als Geometrie 2 vertikalen Linienfolge schneidet drei Gitterzellen. Der Umschlag des als Geometrie 3 gekennzeichneten Punkts schneidet nur eine Gitterzelle. Der Umschlag der als Geometrie 4 gekennzeichneten Linienfolge schneidet zehn Gitterzellen.

Der Spatial Extender verwendet die in der Anweisung CREATE INDEX angegebenen Gitterstufenparameter und prüft jedes räumliche Objekt, um die Koordinaten und die Anzahl der Gitterblöcke, in denen die Objekte enthalten sind, festzustellen. In Abb. 8 sind die Gitterstufen 10.0e0, 30.0e0 und 60.0e0 dargestellt mit zunehmenden Linienstärken und unterschiedlichen Grauschattierungen. Die Zellschnittpunkte der vertikalen Linienfolgen- und des Punktumschlags werden in dem Index auf der Stufe 10.0e0 eingegeben, da beide weniger als vier Schnittpunkte erzeugen. Das Polygon schneidet neun 10.0e0-Gitterzellen und wird daher hochgestuft auf die 30.0e0-Stufe. Auf dieser Stufe schneidet das Polygon zwei Gitterzellen, die in den Index eingegeben werden. Die als Geometrie 4 gekennzeichnete Linienfolge schneidet zehn 10.0e0-Gitterzellen und wird daher hochgestuft auf die 30.0e0-Gitterstufe. Auch auf dieser Stufe schneidet sie noch sechs Gitterzellen und wird daher erneut hochgestuft auf die 60.0e0-Gitterstufe, wo sie noch zwei Schnittpunkte generiert. Die 60.0e0-Gitterschnittpunkte der Linienfolge werden anschließend in den Index eingegeben. Hätte die Linienfolge auf dieser Ebene vier oder mehr Schnittpunkte generiert, wären diese dennoch in den Index eingetragen

worden, da dies die höchste Stufe ist, auf die eine Geometrie hochgestuft werden kann.

Tabelle 40. Die Schnittpunkte der Geometrien im dreistufigen Index

Geometrie	Gitter X	Gitter Y
<i>Die Schnittpunkte zwischen der vertikalen Linienfolge und dem Punkt auf der Stufe 1 (10.0e0-Gittergröße)</i>		
2	50.0	30.0
2	50.0	40.0
2	50.0	50.0
3	20.0	20.0
<i>Die Schnittpunkte des Polygons auf der Stufe 2 (30.0e0-Gittergröße)</i>		
1	0.0	30.0
1	30.0	30.0
<i>Die Schnittpunkte der Linienfolge auf der Stufe 3 (60.0e0-Gittergröße)</i>		
4	0.0	0.0
4	60.0	0.0

Der Spatial Extender erstellt nicht wirklich eine Polygongitterstruktur. Der Spatial Extender manifestiert jede Gitterstufe parametrisch durch die Definition des Ursprungs am Abstand X,Y des räumlichen Bezugssystems der Spalten. Anschließend wird das Gitter in den positiven Koordinatenbereich erweitert. Mit einem parametrischen Gitter generiert der Spatial Extender die Schnittpunkte mathematisch.

Richtlinien zur Verwendung eines räumlichen Index

Der Spatial Extender verbessert mit Hilfe eines räumlichen Index die Leistung einer räumlichen Abfrage. Betrachten Sie die einfachste und wahrscheinlich häufigste räumliche Abfrage, die Feldabfrage. Diese Abfrage fordert den Spatial Extender auf, alle Geometrien zurückzugeben, die vollständig oder teilweise innerhalb eines benutzerdefinierten Feldes liegen. Wenn kein Index vorhanden ist, muss der Spatial Extender alle Geometrien mit dem Feld vergleichen. Mit einem Index kann der Spatial Extender jedoch alle Indexeinträge finden, deren linke untere Koordinate größer oder gleich der des Feldes ist und deren obere rechte Koordinate kleiner oder gleich der des Feldes ist. Da der Index nach diesem Koordinatensystem geordnet ist, kann der Spatial Extender schnell eine Liste der Kandidatengeometrien abrufen. Dieser Prozess wird als *erster Arbeitsgang* bezeichnet.

Ein *zweiter Arbeitsgang* ermittelt, ob die Umschläge der Kandidaten einen Kasten schneiden. Eine für den ersten Arbeitsgang qualifizierte Geometrie, deren Gitterzellenumschlag den Kasten schneidet, kann selbst einen Umschlag haben, der den Kasten nicht schneidet.

Ein *dritter Arbeitsgang* vergleicht die tatsächlichen Koordinaten des Kandidaten mit dem Kasten, um festzustellen, ob ein Teil der Geometrie tatsächlich in dem Kasten liegt. Dieser letzte und relativ komplexe Vergleichsprozess arbeitet mit einer Liste von Kandidaten, die aus einer Untermenge der Gesamtpopulation bestehen, die durch die beiden ersten Arbeitsgänge erheblich reduziert wurde.

Alle räumlichen Abfragen mit Ausnahme der Funktion `EnvelopesIntersect` führen die drei Arbeitsgänge aus. Diese Funktion führt nur die beiden ersten Arbeitsgänge aus. Die Funktion `EnvelopesIntersect` wurde für Anzeigeoperationen konzipiert, die häufig ihre eigenen integrierten Ausschnittprozeduren verwenden und die Unterteilung des dritten Arbeitsgangs nicht erfordern.

Gitterzellengröße auswählen

Die unregelmäßige Form der Geometrieumschläge kompliziert die Auswahl der Gitterzellengröße. Wegen dieser Unregelmäßigkeit schneiden manche Geometrieumschläge mehrere Gitter, während andere in eine einzige Gitterzelle passen. Andererseits schneiden manche Gitterzellen, je nach der räumlichen Verteilung der Daten, viele Geometrieumschläge.

Damit ein räumlicher Index gut funktioniert, ist die Auswahl der richtigen Anzahl und Größe der Gitter sehr wichtig. Betrachten Sie eine räumliche Spalte mit einer Geometrie einheitlicher Größe. In diesem Fall genügt eine einzige Gitterstufe. Beginnen Sie mit einer Gitterzellengröße, die den durchschnittlichen Geometrieumschlag umfasst. Beim Testen Ihrer Anwendung stellen Sie eventuell fest, dass durch die Vergrößerung der Gitterzellengröße die Leistung Ihrer Abfragen verbessert wird. Dies liegt daran, dass jede Gitterzelle mehrere Geometrien enthält und der erste Arbeitsgang nicht qualifizierte Geometrien schneller verwerfen kann. Sie werden jedoch ebenfalls feststellen, dass die Leistung bei einer weiteren Vergrößerung der Zellengröße wieder nachlässt. Das liegt daran, dass der zweite Arbeitsgang schließlich mit mehr Kandidaten arbeiten muss.

Anzahl der Stufen auswählen

Wenn die Objekte, die Sie indexieren wollen, ungefähr gleich groß sind, können Sie mit einer einzigen Gitterstufe arbeiten. Dies ist zwar richtig; es enthalten jedoch nicht alle Spalten eine Geometrie der gleichen relativen Größe. Normalerweise können Geometrien räumlicher Spalten in Intervallen verschiedener Größen gruppiert werden. Denken Sie beispielsweise an ein Straßennetz, in dem die Geometrien in Straßen, Hauptstraßen und Fernstraßen unterteilt sind. Die Straßen sind alle ungefähr gleich lang und können in einer Intervallgröße gruppiert werden. Dies gilt auch für die Hauptstraßen und Fernstraßen. Die Straßen, die für ein Größenintervall stehen, können daher in der erste Gitterstufe gruppiert werden, die Hauptstraßen in der zweiten und die Fernstraßen in der dritten. Ein weiteres Beispiel bildet eine Spalte "Landkreisflächen", die Gruppen kleiner städtischer Flächen enthält, die von größeren ländlichen Flächen umgeben sind. In diesem Fall gibt es zwei Größenintervalle und zwei Gitterstufen, eine für die kleinen städtischen Flächen und eine andere für die größeren ländlichen Flächen. Diese Situationen sind sehr häufig und erfordern die Verwendung eines mehrstufigen Gitters.

Zur Auswahl der Zellengröße jeder Gitterstufe wählen Sie Gitterzellengrößen aus, die etwas größer sind als jedes Größenintervall. Testen Sie den Index durch Ausführen von Abfragen, gegen die räumlichen Spalten.

Jede zusätzliche Stufe erfordert eine zusätzliche Indexsuche. Versuchen Sie, die Gittergrößen leicht nach oben oder unten anzupassen, um festzustellen, ob dadurch eine Verbesserung der Leistung erzielt wird.

Kapitel 13. Geometrien und zugeordnete räumliche Funktionen

In diesem Kapitel werden die Informationseinheiten, die so genannten *Geometrien* beschrieben, die aus Koordinaten bestehen und geografische Funktionen symbolisieren. Das Kapitel stellt außerdem räumliche Funktionen vor, die Geometrien als Eingabe verwenden und Ergebnisse zurückgeben, mit denen Sie geografische Eigenschaften einfacher analysieren und räumliche Daten zwischen geografischen Informationssystemen verschieben können. Folgende Themen werden hierbei behandelt:

- Das Wesen der Geometrien
- Merkmale der Geometrien; Funktionen, die Informationen zu diesen Merkmalen zurückgeben
- Exemplarfähige Geometrien; Funktionen, die darauf ausgeführt werden
- Funktionen zur Ausführung folgender Aufgaben:
 - Aufzeigen von Beziehungen und Vergleichen zwischen geografischen Merkmalen
 - Generieren von Geometrien
 - Umwandeln von Geometriewerten in importierbare und exportierbare Formate

Informationen zu Geometrien

Das Oxford American Dictionary definiert *Geometrie* als den „Zweig der Mathematik, der sich mit den Eigenschaften von und den Beziehungen zwischen Geraden, Winkeln, Flächen und Körpern beschäftigt.“ Am 11. August 1997 hat das Open GIS Consortium Inc. (OGC) in seiner Veröffentlichung *Open GIS Features for ODBC (SQL) Implementation Specification* eine andere Definition für den Begriff geprägt. Das Wort *Geometrie* wurde ausgewählt, um die geometrischen Eigenschaften zu kennzeichnen, mit deren Hilfe Kartographen seit mehr als einem Jahrtausend die Erde darstellen. Eine sehr abstrakte Definition dieser neuen Bedeutung der Geometrie wäre „ein Punkt oder eine Gruppierung von Punkten, die ein Merkmal der Oberfläche symbolisieren.“

Im Spatial Extender wäre eine *operative* Definition der Geometrie „ein Modell einer geografischen Eigenschaft.“ Das Modell kann als Koordinaten der Eigenschaft ausgedrückt werden oder in manchen Fällen als visuelles Symbol. Das Modell umfasst Informationen; die Koordinaten kennzeichnen beispielsweise die Position der Eigenschaft in bezug auf feste Referenzpunkte, und das Symbol umreißt ihre Form.

Darüber hinaus kann das Modell verwendet werden, um Informationen zu erstellen; die Funktion `ST_Overlaps` kann beispielsweise die Koordinaten von zwei benachbarten Regionen als Eingabe verwenden und als Ausgabe Informationen dazu zurückgeben, ob sich die Regionen überlappen oder nicht.

Die Koordinaten eines Merkmals, das von einer Geometrie symbolisiert wird, werden als *Merkmale* der Geometrie betrachtet. Verschiedene Arten von Geometrien haben ebenfalls Merkmale, z. B.:

- Ein *Innenbereich* steht für den Inhalt eines Merkmals, das von der Geometrie symbolisiert wird.
- Ein *Außenbereich* steht für den Bereich um das Merkmal herum.
- Eine *Begrenzung* steht für die Linie, an der der Inhalt aufhört und der Umgebungsbereich beginnt.

Diese und weitere Merkmale werden im Abschnitt „Merkmale und zugeordnete Funktionen“ auf Seite 160 beschrieben.

Die vom Spatial Extender unterstützten Geometrien bilden eine Hierarchie, wie in Abb. 9 auf Seite 159 dargestellt. Sechs Teile der Hierarchie sind exemplarfähig; sie können als visuelle Symbole ausgedrückt werden, die ebenfalls in der Abbildung dargestellt sind.

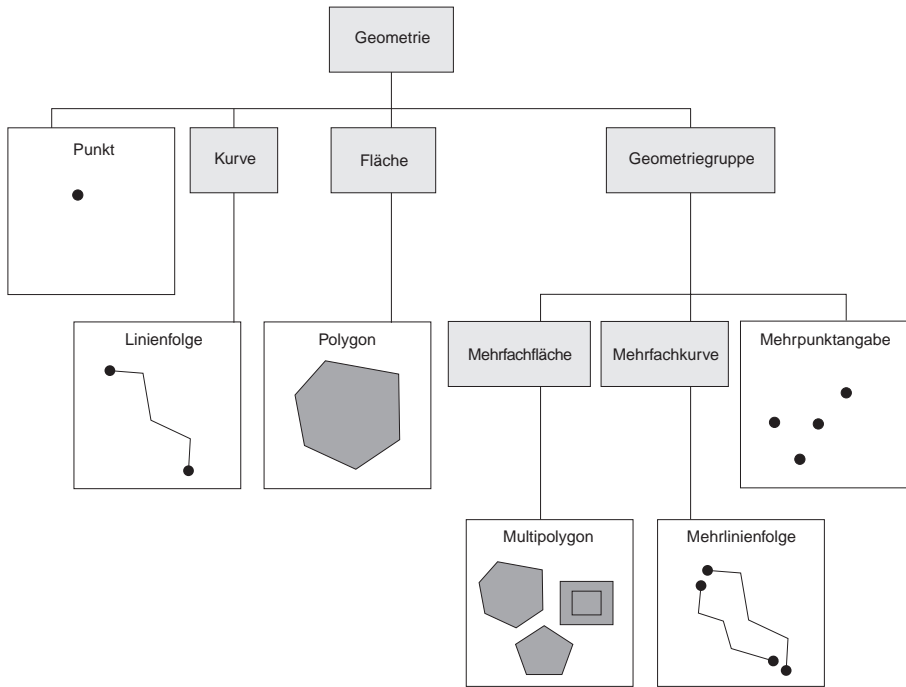


Abbildung 9. Hierarchie der vom Spatial Extender unterstützten Geometrien. Exemplarfähige Geometrien können als visuelle Symbole ausgedrückt werden. Diese Symbole werden unter den Namen dieser Geometrien angezeigt.

Wie in Abb. 9 angedeutet, ist eine Superklasse namens *Geometrie* der Root-Knoten der Hierarchie. Die Subtypen sind in zwei Kategorien unterteilt: die Subtypen der Basisgeometrien und die Subtypen der homogenen Gruppen. Die Basisgeometrien umfassen:

- *Punkte* für die eindeutigen Merkmale, die den Ort belegen, an dem eine Ost-West-Koordinatenlinie (z. B. ein Breitengrad) eine Nord-Süd-Koordinate (z. B. einen Längengrad) schneidet. Angenommen, die Notation einer Karte mit großem Maßstab zeigt jede Stadt auf der Karte an einem Schnittpunkt eines Breiten- und Längengrads. Bei diesem Maßstab könnte jede Stadt durch einen Punkt symbolisiert werden.
- *Linienfolgen* für lineare geografische Merkmale (z. B. Straßen, Kanäle oder Rohrleitungen).
- *Polygone* für mehrseitige geografische Merkmale (z. B. Industriegebiete, Wälder oder Naturschutzgebiete).

Diese homogenen Gruppen umfassen:

- *Mehrpunktangaben* für mehrteilige Merkmale, deren Komponenten sich jeweils am Schnittpunkt einer Ost-West- und einer Nord-Süd-Koordinate

befinden (z. B. eine Inselkette, deren einzelne Inseln auf den Schnittpunkten eines Breiten- und Längengrades liegen).

- *Mehrlinienfolgen* für mehrteilige Merkmale aus linearen Einheiten der Komponenten (z. B. Fluss- oder Straßennetze).
- *Multipolygone* für mehrteilige Merkmale aus mehrseitigen Einheiten oder Komponenten (z. B. das Ackerland in einer bestimmten Region oder eine Seenplatte).

Homogene Gruppen sind, wie der Name schon andeutet, Gruppen von Basisgeometrien. Neben den gemeinsamen Basisgeometrieeigenschaften haben homogene Gruppen auch eigene Eigenschaften.

Die vom Spatial Extender unterstützten Typen von räumlichen Daten sind Implementierungen der in Abb. 9 auf Seite 159 dargestellten Geometrien. Eine Beschreibung dieser Datentypen finden Sie im Abschnitt „Informationen zu Typen von räumlichen Daten“ auf Seite 45.

Merkmale und zugeordnete Funktionen

In diesem Abschnitt werden die Merkmale von Geometrien beschrieben sowie die räumlichen Funktionen, die diesen Merkmalen zugeordnet sind. Folgende Merkmale werden unterschieden:

- Die Klasse, zu der eine Geometrie gehört
- Koordinaten und Maße
- Innenbereich, Begrenzung und Außenbereich einer Geometrie
- Z-Koordinaten
- Maße
- Die Qualität der Komplexität (einfach oder nicht einfach)
- Die Qualität des Inhalts (leer oder nicht leer)
- Umschlag einer Geometrie
- Dimension
- Die Kennung eines der Geometrie zugeordneten räumlichen Bezugssystems

Klasse

Jede Geometrie gehört zu einer Klasse in der in Abb. 9 auf Seite 159 dargestellten Hierarchie. Wie im Abschnitt „Informationen zu Geometrien“ auf Seite 157 angegeben, gibt es sechs exemplarfähige Subtypen in der Hierarchie – Punkte, Linienfolgen, Polygone, Mehrpunktangaben, Mehrlinienfolgen und Multipolygone. Die Superklasse und andere Subtypen sind nicht exemplarfähig.

Die Funktion `ST_GeometryType` verwendet eine Geometrie und gibt eine Zeichenfolgenkennung für den exemplarfähigen Subtyp zurück. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_GeometryType`“ auf Seite 253.

Der Eingabeparameter für die Funktion `ST_IsValid()` ist ein `ST_Geometry`-Wert. Die Funktion gibt 1 (TRUE) zurück, wenn die Geometrie gültig ist, und 0 (FALSE), wenn sie nicht gültig ist. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_IsValid`“ auf Seite 274.

Koordinaten und Maße

Alle Geometrien beinhalten mindestens eine X-Koordinate und eine Y-Koordinate. Darüberhinaus kann eine Geometrie eine oder mehrere Z-Koordinaten und Maße umfassen. Nachfolgend wird auf folgende Themen eingegangen:

- X- und Y-Koordinaten
- Z-Koordinaten und Maße
- Die Funktion `ST_CoordDim`

X- und Y-Koordinaten

Ein *X-Koordinatenwert* kennzeichnet eine Position relativ zu einem Bezugspunkt im Osten oder Westen. Ein *X-Koordinatenwert* kennzeichnet eine Position relativ zu einem Bezugspunkt im Norden oder Süden. Weitere Informationen hierzu finden Sie in den Abschnitten „Das Wesen der räumlichen Daten“ auf Seite 6 und „Informationen zu Koordinaten und räumlichen Bezugssystemen“ auf Seite 35.

Z-Koordinaten und Maße

In diesem Abschnitt werden Z-Koordinaten, Maße und ihre Verwendung im DB2 Spatial Extender beschrieben.

Z-Koordinaten

Manche Geometrien haben eine zugeordnete Höhe oder Tiefe. Jeder Punkt der Geometrie eines Merkmals kann eine wahlfreie Z-Koordinate enthalten, die eine Höhe oder Tiefe gegenüber der Erdoberfläche angibt.

Mit der Funktion `AsShape` können Sie einen Geometriewert in eine ESRI-Formdarstellung umwandeln. Wenn der Geometriewert Z-Koordinaten oder Maße umfasst, werden diese in der Formdarstellung beibehalten.

Die Prädikatfunktion `Is3d` verwendet eine Geometrie und gibt 1 zurück, wenn die Funktion Z-Koordinaten hat. Andernfalls gibt sie 0 zurück. Weitere Informationen hierzu finden Sie im Abschnitt „`Is3d`“ auf Seite 205.

Maße

Ein Maß ist ein Wert, der Informationen über ein geografisches Merkmal enthält, die zusammen mit den Koordinaten für die Position des Merkmals gespeichert werden. Angenommen, Sie stellen in Ihrem GIS Transportsysteme dar. Wenn Ihre Anwendung Werte verarbeiten soll, die lineare Entfernungen

(Luftlinie) angeben, können Sie diese Werte zusammen mit den Koordinaten für die Position der Systeme speichern. Maße werden als Zahlen mit doppelter Genauigkeit gespeichert.

Z-Koordinaten und Maße verwenden

In DB2 Spatial Extender dienen Z-Koordinaten und Maße zur Übermittlung von Informationen an Anwendungen und nicht zur Definition von Standorten. (Beispiele finden Sie unter "Koordinatensysteme, Koordinaten und Maße", einem Unterabschnitt in Kapitel 3 "Ressourcen einrichten".) Wenn die meisten räumlichen Funktionen Punkte verarbeiten, die Z-Koordinaten und/oder Maße beinhalten, ignorieren die Funktionen dementsprechend diese Werte. Andere räumliche Funktionen arbeiten allerdings mit Z-Koordinaten und Maßen:

- Mit den Funktionen `Is3d`, `IsMeasured` und `ST_CoordDim` können Sie ermitteln, ob eine Geometrie eine Z-Koordinate und/oder ein Maß beinhaltet.
- Mit den Funktionen `M` und `Z` können Sie das Maß und die Z-Koordinate eines Punkts ermitteln.
- Das Prädikat `IsMeasured` verwendet eine Geometrie und gibt 1 (TRUE) zurück, wenn Sie Maße enthält und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „`IsMeasured`“ auf Seite 206.

Die Funktion `ST_CoordDim`

`ST_CoordDim` gibt einen Wert zurück, der die Koordinatentypen einer Geometrie angibt und ob die Geometrie auch Maße enthält. Dieser Wert wird als *Koordinatendimension* bezeichnet.

`ST_CoordDim` kann die Koordinatendimension 2, 3 oder 4 zurückgeben:

- Wenn die Eingabe für `ST_CoordDim` ein Punkt ist, bedeutet der Rückgabewert 2, dass der Punkt aus einer X-Koordinate und einer Y-Koordinate besteht. Wenn es sich bei der Eingabe um einen anderen Geometrietyt als einen Punkt handelt, bedeutet 2, dass jeder Punkt in dieser Geometrie aus einer X-Koordinate und einer Y-Koordinate besteht.
- Wenn die Eingabe für `ST_CoordDim` ein Punkt ist, bedeutet der Rückgabewert 3, dass der Punkt aus einer X-Koordinate, einer Y-Koordinate und entweder einer Z-Koordinate oder einem Maß besteht. Wenn es sich bei der Eingabe um einen anderen Geometrietyt als einen Punkt handelt, bedeutet der Wert 3, dass jeder Punkt in dieser Geometrie aus einer X-Koordinate, einer Y-Koordinate und entweder einer Z-Koordinate oder einem Maß besteht.
- Wenn die Eingabe für `ST_CoordDim` ein Punkt ist, bedeutet der Rückgabewert 4, dass der Punkt aus einer X-Koordinate, einer Y-Koordinate, einer Z-Koordinate und einem Maß besteht. Wenn es sich bei der Eingabe um einen anderen Geometrietyt als einen Punkt handelt, bedeutet der Wert 4, dass jeder Punkt in dieser Geometrie aus einer X-Koordinate, einer Y-Koordinate, einer Z-Koordinate und einem Maß besteht.

Innenbereich, Begrenzung und Außenbereich

Alle Geometrien belegen eine Position im Raum; diese Position ist durch ihren Innenbereich, ihre Begrenzung und ihren Außenbereich gekennzeichnet. Der Außenbereich einer Geometrie ist der gesamte Raum, der nicht von der Geometrie belegt wird. Die Begrenzung einer Geometrie dient als Schnittstelle zwischen ihrem Innen- und ihrem Außenbereich. Der Innenbereich ist der von der Geometrie belegte Raum. Subtypen übernehmen die Merkmale von Innenbereich und Außenbereich direkt; die Merkmale der Begrenzung unterscheiden sich jedoch für die einzelnen Subtypen.

Die Funktion `ST_Boundary` verwendet eine Geometrie und gibt eine Geometrie zurück, die die Begrenzung der Quellengeometrie darstellt. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Boundary`“ auf Seite 227.

Einfach oder nicht einfach

Manche Subtypen einer Geometrie (Linienfolge, Mehrpunktangaben und Mehrlinienfolgen) sind entweder einfach oder nicht einfach. Ein Subtyp ist einfach, wenn er alle topologischen Regeln des Subtyps einhält; werden nicht alle diese Regeln eingehalten, so ist der Subtyp nicht einfach. Eine Linienfolge ist einfach, wenn sie ihren eigenen Innenbereich nicht schneidet. Eine Mehrpunktangabe ist einfach, wenn keines ihrer Elemente den gleichen Koordinatenraum belegt. Eine Mehrlinienfolge ist einfach, wenn keiner der Innenbereiche ihrer Elemente durch seinen eigenen Innenbereich geschnitten wird.

Die Prädikatfunktion `ST_IsSimple` verwendet eine Geometrie und gibt 1 (TRUE) zurück, wenn es sich um eine einfache Geometrie handelt und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „`ST_IsSimple`“ auf Seite 273.

Leer oder nicht leer

Eine Geometrie ist leer, wenn sie keine Punkte enthält. Der Umschlag, die Begrenzung, der Innenbereich und der Außenbereich einer leeren Geometrie sind NULL. Eine leere Geometrie ist immer einfach; sie kann Z-Koordinaten oder Maße enthalten. Leere Linienfolgen und Mehrlinienfolgen haben die Länge 0. Leere Polygone und Multipolygons haben die Fläche 0.

Die Prädikatfunktion `ST_IsEmpty` verwendet eine Geometrie und gibt 1 (TRUE) zurück, wenn sie leer ist und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „`ST_IsEmpty`“ auf Seite 269.

Umschlag

Der Umschlag einer Geometrie ist die Begrenzungsgeometrie, die durch die minimalen und maximalen Koordinaten (X,Y) gebildet wird. Die Umschläge der Geometrien bilden ein begrenzendes Rechteck, außer in den folgenden besonderen Fällen:

- Der Umschlag eines Punkts ist der Punkt selbst, da seine minimalen und maximalen X-Koordinaten identisch sind und seine minimalen und maximalen Y-Koordinaten identisch sind.
- Der Umschlag einer horizontalen oder vertikalen Linienfolge ist eine Linienfolge, die durch die Begrenzung (die Endpunkte) der Quellenlinie gebildet wird.

Die Funktion `ST_Envelope` verwendet eine Geometrie und gibt eine Begrenzungsgeometrie zurück, die ihren Umschlag darstellt. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Envelope`“ auf Seite 247.

Dimension

Eine Geometrie kann eine Dimension 0, 1 oder 2 haben. Die Dimensionen sind wie folgt definiert:

- 0 Hat weder Länge noch Fläche
- 1 Hat eine Länge
- 2 Hat eine Fläche

Die Subtypen Punkt und Mehrpunktangabe haben die Dimension 0. Punkte stellen dimensionale Merkmale dar, die mit einer einzigen Koordinate modelliert werden können; Mehrpunkt-Subtypen hingegen stellen Daten dar, die mit einer Gruppe von nicht miteinander verbundenen Koordinaten modelliert werden müssen.

Die Subtypen Linienfolge und Mehrlinienfolge haben die Dimension 1. In diesen Unterklassen werden Straßenabschnitte, verzweigte Fluss-Systeme und andere lineare Merkmale gespeichert.

Die Subtypen Polygon und Multipolygon haben die Dimension 2. Merkmale, deren Umfang eine definierbare Fläche umfassen, wie beispielsweise Wälder, Flächen oder Wasserflächen, können als Datentyp Polygon oder Multipolygon dargestellt werden.

Die Dimension ist nicht nur als Merkmal des Subtyps von Bedeutung, sondern spielt auch eine Rolle beim Festlegen der räumlichen Beziehung zwischen zwei Merkmalen. Die Dimension eines resultierenden Merkmals gibt an, ob die Operation erfolgreich war oder nicht. Der Spatial Extender untersucht die Dimension der Merkmale, um festzustellen, wie sie verglichen werden sollen.

Die Funktion `ST_Dimension` verwendet eine Geometrie und gibt ihre Dimension als ganze Zahl zurück. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Dimension`“ auf Seite 241.

Kennung des räumlichen Bezugssystems

Das räumliche Bezugssystem kennzeichnet die Koordinatenumsetzung für jede Geometrie.

Alle in der Datenbank bekannten räumlichen Bezugssysteme können über die Katalogsicht DB2GSE.SPATIAL_REF_SYS aufgerufen werden. Informationen zu dieser Sicht finden Sie im Abschnitt „DB2GSE.SPATIAL_REF_SYS“ auf Seite 145.

Die Funktion ST_SRID verwendet eine Geometrie und gibt ihre räumliche Bezugs-ID als ganze Zahl zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_SRID“ auf Seite 307.

Die Funktion ST_Transform ordnet eine Geometrie einem anderen räumlichen Bezugssystem zu als dem momentan zugeordneten. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Transform“ auf Seite 312.

Exemplarfähige Geometrien und zugeordnete Funktionen

Dieser Abschnitt umreißt die sechs Subtypen exemplarfähiger Geometrien und beschreibt die Funktionen, die damit arbeiten. Es handelt sich um folgende Subtypen:

- Punkte
- Linienfolgen
- Polygone
- Mehrpunktangaben
- Mehrlinienfolgen
- Multipolygone

Eine Illustration der Hierarchie, zu der diese Subtypen gehören, sowie der ihnen zugeordneten visuellen Symbole finden Sie in Abb. 9 auf Seite 159.

Punkte

Ein Punkt ist eine Geometrie mit der Dimension 0, die eine einzelne Position in einem Koordinatensystem belegt. Ein Punkt enthält eine X- und eine Y-Koordinate, die diese Position definiert. Er kann auch eine Z-Koordinate und ein Maß enthalten.

Ein Punkt ist einfach und hat die Begrenzung NULL. Punkte werden häufig verwendet, um Merkmale wie Ölquellen, markante Landschaftspunkte oder Erhebungen zu kennzeichnen.

Funktionen, die nur mit dem Subtyp Punkt arbeiten:

PointFromShape

Verwendet eine Form des Typs Point (Punkt) und die Kennung eines räumlichen Bezugssystems und gibt einen Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „PointFromShape“ auf Seite 218.

ST_Point

Verwendet eine X-Koordinate, die ihr zugeordnete Y-Koordinate und die Kennung des räumlichen Bezugssystems, zu dem diese Koordinaten gehören, und gibt den von diesen Koordinaten definierten Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Point“ auf Seite 295.

ST_PointFromText

Verwendet eine bekannte OGC-Textdarstellung (WKT) eines Punkts und gibt den Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointFromText“ auf Seite 296.

ST_PointFromWKB

Verwendet eine bekannte binäre Darstellung des Typs Polygon und die Kennung eines räumlichen Bezugssystems und gibt ein Polygon zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointFromWKB“ auf Seite 297.

ST_X Gibt den X-Koordinatenwert eines Datentyps ST_Point als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_X“ auf Seite 319.

ST_Y Gibt den Y-Koordinatenwert eines Datentyps ST_Point als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Y“ auf Seite 320.

Z Gibt den Z-Koordinatenwert eines Datentyps ST_Point als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „Z“ auf Seite 321.

M Gibt das Maß eines Datentyps ST_Point als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „M“ auf Seite 213.

Linienfolgen

Eine Linienfolge ist ein eindimensionales Objekt, das als Folge von Punkten gespeichert wird, die einen linear interpolierten Pfad definieren. Die Linienfolge ist einfach, wenn sie ihren Innenbereich nicht schneidet. Die Endpunkte (die Begrenzung) einer geschlossenen Linienfolge belegen den gleichen Punkt im Raum. Eine Linienfolge ist ein Ring, wenn sie geschlossen ist und ihr Innenbereich sich nicht schneidet. Neben den anderen aus der Geometrie der Superklasse übernommenen Merkmalen haben Linienfolgen eine Länge. Linienfolgen werden häufig zur Definition von Straßen, Flüssen oder Stromkabeln verwendet.

Eine einfache Linienfolge, deren Start- und Endpunkt identisch sind, wird als *Ring* bezeichnet.

Die Endpunkte bilden normalerweise die Begrenzung einer Linienfolge, es sei denn, die Linienfolge ist geschlossen; in diesem Fall ist die Begrenzung NULL. Der Innenbereich einer Linienfolge ist der verbundene Pfad zwischen den beiden Endpunkten, es sei denn, die Linienfolge ist geschlossen; in diesem Fall ist der Innenbereich fortlaufend.

Funktionen, die mit Linienfolgen arbeiten:

ST_StartPoint

Verwendet eine Linienfolge und gibt ihren ersten Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_StartPoint“ auf Seite 308.

ST_EndPoint

Verwendet eine Linienfolge und gibt ihren letzten Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Endpoint“ auf Seite 246.

ST_PointN

Verwendet eine Linienfolge und einen Index zum n -ten Punkt und gibt diesen Punkt zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointN“ auf Seite 299.

ST_Length

Verwendet eine Linienfolge und gibt ihre Länge als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Length“ auf Seite 276.

ST_NumPoints

Verwendet eine Linienfolge und gibt die Anzahl ihrer Punkte als ganze Zahl zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_NumPoints“ auf Seite 290.

ST_IsRing

Verwendet eine Linienfolge und gibt 1 (TRUE) zurück, wenn die Linienfolge ein Ring ist und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „ST_IsRing“ auf Seite 271.

ST_IsClosed

Verwendet eine Linienfolge und gibt 1 (TRUE) zurück, wenn die Linienfolge geschlossen ist und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „ST_IsClosed“ auf Seite 267.

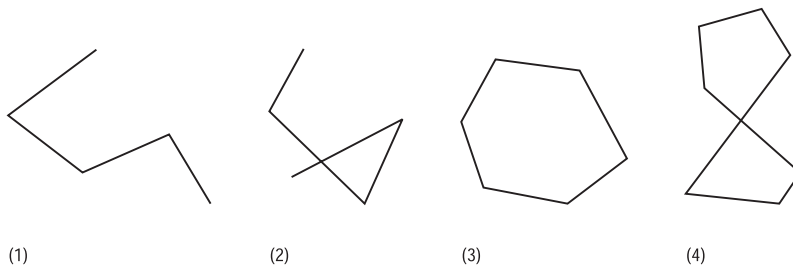


Abbildung 10. Linienfolgeobjekte.

1. Eine einfache, nicht geschlossene Linienfolge.
2. Eine nicht einfache, nicht geschlossene Linienfolge.
3. Eine geschlossene, einfache Linienfolge (ein Ring).
4. Eine geschlossene, nicht einfache Linienfolge (kein Ring).

Polygone

Ein Polygon ist eine zweidimensionale Fläche, die als Folge von Punkten gespeichert wird, die ihren Außenbereich, ihren Begrenzungsring sowie null oder mehr innere Ringe definieren. Die Ringe eines Polygons können sich nicht überlappen. Per Definition sind Polygone immer einfach. Meist definieren Polygone Grundstücke, Wasserflächen und andere Merkmale mit einer räumlichen Ausdehnung.

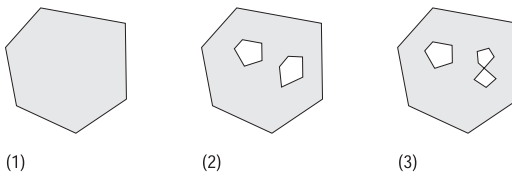


Abbildung 11. Polygone.

1. Ein Polygon, dessen Begrenzung durch einen äußeren Ring definiert ist.
2. Ein Polygon, dessen Begrenzung durch einen äußeren Ring und zwei innere Ringe definiert ist. Die Fläche in den inneren Ringen ist Teil des Außenbereichs der Polygone.
3. Ein zulässiges Polygon, da sich die Ringe an einem einzigen Tangentialpunkt berühren.

Der Außenbereich und alle inneren Ringe definieren die Begrenzung eines Polygons, und der zwischen den Ringen eingeschlossene Bereich definiert den Innenbereich des Polygons. Die Ringe eines Polygons können sich in einem Tangentialpunkt berühren, aber niemals schneiden. Neben den anderen aus der Geometrie der Superklasse übernommenen Merkmalen haben Polygone eine Fläche.

Funktionen, die mit Polygonen arbeiten:

ST_Area

Verwendet ein Polygon und gibt seine Fläche als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Area“ auf Seite 223.

ST_ExteriorRing

Verwendet ein Polygon und gibt seinen äußeren Ring als Linienfolge zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_ExteriorRing“ auf Seite 250.

ST_NumInteriorRing

Verwendet ein Polygon und gibt die Anzahl der darin enthaltenen inneren Ringe zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_NumInteriorRing“ auf Seite 289.

ST_InteriorRingN

Verwendet ein Polygon und einen Index und gibt den n -ten inneren Ring als Linienfolge zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_InteriorRingN“ auf Seite 259.

ST_Centroid

Verwendet ein Polygon und gibt einen Punkt zurück, der den Mittelpunkt der Ausdehnung des Polygons darstellt. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Centroid“ auf Seite 231.

ST_PointOnSurface

Verwendet ein Polygon und gibt einen Punkt zurück, der garantiert in der Fläche des Polygons liegt. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointOnSurface“ auf Seite 300.

ST_Perimeter

Verwendet ein Polygon und gibt den Umfang seiner Fläche zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Perimeter“ auf Seite 294.

Mehrpunktangaben

Eine Mehrpunktangabe ist eine Gruppe von Punkten; sie hat wie ihre Elemente die Dimension 0. Eine Mehrpunktangabe ist einfach, wenn ihre Elemente keine identischen Koordinatenräume belegen. Die Begrenzung einer Mehrpunktangabe ist NULL. Mehrpunktangaben können zur Definition von Phänomenen wie Funkverteilungsmustern oder den Zentren eines Epidemieausbruchs verwendet werden.

Funktionen, die mit Mehrpunktangaben arbeiten:

ST_NumGeometries

Verwendet eine homogene Gruppe und gibt die Anzahl der darin enthaltenen Basisgeometrieelemente zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_NumGeometries“ auf Seite 288.

ST_GeometryN

Verwendet eine homogene Gruppe und einen Index und gibt die n-te Basisgeometrie zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_GeometryN“ auf Seite 252.

Mehrlinienfolgen

Eine Mehrlinienfolge ist eine Gruppe von Linienfolgen. Mehrlinienfolgen sind einfach, wenn sie sich lediglich in den Endpunkten der Linienfolgen berühren, aber nicht schneiden. Mehrlinienfolgen sind nicht einfach, wenn sich die Innenbereiche der Linienfolgen schneiden.

Die Begrenzung einer Mehrlinienfolge sind die sich nicht berührenden Endpunkte der Linienfolgeelemente. Die Mehrlinienfolge gilt als geschlossen, wenn alle ihre Linienfolgen geschlossen sind. Die Begrenzung einer Mehrlinienfolge ist NULL, wenn sich alle Endpunkte aller ihrer Elemente berühren. Neben den anderen aus der Geometrie der Superklasse übernommenen Merkmalen haben Mehrlinienfolgen eine Länge. Mehrlinienfolgen werden zur Definition von Bach- der Straßennetzen verwendet.

Funktionen, die mit Mehrlinienfolgen arbeiten:

ST_Length

Verwendet eine Mehrlinienfolge und gibt die kumulierte Länge aller ihrer Linienfolgen als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Length“ auf Seite 276.

ST_IsClosed

Verwendet eine Mehrlinienfolge und gibt 1 (TRUE) zurück, wenn die Mehrlinienfolge geschlossen ist und andernfalls 0 (FALSE). Weitere Informationen hierzu finden Sie im Abschnitt „ST_IsClosed“ auf Seite 267.

ST_NumGeometries

Verwendet eine homogene Gruppe und gibt die Anzahl der darin enthaltenen Basisgeometrieelemente zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_NumGeometries“ auf Seite 288.

ST_GeometryN

Verwendet eine homogene Gruppe und einen Index und gibt die n-te Basisgeometrie zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_GeometryN“ auf Seite 252.

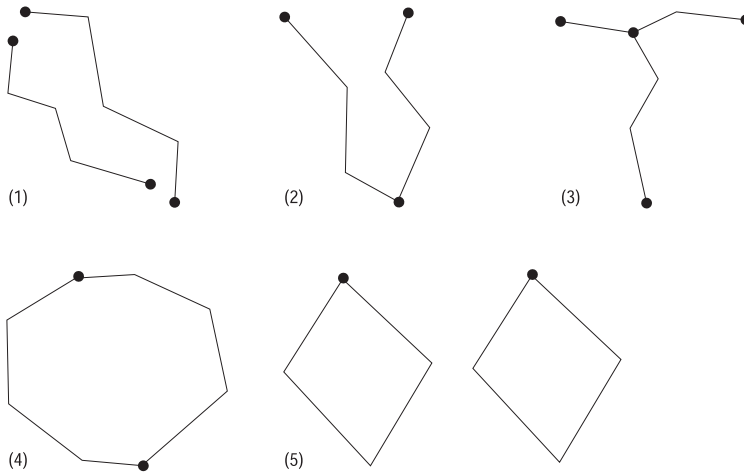


Abbildung 12. Mehrlinienfolgen.

1. Eine einfache Mehrlinienfolge, deren Begrenzung durch die vier Endpunkte ihrer beiden Linienfolgen definiert ist.
2. Eine einfache Mehrlinienfolge, da sich nur die Endpunkte der Linienfolgen berühren. Die Begrenzung ist durch die beiden sich nicht berührenden Endpunkt definiert.
3. Eine nicht einfache Mehrlinienfolge, da der Innenbereich einer ihrer Linienfolgen geschnitten wird. Die Begrenzung dieser Mehrlinienfolge ist durch die vier Endpunkte einschließlich des Verbindungspunkts definiert.
4. Eine einfache, nicht geschlossene Mehrlinienfolge. Sie ist nicht geschlossen, da ihre Linienfolgen nicht geschlossen sind. Sie ist einfach, weil keiner der Innenbereiche der darin enthaltenen Linienfolgen geschnitten wird.
5. Eine einfache geschlossene Mehrlinienfolge. Sie ist geschlossen, weil alle ihre Elemente geschlossen sind. Sie ist einfach, weil keines ihrer Elemente im Innenbereich geschnitten wird.

Multipolygone

Die Begrenzung eines Multipolygons ist die kumulierte Länge der äußeren und inneren Ringe seiner Elemente. Der Innenbereich eines Multipolygons ist definiert als die kumulierten Innenbereiche seiner Elementpolygone. Die Begrenzung der Elemente eines Multipolygons können sich nur in einem Tangentialpunkt berühren. Neben den weiteren vom Typ `ST_MultiSurface` übernommenen Merkmalen haben Multipolygone eine Fläche. Multipolygone definieren Merkmale wie beispielsweise ein Waldstück oder ein nicht zusammenhängendes Stück Land wie beispielsweise eine Inselkette.

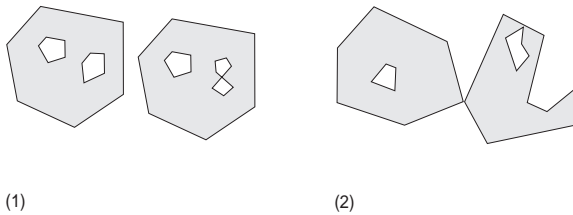


Abbildung 13. Multipolygone.

1. Ein Multipolygon mit zwei Polygonelementen. Die Begrenzung ist durch die beiden äußeren Ringe und die drei inneren Ringe definiert.
2. Ein Multipolygon mit zwei Polygonelementen. Die Begrenzung ist durch die beiden äußeren Ringe und die beiden inneren Ringe definiert. Die beiden Polygonelemente berühren sich an einem Tangentialpunkt.

Funktionen, die mit Multipolygonen arbeiten:

ST_Area

Verwendet eine Multipolygon und gibt die kumulierte Fläche seiner Polygonelemente als Zahl mit doppelter Genauigkeit zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Area“ auf Seite 223.

ST_Centroid

Verwendet ein Multipolygon und gibt einen Punkt zurück, der seinen geometrisch gewichteten Mittelpunkt darstellt. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Centroid“ auf Seite 231.

ST_NumGeometries

Verwendet eine homogene Gruppe und gibt die Anzahl der darin enthaltenen Basisgeometrielemente zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_NumGeometries“ auf Seite 288.

ST_GeometryN

Verwendet eine homogene Gruppe und einen Index und gibt die n-te Basisgeometrie zurück. Weitere Informationen hierzu finden Sie im Abschnitt „ST_GeometryN“ auf Seite 252.

Funktionen, die Beziehungen und Vergleiche zeigen, Geometrien generieren und Werteformate umsetzen

In den vorangegangenen Abschnitten wurden drei Kategorien räumlicher Funktionen vorgestellt:

- Funktionen zu den Merkmalen der Geometrien
- Funktionen zu den spezifischen Geometrien

In diesem Abschnitt werden drei weitere Kategorien vorgestellt:

- Funktionen, die Möglichkeiten der Beziehung und des Vergleichs geografischer Merkmale festlegen

- Funktionen, die neue Geometrien generieren
- Funktionen, die die Werte einer Geometrie in ein Format umsetzen, das importiert oder exportiert werden kann

Funktionen, die Beziehungen oder Vergleiche zwischen geografischen Merkmalen zeigen

Verschiedene räumliche Funktionen geben Informationen zurück über die Beziehung oder den Vergleich zwischen verschiedenen geografischen Merkmalen. Die meisten dieser Funktionen geben einen ganzzahligen Wert zurück. In diesem Abschnitt werden die Prädikatfunktionen zunächst im Allgemeinen und dann im Einzelnen beschrieben.

Prädikatfunktionen

Prädikatfunktionen geben 1 (TRUE) zurück, wenn ein Vergleich die Kriterien der Funktion erfüllt oder 0 (FALSE), wenn der Vergleich fehlschlägt. Prädikate, die einen Test auf eine räumliche Beziehung durchführen, vergleichen Paare von Geometrien, die einen unterschiedlichen Typ oder unterschiedliche Dimensionen haben können.

Prädikate vergleichen die X- und Y-Koordinaten der übergebenen Geometrien. Die Z-Koordinaten und das Maß (sofern vorhanden) werden nicht berücksichtigt. Auf diese Weise können Geometrien, die Z-Koordinaten oder ein Maß haben, mit anderen Geometrien ohne diese Merkmale verglichen werden.

Das *Dimensionally Extended 9 Intersection Model (DE-9IM)*¹ ist ein mathematischer Ansatz, der die paarweise räumliche Beziehung zwischen Geometrien mit verschiedenen Typen und Dimensionen definiert. Dieses Modell drückt die räumliche Beziehung zwischen allen Typen von Geometrien als paarweise Schnittmengen ihrer Innenbereiche, Begrenzungen und Außenbereiche unter Berücksichtigung der Dimension der resultierenden Schnittmengen aus.

Die gegebenen Geometrien a und b : $I(a)$, $B(a)$ und $E(a)$ stellen Innenbereich, Begrenzung und Außenbereich von a dar. $I(b)$, $B(b)$ und $E(b)$ stellen den Innenbereich, die Begrenzung und den Außenbereich von b dar. Die Schnittmengen von $I(a)$, $B(a)$ und $E(a)$ mit $I(b)$, $B(b)$ und $E(b)$ ergeben eine 3-mal-3-Matrix. Jede Schnittmenge kann Geometrien verschiedener Dimensionen ergeben. Die Schnittmenge der Begrenzungen zweier Polygone besteht

1. Das DE-9IM wurde von Clementini und Felice entwickelt, die das 9 Intersection Model von Egenhofer und Herring dimensional erweiterten. DE-9IM ist das Ergebnis der Zusammenarbeit der vier Autoren Clementini, Eliseo, Di Felice und van Ostrom. Diese Autoren haben das Modell in "A Small Set of Formal Topological Relationships Suitable for End-User Interaction," D. Abel and B.C. Ooi (Ed.), *Advances in Spatial Database—Third International Symposium. SSD '93*. LNCS 692. Pp. 277-295, veröffentlicht. Das 9 Intersection Modell vom Springer-Verlag Singapore (1993) Egenhofer M.J. und Herring, J. wurde in "Categorizing binary topological relationships between regions, lines, and points in geographic databases," *Tech. Report, Department of Surveying Engineering, University of Maine, Orono, ME 1991*, veröffentlicht.

beispielsweise aus einem Punkt und einer Linienfolge; in diesem Fall gibt die dim-Funktion die maximale Dimension 1 zurück.

Die dim-Funktion gibt den Wert -1 , 0 , 1 oder 2 zurück. Die 1 entspricht einer Nullmenge oder $\text{dim}(\text{null})$; dieses Ergebnis wird zurückgegeben, wenn keine Schnittpunkte gefunden wurden.

	Innenbereich	Begrenzung	Außenbereich
Innenbereich	$\text{dim}(I(a) \cap I(b))$	$\text{dim}(I(a) \cap B(b))$	$\text{dim}(I(a) \cap E(b))$
Begrenzung	$\text{dim}(B(a) \cap I(b))$	$\text{dim}(B(a) \cap B(b))$	$\text{dim}(B(a) \cap E(b))$
Außenbereich	$\text{dim}(E(a) \cap I(b))$	$\text{dim}(E(a) \cap B(b))$	$\text{dim}(E(a) \cap E(b))$

Sie können das Ergebnis der räumlichen Beziehungsprädikate verstehen oder überprüfen, indem Sie die Ergebnisse des Prädikats mit einer Mustermatrix vergleichen, die die akzeptablen Werte des DE-9IM darstellt.

Die Mustermatrix enthält die akzeptablen Werte für alle Schnittmengen-Matrixzellen. Die möglichen Musterwerte sind:

- T** Es muss eine Schnittmenge vorhanden sein, $\text{dim} = 0, 1$ oder 2 .
- F** Es darf keine Schnittmenge vorhanden sein, $\text{dim} = -1$.
- *** Es spielt keine Rolle, ob eine Schnittmenge vorhanden ist, $\text{dim} = -1, 0, 1$ oder 2 .
- 0** Es muss eine Schnittmenge vorhanden sein, und ihre maximale Dimension muss 0 sein, $\text{dim} = 0$.
- 1** Es muss eine Schnittmenge vorhanden sein, und ihre maximale Dimension muss 1 sein, $\text{dim} = 1$.
- 2** Es muss eine Schnittmenge vorhanden sein, und ihre maximale Dimension muss 2 sein, $\text{dim} = 2$.

Die folgende Mustermatrix für das Prädikat `ST_Within` enthält beispielsweise die Werte T, F und *.

Tabelle 41. Matrix für ST_Within. Die Mustermatrix für das Prädikat `ST_Within` für Geometriekombinationen.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	F
	Begrenzung	*	*	F
	Außenbereich	*	*	*

Das Prädikat `ST_Within` gibt `TRUE` zurück, wenn sich die Innenbereiche der beiden Geometrien schneiden und wenn der Innenbereich und die Begrenzung von *a* sich nicht mit dem Außenbereich von *b* schneidet. Alle weitere Bedingungen sind nicht von Bedeutung.

Jedes Prädikat hat mindestens eine Mustermatrix; einige erfordern jedoch auch mehrere Matrizen, um die Beziehungen zwischen verschiedenen Kombinationen von Geometrietypen zu beschreiben.

ST_Equals

`ST_Equals` gibt 1 (`TRUE`) zurück, wenn zwei Geometrien des gleichen Typs identische Koordinaten X,Y haben.





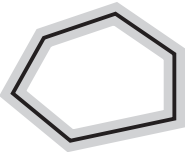
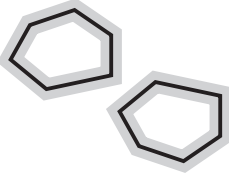
	
Punkt / Punkt	Mehrpunktangabe / Mehrpunktangabe
	
Linienfolge / Linienfolge	Mehrlinienfolge / Mehrlinienfolge
	
Polygon / Polygon	Multipolygon / Multipolygon

Abbildung 14. `ST_Equals`. Geometrien sind gleich, wenn sie übereinstimmende Koordinaten X,Y haben.

Tabelle 42. Matrix für Gleichheit. Die DE-9IM Mustermatrix für die Gleichheit stellt sicher, dass sich die Innenbereiche schneiden und dass kein Teil des Innenbereichs oder der Begrenzung den Außenbereich des anderen schneidet.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	F
	Begrenzung	*	*	F
	Außenbereich	F	F	*

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Equals“ auf Seite 249.

ST_OrderingEquals

ST_OrderingEquals vergleicht zwei Geometrien und gibt 1 (TRUE) zurück, wenn die Geometrien gleich sind und die Koordinaten die gleiche Reihenfolge haben; andernfalls wird 0 (FALSE) zurückgegeben. Weitere Informationen hierzu finden Sie im Abschnitt „ST_OrderingEquals“ auf Seite 291.

ST_Disjoint

ST_Disjoint gibt 1 (TRUE) zurück, wenn die Schnittmenge der beiden Geometrien eine leere Menge ist.

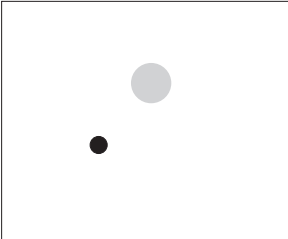
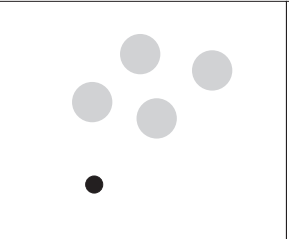
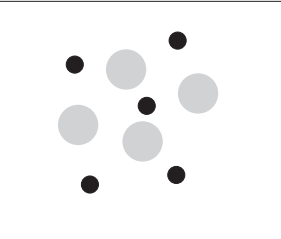
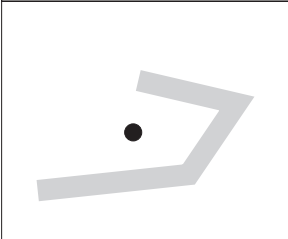
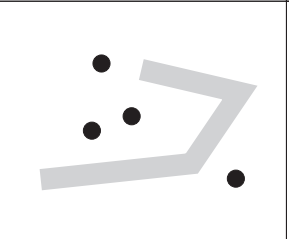
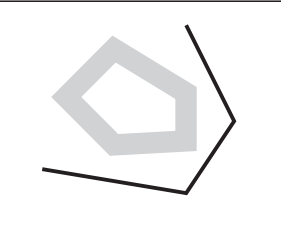
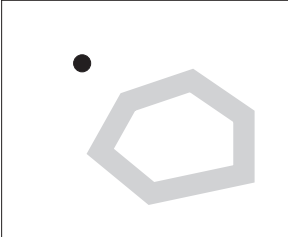
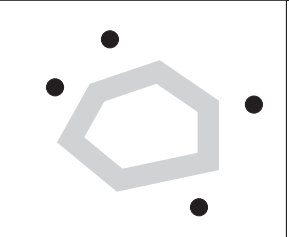
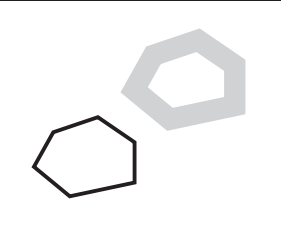
		
Punkt / Punkt	Punkt / Mehrpunktangabe	Mehrpunktangabe / Mehrpunktangabe
		
Punkt / Linienfolge	Mehrlinienfolge / Linienfolge	Polygon / Linienfolge
		
Punkt / Polygon	Mehrpunktangabe / Multipolygon	Polygon / Polygon

Abbildung 15. *ST_Disjoint*. Geometrien sind nicht verbunden, wenn sie einander nicht schneiden.

Tabelle 43. Matrix für *ST_Disjoint*. Die Mustermatrix des Prädikats *ST_Disjoint* gibt nur an, dass sich weder die Innenbereiche noch die Begrenzungen der Geometrien schneiden.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	F	F	*
	Begrenzung	F	F	*
	Außenbereich	*	*	*

Weitere Informationen hierzu finden Sie im Abschnitt „*ST_Disjoint*“ auf Seite 243.

ST_Intersects

ST_Intersects gibt 1 (TRUE) zurück, wenn die Schnittmenge keine leere Menge ist. ST_Intersects gibt genau das entgegengesetzte Ergebnis von ST_Disjoint zurück.

Das Prädikat ST_Intersects gibt TRUE zurück, wenn die Bedingung einer der folgenden Mustermatrizen TRUE zurückgibt.

Tabelle 44. Matrix für ST_Intersects (1). Das Prädikat ST_Intersects gibt TRUE zurück, wenn sich die Innenbereiche der beiden Geometrien schneiden.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	*
	Begrenzung	*	*	*
	Außenbereich	*	*	*

Tabelle 45. Matrix für ST_Intersects (2). Das Prädikat ST_Intersects gibt TRUE zurück, wenn die Begrenzung der ersten Geometrie die Begrenzung der zweiten Geometrie schneidet.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	*	T	*
	Begrenzung	*	*	*
	Außenbereich	*	*	*

Tabelle 46. Matrix für ST_Intersects (3). Das Prädikat ST_Intersects gibt TRUE zurück, wenn die Begrenzung der ersten Geometrie den Innenbereich der zweiten Geometrie schneidet.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	*	*	*
	Begrenzung	T	*	*
	Außenbereich	*	*	*

Tabelle 47. Matrix für ST_Intersects (4). Das Prädikat ST_Intersects gibt TRUE zurück, wenn sich die Begrenzungen der beiden Geometrien schneiden.

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	*	*	*
	Begrenzung	*	T	*
	Außenbereich	*	*	*

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Intersects“ auf Seite 266.

EnvelopesIntersect

Diese Funktion gibt 1 (TRUE) zurück, wenn sich die Umschläge der beiden Geometrien schneiden. Es ist eine hilfreiche Funktion, die `ST_Intersects` (`ST_Envelope(g1), ST_Envelope(g2)`) effizient implementiert. Weitere Informationen hierzu finden Sie im Abschnitt „EnvelopesIntersect“ auf Seite 201.

ST_Touches

`ST_Touches` gibt 1 (TRUE) zurück, wenn keiner der gemeinsamen Punkte der beiden Geometrien den Innenbereich beider Geometrien schneidet. Mindestens eine der Geometrien muss eine Linienfolge, ein Polygon, eine Mehrlinienfolge oder ein Multipolygon sein.

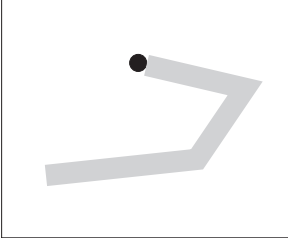
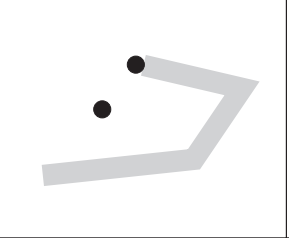
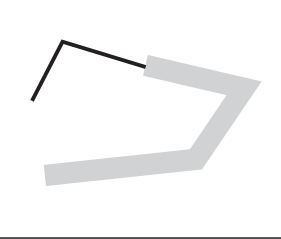
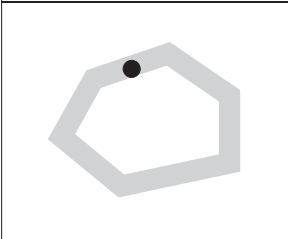
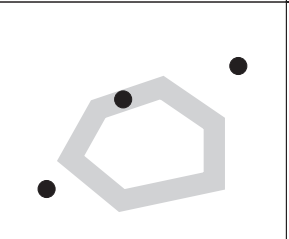
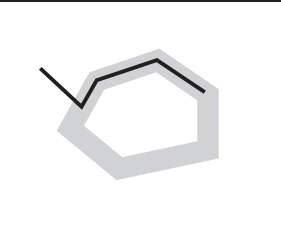
		
Punkt / Linienfolge	Mehrpunktangabe / Linienfolge	Linienfolge / Linienfolge
		
Punkt / Polygon	Mehrpunktangabe / Polygon	Linienfolge / Polygon

Abbildung 16. `ST_Touches`

Die Mustermatrizen zeigen, dass das Prädikat `ST_Touches` TRUE zurückgibt, wenn sich die Innenbereiche der Geometrie nicht schneiden und die Begrenzungen einer der beiden Geometrien den Innenbereich oder die Begrenzung der jeweils anderen Geometrie schneiden.

Tabelle 48. Matrix für `ST_Touches` (1)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	F	T	*
	Begrenzung	*	*	*
	Außenbereich	*	*	*

Tabelle 49. Matrix für ST_Touches (2)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	F	*	*
	Begrenzung	T	*	*
	Außenbereich	*	*	*

Tabelle 50. Matrix für ST_Touches (3)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	F	*	*
	Begrenzung	*	T	*
	Außenbereich	*	*	*

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Touches“ auf Seite 311.

ST_Overlaps

ST_Overlaps vergleicht zwei Geometrien der gleichen Dimension. Sie gibt 1 (TRUE) zurück, wenn die Schnittmenge eine Geometrie ergibt, die sich von beiden Geometrien unterscheidet, aber die gleiche Dimension hat.

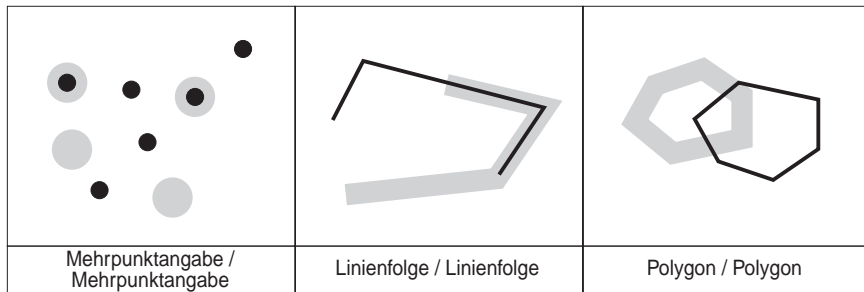


Abbildung 17. ST_Overlaps

Die Mustermatrix in Tabelle 51 auf Seite 181 gilt für die Überlagerungen Polygon/Polygon, Mehrpunktangabe/Mehrpunktangabe und Multipolygon/Multipolygon. Für diese Kombinationen gibt das Überlagerungsprädikat TRUE zurück, wenn der Innenbereich beider Geometrien den Innenbereich und Außenbereich der anderen Geometrie schneidet.

Tabelle 51. Matrix für ST_Overlaps (1)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	T
	Begrenzung	*	*	*
	Außenbereich	T	*	*

Die Mustermatrix in Tabelle 52 gilt für die Überlagerungen Linienfolge/Linienfolge und Mehrlinienfolge/Mehrlinienfolge. In diesem Fall muss die Schnittmenge der Geometrien eine Geometrie mit der Dimension 1 (andere Linienfolge) ergeben. Wenn die Dimension der Schnittmenge der Innenbereiche 1 ist, gibt das Prädikat ST_Overlaps FALSE zurück; das Prädikat ST_Crosses gibt jedoch TRUE zurück.

Tabelle 52. Matrix für ST_Overlaps (2)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	1	*	T
	Begrenzung	*	*	*
	Außenbereich	T	*	*

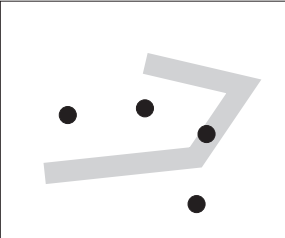



Weitere Informationen hierzu finden Sie im Abschnitt „ST_Overlaps“ auf Seite 292.

ST_Crosses

ST_Crosses verwendet zwei Geometrien und gibt 1 (TRUE) zurück, wenn folgende Bedingungen erfüllt sind:

- Die Schnittmenge ergibt eine Geometrie, deren Dimension niedriger ist als die maximale Dimension der Quellengeometrien.
- Die Schnittmenge liegt im Innenbereich beider Quellengeometrien.

ST_Crosses gibt 1 (TRUE) nur für Vergleiche Mehrpunktangabe/Polygon, Mehrpunktangabe/Linienfolge, Linienfolge/Linienfolge, Linienfolge/Polygon und Linienfolge/Multipolygon zurück.

	
Mehrpunktangabe / Linienfolge	Linienfolge / Linienfolge
	
Mehrpunktangabe / Polygon	Linienfolge / Polygon

Die Mustermatrix in Tabelle 53 gilt für Mehrpunktangabe/Linienfolge, Mehrpunktangabe/Mehrlinienfolge, Mehrpunktangabe/Polygon, Mehrpunktangabe/Multipolygon, Linienfolge/Polygon und Linienfolge/Multipolygon. Die Matrix gibt an, dass sich die Innenbereiche schneiden müssen und dass der Innenbereich der primären Geometrie (Geometrie *a*) den Außenbereich der sekundären Geometrie (Geometrie *b*) schneiden muss.

Tabelle 53. Matrix für *ST_Crosses* (1)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	T
	Begrenzung	*	*	*
	Außenbereich	*	*	*

Die Mustermatrix in Tabelle 54 auf Seite 183 gilt für die Überlagerungen Linienfolge/Linienfolge, Linienfolge/Mehrlinienfolge und Mehrlinienfolge/Mehrlinienfolge. Die Matrix gibt an, dass die Dimension der Schnittmenge der Innenbereiche 0 (Berührung in einem Punkt) sein muss. Wenn die Dimension dieser Schnittmenge 1 ist (Schnitt in einer Linienfolge), gibt das Prädikat *ST_Crosses* FALSE zurück; das Prädikat *ST_Overlaps* gibt jedoch TRUE zurück.

Tabelle 54. Matrix für ST_Crosses (2)

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	0	*	*
	Begrenzung	*	*	*
	Außenbereich	*	*	*

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Crosses“ auf Seite 238.

ST_Within

ST_Within gibt 1 (TRUE) zurück, wenn die Geometrie vollständig in der zweiten Geometrie enthalten ist. ST_Within gibt genau das entgegengesetzte Ergebnis von ST_Contains zurück.

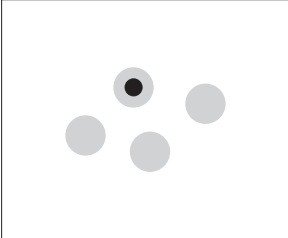
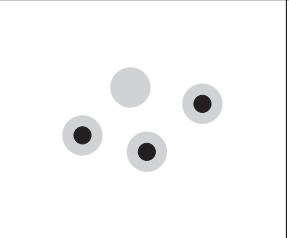
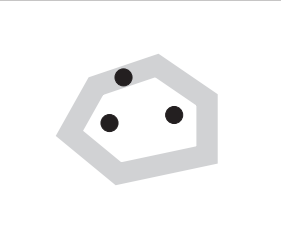
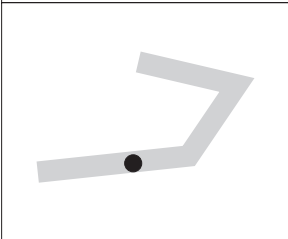
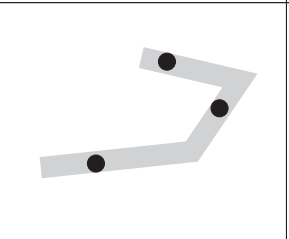
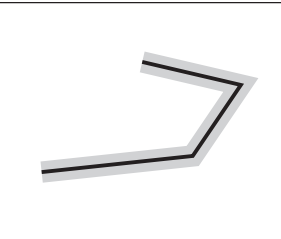
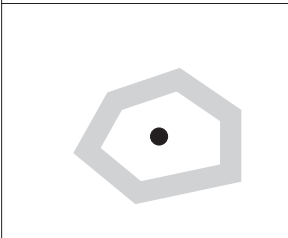
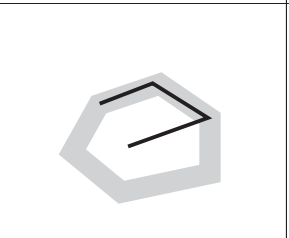
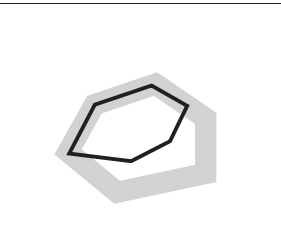
		
Punkt / Mehrpunktangabe	Mehrpunktangabe / Mehrpunktangabe	Mehrpunktangabe / Polygon
		
Punkt / Linienfolge	Mehrpunktangabe / Linienfolge	Linienfolge / Linienfolge
		
Punkt / Polygon	Linienfolge / Polygon	Polygon / Polygon

Abbildung 18. ST_Within

Die Mustermatrix von `ST_Within` gibt an, dass sich die Innenbereiche der beiden Geometrien schneiden müssen und dass der Innenbereich und die Begrenzung der primären Geometrie (Geometrie *a*) den Außenbereich der sekundären Geometrie (Geometrie *b*) nicht schneiden dürfen.

Tabelle 55. Matrix für ST_Within

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	F
	Begrenzung	*	*	F
	Außenbereich	*	*	*

Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Within`“ auf Seite 315.

ST_Contains

`ST_Contains` gibt 1 (TRUE) zurück, wenn die zweite Geometrie vollständig in der ersten Geometrie enthalten ist. `ST_Contains` gibt genau das entgegengesetzte Ergebnis von `ST_Within` zurück.

Mehrpunktangabe / Punkt	Mehrpunktangabe / Mehrpunktangabe	Polygon / Mehrpunktangabe
Linienfolge / Punkt	Linienfolge / Mehrpunktangabe	Linienfolge / Linienfolge
Polygon / Punkt	Polygon / Linienfolge	Polygon / Polygon

Abbildung 19. *ST_Contains*

Die Mustermatrix von *ST_Contains* gibt an, dass sich die Innenbereiche der beiden Geometrien schneiden müssen und dass der Innenbereich und die Begrenzung der zweiten Geometrie (Geometrie *b*) den Innenbereich der primären Geometrie (Geometrie *a*) nicht schneiden dürfen.

Tabelle 56. Matrix für *ST_Contains*

		b		
		Innenbereich	Begrenzung	Außenbereich
a	Innenbereich	T	*	*
	Begrenzung	*	*	*
	Außenbereich	F	F	*

Weitere Informationen hierzu finden Sie im Abschnitt „*ST_Contains*“ auf Seite 232.

ST_Relate

Die Funktion `ST_Relate` vergleicht zwei Geometrien und gibt 1 (TRUE) zurück, wenn die Geometrien die von der Mustermatrix-Zeichenfolge DE-91M definierten Bedingungen erfüllen; andernfalls wird 0 (FALSE) zurückgegeben. Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Relate`“ auf Seite 305.

ST_Distance

Die Funktion `ST_Distance` liefert die Mindestentfernung, die zwei nicht verbundene Merkmale trennt. Wenn die Merkmale nicht getrennt sind, meldet die Funktion den Mindestabstand 0.

`ST_Distance` könnte beispielsweise die kürzeste Entfernung angeben, mit der ein Flugzeug die Strecken zwischen zwei Standorten überbrücken könnte. Abb. 20 verdeutlicht diese Angabe.



Abbildung 20. Mindestentfernung zwischen zwei Städten. `ST_Distance` kann die Koordinaten der Standorte von Los Angeles und Chicago als Eingabe verwenden und einen Wert zurückgeben, der die Mindestentfernung zwischen diesen beiden Standorten angibt.

Weitere Informationen hierzu finden Sie im Abschnitt „`ST_Distance`“ auf Seite 245.

Funktionen, die neue Geometrien aus vorhandenen generieren

Der Spatial Extender bietet Prädikate und Umsetzungsfunktionen, die neue Geometrien aus vorhandenen generieren.

ST_Intersection

Die Funktion `ST_Intersection` gibt die Schnittmenge der beiden Geometrien zurück. Die Schnittmenge wird immer als Gruppe zurückgegeben, die die Mindestdimension der Quellgeometrien darstellt. Für eine Linienfolge, die ein Polygon schneidet, gibt die Funktion `ST_Intersection` beispielsweise eine Mehrlinienfolge zurück, die aus einem Abschnitt dieser Linienfolge besteht, der zum Innenbereich und zur Begrenzung des Polygons gehört. Die Mehrlinienfolge enthält mehrere Linienfolgen, wenn die Quellenlinienfolge das Polygon mit zwei oder mehr nicht zusammenhängenden Segmenten schnei-

det. Wenn sich die Geometrien nicht schneiden oder wenn die Schnittmenge eine Dimension ergibt, die kleiner ist als die der Quellgeometrien, wird eine leere Geometrie zurückgegeben.

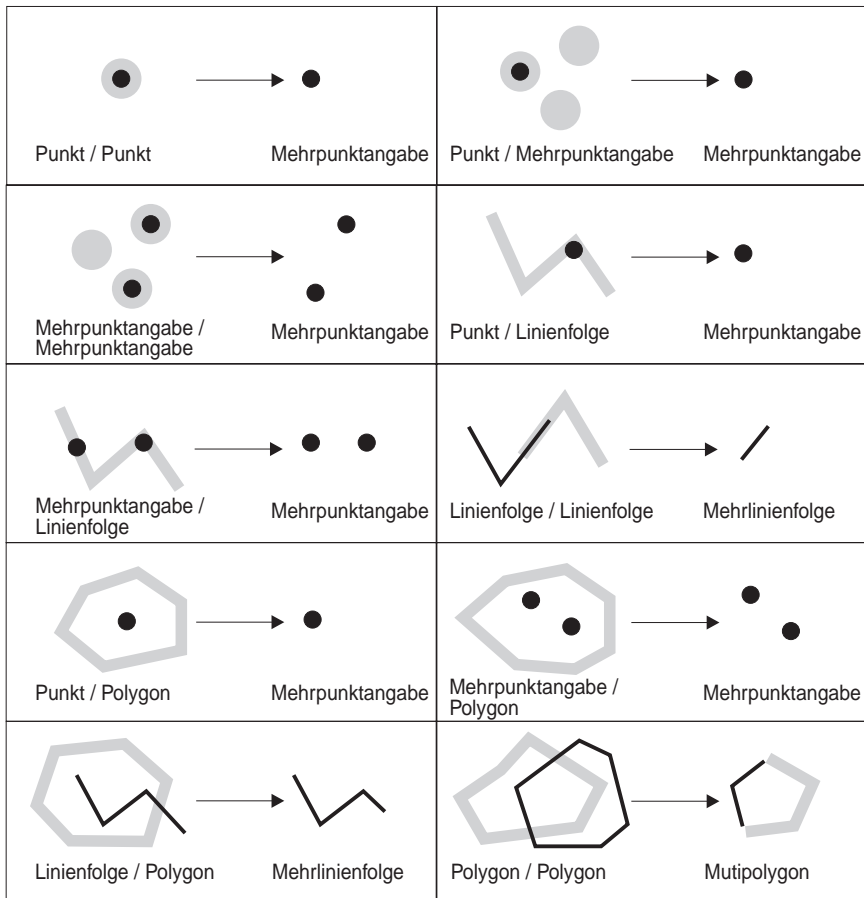


Abbildung 21. *ST_Intersection*. Beispiele zur Funktion *ST_Intersection*.

Weitere Informationen hierzu finden Sie im Abschnitt „*ST_Intersection*“ auf Seite 264.

ST_Difference

Die Eingabe für *ST_Difference* sind zwei Geometrien. Die erste wird als *primäre Geometrie*, die zweite als *sekundäre Geometrie* bezeichnet. Die Funktion *ST_Difference* gibt den Teil der primären Geometrie zurück, der nicht von der zweiten Geometrie geschnitten wird. Dies entspricht der logischen räumlichen Verknüpfung AND NOT. Die Funktion *ST_Difference* arbeitet nur mit Geometrien gleicher Dimensionen und gibt eine Gruppe zurück, die die gleiche Dimension hat wie die Quellgeometrien. Falls die Quellgeometrien gleich

sind, wird eine leere Geometrie zurückgegeben. Wenn die Dimensionen der Geometrien, die als Eingabe für ST_Difference verwendet werden, nicht übereinstimmen, gibt ST_Difference Null zurück.

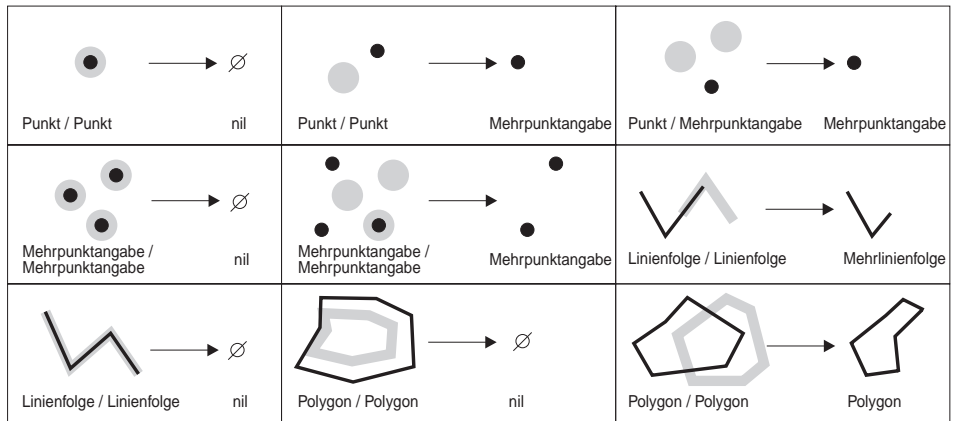


Abbildung 22. ST_Difference

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Difference“ auf Seite 240.

ST_Union

Die Funktion ST_Union gibt die Verknüpfungsmenge der beiden Geometrien zurück. Dies entspricht der logischen räumlichen Funktion OR. Die Quellengeometrien müssen die gleiche Dimension haben. ST_Union gibt das Ergebnis immer als Gruppe zurück.

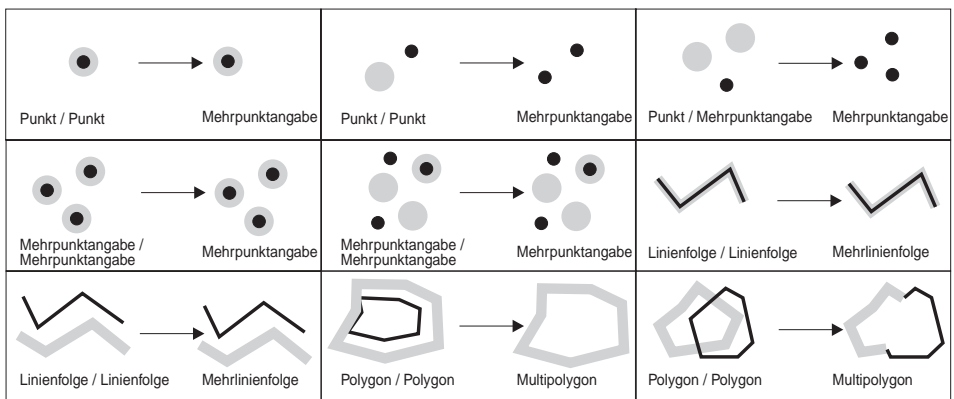


Abbildung 23. ST_Union

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Union“ auf Seite 314.

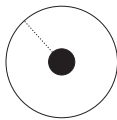
ST_SymmetricDiff

Die Funktion `ST_SymmetricDiff` gibt die symmetrische Differenz (das Boolesche logische XOR des Raums) zweier sich schneidender Geometrien zurück, die dieselbe Dimension aufweisen. Wenn diese Geometrien übereinstimmen, gibt `ST_SymmetricDiff` eine leere Geometrie zurück. Wenn sie nicht übereinstimmen, liegt ein Abschnitt einer oder beider Geometrien außerhalb der Schnittbereiche. `ST_SymmetricDiff` gibt den sich nicht schneidenden Abschnitt bzw. die sich nicht schneidenden Abschnitte als Gruppe, z. B. als Multipolygon zurück.

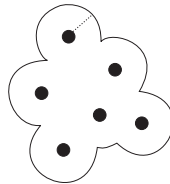
Wenn `ST_SymmetricDiff` Geometrien mit unterschiedlicher Dimension als Eingabe erhält, gibt die Funktion eine Null zurück.

ST_Buffer

Die Funktion `ST_Buffer` generiert eine Geometrie durch Einkreisen einer Geometrie mit einem angegebenen Abstand. Ein Polygon ergibt sich, wenn eine primäre Geometrie gepuffert wird oder wenn die Elemente einer Gruppe eng genug beieinander liegen, so dass sich alle Polygonpuffer überlappen. Wenn genügend Abstand zwischen den Elementen einer gepufferten Gruppe liegt, ergeben sich einzelne Pufferpolygone; in diesem Fall gibt die Funktion `ST_Buffer` ein Multipolygon zurück.



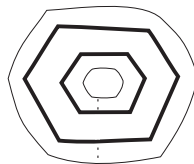
Punkt puffern



Mehrpunktangabe puffern



Linienfolge puffern



Polygon mit einem inneren Ring puffern

Abbildung 24. `ST_Buffer`

Die Funktion `ST_Buffer` verwendet positive und negative Abstände; für negative Puffer werden jedoch nur Geometrien mit der Dimension 2 (Polygone und Multipolygone) angewendet. Der absolute Wert des Pufferabstands

wird verwendet, wenn die Dimension der Quellengeometrie weniger als 2 (alle Geometrien außer Polygon oder Multipolygon) beträgt.

Im allgemeinen generieren positive Pufferabstände für äußere Ringe Polygonringe, die sich außerhalb der Mitte der Quellengeometrie befinden; negative Pufferabstände generieren Polygon- oder Multipolygonringe zur Mitte hin. Für innere Ringe eines Polygons oder Multipolygons generiert ein positiver Pufferabstand einen Pufferring zur Mitte hin, und ein negativer Pufferabstand generiert einen Pufferring außerhalb der Mitte.

Der Pufferungsprozess mischt überlappende Polygone. Negative Abstände größer als die Hälfte der maximalen Breite des Innenbereichs eines Polygons ergeben eine leere Geometrie.

Weitere Informationen hierzu finden Sie im Abschnitt „ST_Buffer“ auf Seite 229.

LocateAlong

Für Geometrien mit Maßen kann die Position eines bestimmten Maßes mit der Funktion LocateAlong ermittelt werden. LocateAlong gibt die Position als Mehrpunktangabe zurück. Wenn die Dimension der Quellengeometrie 0 ist (beispielsweise ein Punkt und eine Mehrpunktangabe), ist eine exakte Übereinstimmung erforderlich; Punkte, mit übereinstimmendem Maß werden als Mehrpunktangabe zurückgegeben. Für Quellengeometrien mit einer Dimension größer als Null wird die Position interpoliert. Wenn das Maß beispielsweise als 5.5 eingegeben wird und die Maße an den Scheitelpunkten einer Linienfolge 3, 4, 5, 6 und 7 sind, wird der interpolierte Punkt, der exakt zwischen den Scheitelpunkten 5 und 6 liegt, zurückgegeben.

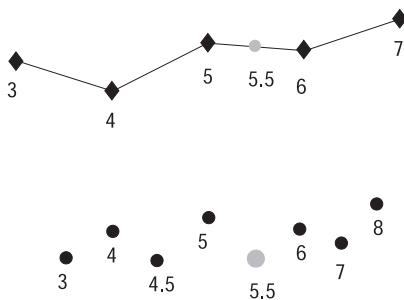


Abbildung 25. LocateAlong

Weitere Informationen hierzu finden Sie im Abschnitt „LocateAlong“ auf Seite 209.

LocateBetween

Die Funktion `LocateBetween` gibt entweder die Gruppe der Pfade oder Positionen zurück zwischen zwei Maßen einer Quellengeometrie mit Maßen. Wenn die Dimension der Quellengeometrie 0 ist, gibt `LocateBetween` eine Mehrpunktangabe mit allen Punkten zurück, deren Maß zwischen den beiden Quellenmaßen liegt. Für Quellengeometrien mit einer Dimension größer als 0 gibt `LocateBetween` eine Mehrlinienfolge zurück, wenn ein Pfad interpoliert werden kann; andernfalls gibt `LocateBetween` eine Mehrpunktangabe mit den Punktpositionen zurück. Ein leerer Punkt wird zurückgegeben, wenn `LocateBetween` keinen Pfad interpolieren kann oder keine Position zwischen den Maßen gefunden wird. `LocateBetween` führt eine inklusive Suche der Geometrien durch; die Maße der Geometrien müssen daher größer oder gleich dem Maß *from* und kleiner oder gleich dem Maß *to* sein.

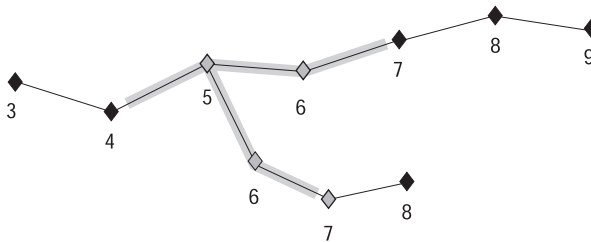


Abbildung 26. `LocateBetween`

Weitere Informationen hierzu finden Sie im Abschnitt „`LocateBetween`“ auf Seite 211.

ST_ConvexHull

Die Funktion `ST_ConvexHull` gibt ein konvexes Hülsenpolygon einer Geometrie zurück, die mindestens drei Scheitelpunkte haben muss, die eine konvexe Form bilden. Wenn die Scheitelpunkte der Geometrie keine konvexe Form bilden, gibt `ST_ConvexHull` Null zurück. `ST_ConvexHull` ist häufig der erste Schritt bei der Mosaikbildung zum Erstellen eines TIN-Netzwerks aus einer Reihe von Punkten.

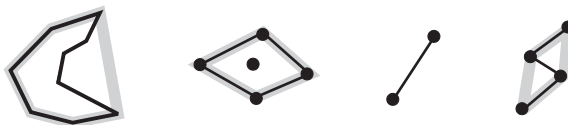


Abbildung 27. `ST_ConvexHull`

Weitere Informationen hierzu finden Sie im Abschnitt „`ST_ConvexHull`“ auf Seite 234.

ST_Polygon

Generiert ein Polygon aus einer Linienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „ST_Polygon“ auf Seite 304.

Funktionen, die das Format der Werte einer Geometrie umsetzen

Der Spatial Extender unterstützt drei GIS-Datenaustauschformate:

- Bekannte Textdarstellung
- Bekannte Binärdarstellung
- ESRI-Formdarstellung

Bekannte Textdarstellung

Der Spatial Extender hat verschiedene Funktionen, die Geometrien aus Textbeschreibungen generieren.

ST_WKTToSQL

Erstellt eine Geometrie aus einer Textdarstellung eines beliebigen Geometrietyps. Es darf keine Kennung für ein räumliches Bezugssystem angegeben werden. Weitere Informationen hierzu finden Sie im Abschnitt „ST_WKTToSQL“ auf Seite 318.

ST_GeomFromText

Erstellt eine Geometrie aus einer Textdarstellung eines beliebigen Geometrietyps. Es muss eine Kennung für ein räumliches Bezugssystem angegeben werden. Weitere Informationen hierzu finden Sie im Abschnitt „ST_GeomFromText“ auf Seite 255.

ST_PointFromText

Erstellt einen Punkt aus einer aus der Textdarstellung eines Punkts. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointFromText“ auf Seite 296.

ST_LineFromText

Erstellt eine Linienfolge aus der Textdarstellung einer Linienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „ST_LineFromText“ auf Seite 278.

ST_PolyFromText

Erstellt ein Polygon aus der Textdarstellung eines Polygons. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PolyFromText“ auf Seite 301.

ST_MPointFromText

Erstellt eine Mehrpunktangabe aus der Textdarstellung einer Mehrpunktangabe. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MPointFromText“ auf Seite 284.

ST_MLineFromText

Erstellt eine Mehrlinienfolge aus der Darstellung einer Mehrlinienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MLineFromText“ auf Seite 281.

ST_MPolyFromText

Erstellt ein Multipolygon aus der Darstellung eines Multipolygons. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MPolyFromText“ auf Seite 286.

Die Textdarstellung ist eine ASCII-Zeichenfolge. Sie ermöglicht den Austausch der Geometrie in ASCII-Textform. Diese Funktionen erfordern keine Definition spezieller Programmstrukturen für die Zuordnung einer binären Darstellung. Diese Funktionen können daher in einem 3GL- oder 4GL-Programm verwendet werden.

Die Funktion ST_AsText setzt eine vorhandene Geometrie um in eine Textdarstellung. Weitere Informationen hierzu finden Sie im Abschnitt „ST_AsText“ auf Seite 226.

Eine ausführliche Beschreibung der bekannten Textdarstellungen finden Sie im Abschnitt „Die bekannten OGC-Textdarstellungen“ auf Seite 335.

Bekannte Binärdarstellung

Der Spatial Extender hat verschiedene Funktionen, die Geometrien aus bekannten binären Darstellungen (WKB-Darstellungen) generieren.

ST_WKBToSQL

Erstellt eine Geometrie aus einer bekannten Binärdarstellung eines beliebigen Geometrietyps. Es darf keine Kennung für ein räumliches Bezugssystem angegeben werden. Weitere Informationen hierzu finden Sie im Abschnitt „ST_WKBToSQL“ auf Seite 316.

ST_GeomFromWKB

Erstellt eine Geometrie aus einer bekannten Binärdarstellung eines beliebigen Geometrietyps. Es muss eine Kennung für ein räumliches Bezugssystem angegeben werden. Weitere Informationen hierzu finden Sie im Abschnitt „ST_GeomFromWKB“ auf Seite 257.

ST_PointFromWKB

Erstellt einen Punkt aus einer bekannten Binärdarstellung eines beliebigen Punkts. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PointFromWKB“ auf Seite 297.

ST_LineFromWKB

Erstellt eine Linienfolge aus einer bekannten Binärdarstellung einer Linienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „ST_LineFromWKB“ auf Seite 279.

ST_PolyFromWKB

Erstellt ein Polygon aus einer bekannten Binärdarstellung eines Polygons. Weitere Informationen hierzu finden Sie im Abschnitt „ST_PolyFromWKB“ auf Seite 302.

ST_MPointFromWKB

Erstellt eine Mehrpunktangabe aus einer bekannten Binärdarstellung einer Mehrpunktangabe. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MPointFromWKB“ auf Seite 285.

ST_MLineFromWKB

Erstellt eine Mehrlinienfolge aus einer bekannten Binärdarstellung einer Mehrlinienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MLineFromWKB“ auf Seite 282.

ST_MPolyFromWKB

Erstellt ein Multipolygon aus einer bekannten Binärdarstellung eines Multipolygons. Weitere Informationen hierzu finden Sie im Abschnitt „ST_MPolyFromWKB“ auf Seite 287.

Die bekannte Binärdarstellung ist ein fortlaufender Byte-Datenstrom. Sie ermöglicht den Austausch der Geometrie zwischen einem ODBC-Client und einer SQL-Datenbank in binärer Form. Diese Geometriefunktionen erfordern die Definition von C-Strukturen für die Zuordnung der binären Darstellung. Sie sind somit für die Verwendung in einem 3GL-Programm konzipiert und eignen sich nicht für 4GL-Umgebungen.

Die Funktion ST_AsBinary setzt eine vorhandene Geometrie um in eine Binärdarstellung. Weitere Informationen hierzu finden Sie im Abschnitt „ST_AsBinary“ auf Seite 225.

Eine ausführliche Beschreibung der bekannten Binärdarstellungen finden Sie im Abschnitt „Die bekannten OGC-Binärdarstellungen (WKB)“ auf Seite 340.

ESRI-Formdarstellung

Der Spatial Extender hat verschiedene Funktionen, die Geometrien aus ESRI-Formdarstellungen generieren. Die ESRI-Formdarstellung unterstützt zusätzlich zu den von Text- und Binärdarstellungen unterstützten zweidimensionalen Darstellungen auch Z-Koordinaten und Maße.

ShapeToSQL

Erstellt eine Geometrie aus einer Form eines beliebigen Geometrietyps. Es darf keine Kennung für ein räumliches Bezugssystem angegeben werden. Weitere Informationen hierzu finden Sie im Abschnitt „ShapeToSQL“ auf Seite 221.

GeometryFromShape

Erstellt eine Geometrie aus einer Form eines beliebigen Geometrietyps. Es muss eine Kennung für ein räumliches Bezugssystem angege-

ben werden. Weitere Informationen hierzu finden Sie im Abschnitt „GeometryFromShape“ auf Seite 203.

PointFromShape

Erstellt einen Punkt aus einer Punktform. Weitere Informationen hierzu finden Sie im Abschnitt „PointFromShape“ auf Seite 218.

LineFromShape

Erstellt eine Linienfolge aus einer Mehrlinienform. Weitere Informationen hierzu finden Sie im Abschnitt „LineFromShape“ auf Seite 207.

PolyFromShape

Erstellt ein Polygon aus einer Mehrlinienform. Weitere Informationen hierzu finden Sie im Abschnitt „PolyFromShape“ auf Seite 219.

MPointFromShape

Erstellt eine Mehrpunktangabe aus einer Formdarstellung einer Mehrpunktangabe. Weitere Informationen hierzu finden Sie im Abschnitt „MPointFromShape“ auf Seite 216.

MLineFromShape

Erstellt eine Mehrlinienfolge aus der Form einer Mehrlinienfolge. Weitere Informationen hierzu finden Sie im Abschnitt „MLineFromShape“ auf Seite 214.

MPolyFromShape

Erstellt ein Multipolygon aus der Form eines Polygons aus mehreren Teilen. Weitere Informationen hierzu finden Sie im Abschnitt „MPolyFromShape“ auf Seite 217.

Die allgemeine Syntax dieser Funktionen ist gleich. Das erste Argument ist die Formdarstellung, die als BLOB-Datentyp eingegeben wird. Das zweite Argument ist die Kennung des räumlichen Bezugs, die der Geometrie zugeordnet werden soll. Die Funktion GeometryFromShape hat beispielsweise die folgende Syntax:

```
GeometryFromShape(shapegeometry, SRID)
```

Zur Zuordnung der Binärdarstellung erfordern diese Formfunktionen die Definition einer C-Struktur. Sie sind somit für die Verwendung in einem 3GL-Programm konzipiert und eignen sich nicht für 4GL-Umgebungen.

Die Funktion AsBinary setzt einen Geometriewert in eine ESRI-Formdarstellung um. Weitere Informationen hierzu finden Sie im Abschnitt „AsShape“ auf Seite 200.

Eine ausführliche Beschreibung der Formdarstellungen finden Sie im Abschnitt „Die ESRI-Formdarstellungen“ auf Seite 344.

Kapitel 14. Räumliche Funktionen für SQL-Abfragen

Dieses Kapitel enthält eine Liste der zur Abfrage räumlicher Daten verfügbaren Funktionen. Jede Funktion wird in einem Abschnitt mit Syntax, Rückgabetyt und Codebeispielen beschrieben. Einige der Beispiele in diesem Kapitel enthalten eine Anweisung `CREATE TABLE`, in der eine oder mehrere Spalten als räumliche Spalten definiert sind.

Wie unter "Informationen zu Typen von räumlichen Daten" in Kapitel 4 ("Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren") beschrieben wurde, bilden die räumlichen Datentypen eine Hierarchie mit `ST_Geometry` als Root-Knoten. Wenn der Text besagt, dass ein Wert eines untergeordneten Typs in dieser Hierarchie als Eingabe für eine Funktion verwendet werden kann, sollte der Leser wissen, dass alternativ ein Wert eines beliebigen Typs, der diesem übergeordneten Typ untergeordnet ist, als Eingabe verwendet werden kann.

In Kapitel 14 heißt es z. B. im Abschnitt "Is3d", dass die Eingabe für die Funktion `Is3d` ein Wert des Datentyps `ST_Geometry` ist. Der Leser sollte jedoch wissen, dass als Eingabe für `Is3d` alternativ ein Wert eines beliebigen unterstützten Datentyps verwendet werden kann, der `ST_Geometry` untergeordnet ist: `ST_Point`, `ST_Curve`, `ST_LineString` usw. Eine Zusammenstellung aller unterstützten untergeordneten Typen enthält Abbildung 6 im *Benutzer- und Referenzhandbuch*.

Folgende Überlegungen sind für räumliche Spalten relevant:

- Die Beispiele in diesem Kapitel sind mit dem Bibliotheksnamen `db2gse` angegeben. Statt jede räumliche Funktion und jeden Typ mit `db2gse` explizit anzugeben, können Sie den Funktionspfad auch so einstellen, dass er `db2gse` enthält.
- Wenn die Eingabe für `ST_Union` zwei Punkte mit identischen Koordinaten sind, gibt die Funktion einen Punkt zurück. Wenn dieser Punkt dann als Eingabe für `ST_IsSimple` verwendet wird, gibt `ST_IsSimple` 1 (`TRUE`) zurück, was so viel bedeutet wie: ja, der Punkt ist einfach).
- Wenn eine Null als Eingabeparameter für eine räumliche Funktion verwendet wird, gibt die Funktion eine Null als Ausgabeparameter zurück.
- Bevor Sie Daten in eine räumliche Spalte eingeben können:
 - Eventuell müssen Sie den Wert für den Parameter `udf_mem_sz` vergrößern. Der vorgeschlagene Anfangswert lautet 2048. Wenn 2048 nicht geeignet ist, vergrößern Sie den Parameter `udf_mem_sz` in Schritten zu 256.

- Sie müssen diesen Parameter als Schicht registrieren. Weitere Informationen zum Registrieren einer räumlichen Spalte als Schicht finden Sie in „Kapitel 4. Räumliche Spalten definieren, als Schichten registrieren und Geocodierer aktivieren“ auf Seite 45.

Verschachtelung

Mehrere verschachtelte Geometriefunktionen können ein Problem darstellen, wenn die innere verschachtelte Funktion einen Geometrietyt zurückgibt, der von der äußeren aufrufenden Funktion nicht akzeptiert wird. Dieses Problem lässt sich beheben, wenn der Geometrietyt der inneren verschachtelten Funktion in einen Typ umgewandelt werden kann, der von der äußeren aufrufenden Funktion akzeptiert wird.

ST_Geometry in einen Subtyp umwandeln

Der Geometrietyt von Funktionen, die den übergeordneten Typ ST_Geometry zurückgeben, kann mit der Funktion Treat in einen Subtyp umgewandelt werden. ST_Geometry-Werte werden z. B. von der Funktion ST_Union zurückgegeben. Wenn ST_Union innerhalb der Funktion ST_PointOnSurface verschachtelt ist, gibt ST_PointOnSurface den folgenden Fehler zurück:

```
SQL00440N No function by the name "ST_POINTONSURFACE"
having compatible arguments was found in the function
path.      SQLSTATE=42884
```

Die Funktion ST_PointOnSurface akzeptiert die Geometrietypen ST_Polygon und ST_MultiPolygon, nicht jedoch den von ST_Union zurückgegebenen Typ ST_Geometry, und zwar auch dann nicht, wenn die Funktion ST_Union den Wert ST_MultiPolygon zurückgibt. In diesem Fall muss daher der Geometrietyt der Funktion ST_Union in ST_MultiPolygon umgewandelt werden.

Bei einer Selbstverknüpfung der Tabelle COUNTIES durch die Vereinigung ihrer Polygonspalte COUNTY müsste die Funktion Treat auf das Ergebnis der Funktion ST_Union angewandt werden, um den Typ ST_Geometry in den Typ ST_MultiPolygon umzuwandeln. Erst dann könnte die Funktion ST_PointOnSurface angewandt werden.

```
SELECT ST_Astext(ST_PointOnSurface(
    TREAT ( ST_Union(c1.county, c2.county) AS ST_MultiPolygon)))
FROM counties AS c1, counties AS c2;
```

Wenn die Funktion ST_Union den Wert ST_MultiPolygon zurückgibt, wandelt die Funktion Treat diesen Wert in den Datentyp ST_MultiPolygon um. Wenn die Funktion ST_Union keinen ST_MultiPolygon-Wert zurückgibt, gibt die Funktion Treat einen Laufzeitfehler zurück.

Weitere Informationen über die Vorgehensweise bei Subtypen enthält das Handbuch *SQL Reference*.

Eine Gruppe in eine Basisgeometrie umwandeln

Mit der Funktion `ST_GeometryN` können Sie ein Element einer Geometrie-gruppe in eine Basisgeometrie umwandeln, die von der äußeren aufrufenden Funktion benötigt wird.

Der von der Funktion `ST_Union` zurückgegebene Wert z. B. ist immer eine als `ST_Geometry` zurückgegebene Geometrie-gruppe. Mit der Funktion `Treat` können Sie den Typ `ST_Geometry` in einen der folgenden Subtypen umwandeln: `ST_MultiPoint`, `ST_MultiLineString`, `ST_MultiPolygon`, `ST_GeomCollection`, `ST_MultiCurve` oder `ST_MultiSurface`. Durch Anwenden der Funktion `ST_GeometryN` auf die Ausgabe der Funktion `Treat` wandeln Sie die Geometrie-gruppe in eine Basisgeometrie um.

Zur Anwendung der Funktion `ST_ExteriorRing` auf die Ergebnisse der Funktion `ST_Union` aus dem Beispiel im vorhergehenden Abschnitt z. B. extrahieren Sie zunächst anhand der Funktion `ST_GeometryN` ein Polygonelement.

```
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;
```

Der Umsetzungsausdruck ist nur beim Übergang von einem übergeordneten zu einem untergeordneten Typ der Hierarchie erforderlich. Weitere Informationen über die Vorgehensweise bei Subtypen enthält das Handbuch *SQL Reference*.

AsShape

AsShape verwendet ein Geometrieobjekt und gibt ein BLOB zurück.

Syntax

```
db2gse.AsShape(g db2gse.ST_Geometry)
```

Rückgabetyt

BLOB(1m)

Beispiele

Der folgende Codeausschnitt zeigt, wie die Funktion AsShape die Zonenpolygone der Tabelle SENSITIVE_AREAS in Formpolygone umwandelt. Diese Formpolygone werden an die Funktion draw_polygon zur Anzeige übergeben.

```
/* Create the SQL expression. */
strcpy(sqlstmt, "select db2gse.AsShape (zone) from SENSITIVE_AREAS
where db2gse.EnvelopesIntersect(zone, db2gse.PolyFromShape(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Prepare the SQL statement. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* Set the pcbvalue1 length of the shape. */
pcbvalue1 = blob_len;

/* Bind the shape parameter */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* Execute the query */
rc = SQLExecute (hstmt);

/* Assign the results of the query (the Zone polygons) to the
  fetched_binary variable. */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* Fetch each polygon within the display window and display it. */

while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
  draw_polygon(fetched_binary);
```

EnvelopesIntersect

EnvelopesIntersect gibt 1 (TRUE) zurück, wenn sich die Umrisse von zwei Geometrien schneiden; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.EnvelopesIntersect(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Die Funktion `get_window` ruft die Koordinaten des Anzeigefensters aus der Anwendung ab. Der Fensterparameter ist tatsächlich eine Struktur in Form eines Polygons mit einer Zeichenfolge von Koordinaten, die das Anzeigepolygon darstellen. Die Funktion `PolyFromShape` wandelt die Form des Anzeigefensters in ein Spatial Extender-Polygon um, das die Funktion `EnvelopesIntersect` als ihren Schnittumriss verwendet. Alle Zonenpolygone `SENSITIVE_AREAS`, die den inneren Bereich oder die Begrenzung des Anzeigefensters schneiden, werden zurückgegeben. Jedes Polygon wird aus dem Ergebnis abgerufen und an die Funktion `draw_polygon` weitergegeben.

```
/* Get the display window coordinates as a polygon shape.
get_window(&window)

/* Create the SQL expression. The db2gse.EnvelopesIntersect function
   will be used to limit the result set to only those zone polygons
   that intersect the envelope of the display window. */
strcpy(sqlstmt, "select db2gse.AsShape(zone) from SENSITIVE_AREAS where
db2gse.EnvelopesIntersect (zone, db2gse.PolyFromShape(cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Set blob_len to the byte length of a 5 point shape polygon. */
blob_len = 128;

/* Prepare the SQL statement. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* Set the pcbvalue1 to the window shape */
pcbvalue1 = blob_len;

/* Bind the shape parameter */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB,
blob_len,0, window, blob_len, &pcbvalue1);

/* Execute the query */
rc = SQLExecute (hstmt);
```

```
/* Assign the results of the query, (the Zone polygons) to the
   fetched_binary variable. */
SQLBindCol(hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* Fetch each polygon within the display window and display it. */
while(SQL_SUCCESS == (rc = SQLFetch(hstmt))
      draw_polygon(fetched_binary);
```

GeometryFromShape

GeometryFromShape verwendet eine Form und die Kennung eines räumlichen Bezugssystems und gibt ein Geometrieobjekt zurück.

Syntax

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), SRID db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Der folgende C-Codeausschnitt enthält ODBC-Funktionen, die in SQL-Funktionen des Spatial Extender eingebettet sind; diese Funktionen fügen Daten in die Tabelle LOTS ein.

Die Tabelle LOTS wurde mit zwei Spalten erstellt: der Spalte LOT_ID, die jedes Grundstück eindeutig kennzeichnet, und der Polygonspalte LOT, die die Geometrie jedes Grundstücks enthält.

```
CREATE TABLE LOTS ( lot_id integer,
                    lot      db2gse.ST_Polygon);
```

Die Funktion GeometryFromShape wandelt Formen in eine Spatial Extender-Geometrie um. Die gesamte Anweisung INSERT wird in shp_sql kopiert. Die Anweisung INSERT enthält Parametermarken zum dynamischen Akzeptieren der LOT_ID-Daten und der LOT-Daten.

```
/* Create the SQL insert statement to populate the lot_id and the
   lot columns. The question marks are parameter markers that
   indicate the lot_id and lot values that will be retrieved at
   runtime. */
strcpy (shp_sql, "insert into LOTS (lot_id, lot) values (?,
db2gse.GeometryFromShape (cast(? as blob(1m)), db2gse.coordref()..srid(0))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Bind the integer key value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Bind the shape to the second parameter. */
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

Is3d

Is3d verwendet ein Geometrieobjekt und gibt 1 (TRUE) zurück, wenn das Objekt 3D-Koordinaten hat; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.Is3d(g db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle THREEED_TEST, die zwei Spalten enthält: die Spalte GID des Typs Integer und die Spalte G1.

```
CREATE TABLE THREEED_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Die Anweisung INSERT fügt zwei Punkte in die Tabelle THREEED_TEST ein. Der erste Punkt enthält keine Z-Koordinaten, der zweite dagegen schon.

```
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)))
```

Die folgende Anweisung SELECT listet den Inhalt der Spalte GID mit den Ergebnissen der Funktion Is3d auf. Die Funktion gibt 0 für die erste Zeile zurück; dieser Wert enthält keine Z-Koordinaten. Für die zweite Zeile wird 1 zurückgegeben; dieser Wert enthält Z-Koordinaten.

```
SELECT gid, db2gse.Is3d (g1) "Ist es 3d?" FROM THREEED_TEST
```

Das folgende Ergebnis wird zurückgegeben.

gid	Ist es 3d?
1	0
2	1

IsMeasured

IsMeasured verwendet ein Geometrieobjekt und gibt 1 (TRUE) zurück, wenn das Objekt Maße hat; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.IsMeasured(g db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle MEASURE_TEST mit zwei Spalten. Die GID-Spalte kennzeichnet die Zeilen eindeutig, und die Spalte G1 speichert die Punktgeometrien.

```
CREATE TABLE MEASURE_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen zwei Datensätze in die Tabelle MEASURE_TEST ein. Der erste Datensatz speichert einen Punkt, der kein Maß hat. Der Punkt des zweiten Datensatzes hat ein Maß.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen die GID-Spalte zusammen mit den Ergebnissen der Funktion IsMeasured. Die Funktion IsMeasured gibt 0 für die erste Zeile zurück, da der Punkt kein Maß hat. Sie gibt 1 für die zweite Zeile zurück, da der Punkt Maßangaben hat.

```
SELECT gid, db2gse.IsMeasured (g1) "Has measures?" FROM MEASURE_TEST
```

gid	Has measures
1	0
2	1

LineFromShape

LineFromShape verwendet eine Form des Typs Point und die Kennung eines räumlichen Bezugssystems und gibt eine Linienfolge zurück.

Syntax

```
db2gse.Line FromShape(ShapeLineString Blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_LineString
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle SEWERLINES mit der eindeutigen ID, der Größenklasse und der Geometrie aller Abwasserkanallinien auf.

Die Anweisung CREATE TABLE erstellt die Tabelle SEWERLINES mit drei Spalten. Die erste Spalte, SEWER_ID, kennzeichnet jede Abwasserkanallinie eindeutig. Die zweite Spalte CLASS des Typs Integer kennzeichnet den Typ des Abwasserkanals; normalerweise gibt diese Spalte die Kapazität des Kanals an. Die dritte Spalte, SEWER, des Typs Linienfolge speichert die Geometrie der Abwasserkanallinie.

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer,
                        sewer db2gse.ST_LineString);

/* Create the SQL insert statement to populate the sewer_id, size class and
   the sewer linestring. The question marks are parameter markers that
   indicate the sewer_id, class and sewer geometry values that will be
   retrieved at runtime. */
strcpy (shp_sql,"insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.Line FromShape (cast(? as blob(1m)),
db2gse.coordref()?.srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Bind the integer key value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* Bind the integer class value to the second parameter. */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);;
```

```
/* Bind the shape to the third parameter. */  
pcbvalue3 = blob_len;  
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, sewer_shape, blob_len, &pcbvalue3);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

LocateAlong

LocateAlong verwendet ein Geometrieobjekt und ein Maß und gibt die in diesem Maß gefundenen Punkte als Mehrpunktangabe (Multipoint) zurück.

Wenn LocateAlong eine Mehrpunktangabe und ein Maß als Eingabe erhält und dieses Maß nicht in der Mehrpunktangabe enthalten ist, gibt LocateAlong POINT EMPTY zurück.

Syntax

```
db2gse.LocateAlong(g db2gse.ST_Geometry, [measure] Double)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle LOCATEALONG_TEST. LOCATEALONG_TEST enthält zwei Spalten: die Spalte GID, die jede Zeile eindeutig kennzeichnet, und die Spalte G1, in der die Mustergeometrie gespeichert wird.

```
CREATE TABLE LOCATEALONG_TEST (gid integer, g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen zwei Zeilen ein. Die erste ist eine Mehrlinienfolge (Multilinestring), die zweite eine Mehrpunktangabe (Multipoint).

```
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)))
```

In der folgenden Anweisung SELECT und der entsprechenden Ergebnisgruppe wird die Funktion LocateAlong angewiesen, die Punkte mit dem Maß 6.5 zu finden. Die erste Zeile gibt eine Mehrpunktangabe (Multipoint) mit zwei Punkten zurück. Die zweite Zeile gibt dagegen einen leeren Punkt zurück. Für lineare Funktionen (Geometrie mit einer Abmessung größer als 0) kann LocateAlong den Punkt interpolieren; bei Mehrpunktangaben (Multipoint) muss das Zielmaß jedoch exakt übereinstimmen.

```

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.

In der folgenden Anweisung SELECT und der entsprechenden Ergebnisgruppe gibt die Funktion LocateAlong für beide Punkte Mehrpunktangaben (Multipoint) zurück. Das Zielmaß von 7 entspricht den Maßen in den Mehrlinien- und Mehrpunkt-Quellendaten.

```

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

LocateBetween verwendet ein Geometrieobjekt und zwei Mess-Standorte und gibt eine Geometrie zurück, die die Gruppe der unterbrochenen Pfade zwischen den beiden Mess-Standorten darstellt.

Syntax

```
db2gse.LocateBetween(g db2gse.ST_Geometry, [measure] Double, [measure]5 Double)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle LOCATEBETWEEN_TEST. LOCATEBETWEEN_TEST enthält zwei Spalten: die Spalte GID, die jede Zeile eindeutig kennzeichnet, und die Mehrlinienspalte G1, in der die Mustergeometrie gespeichert wird.

```
CREATE TABLE LOCATEBETWEEN_TEST (gid integer, g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen zwei Zeilen in die Tabelle LOCATEBETWEEN_TEST ein. Die erste Zeile ist eine Mehrlinienfolge (Multilinestring), die zweite eine Mehrpunktangabe (Multipoint).

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen, wie die Funktion LocateBetween Maße zwischen den Angaben 6.5 und 7.5 (je einschließlich) findet. Die erste Zeile gibt eine Mehrlinienfolge mit mehreren Linienfolgen (Linestrings) zurück. Die zweite Zeile gibt eine Mehrpunktangabe (Multipoint) zurück, da es sich bei den Quelldaten um Mehrpunktangaben handelte. Wenn die Quelldaten eine Dimension von 0 (Punkt oder Mehrpunkt) haben, ist eine exakte Übereinstimmung erforderlich.

```
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
0000000)
```

```
          2 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000 23.9
8000000 7.00000000)
```

```
2 record(s) selected.
```

M

M verwendet einen Punkt und gibt sein Maß zurück.

Syntax

```
db2gse.M(p db2gse.ST_Point)
```

Rückgabebetyp

Double

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle M_TEST. M_TEST enthält zwei Spalten: die Integer-Spalte GID, die die Zeile eindeutig kennzeichnet, und die Point-Spalte PT1, in der die Mustergeometrie gespeichert wird.

```
CREATE TABLE M_TEST (gid integer, pt1 db2gse.ST_Point)
```

Die folgenden INSERT-Anweisungen fügen eine Zeile ein, die einen Punkt mit Maßangaben enthält, und eine Zeile mit einem Punkt ohne Maß.

```
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)))
```

In der folgenden Anweisung SELECT und der entsprechenden Ergebnisgruppe listet die Funktion M die Maßwerte der Punkte auf. Da der erste Punkt kein Maß hat, gibt die Funktion M den Wert NULL zurück.

```
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 verwendet eine Form des Typs Mehrlinienfolge (Multilinestring) und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrlinienfolge zurück.

Syntax

```
db2gse.MLineFromShape(ShapeMultiLineString Blob(1M), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiLineString
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle WATERWAYS mit einer eindeutigen ID, einem Namen und einer Mehrlinienfolge WATER aus.

Die Tabelle WATERWAYS wird erstellt mit den Spalten ID und NAME, die jeden in der Tabelle gespeicherten Fluss und Wasserlauf kennzeichnen. Die Spalte WATER ist eine Mehrlinienfolge, da der Fluss und die Fluss-Systeme häufig eine Zusammenfassung verschiedener Linienfolgen sind.

```
CREATE TABLE WATERWAYS (id          integer,  
                          name       varchar(128),  
                          water      db2gse.ST_MultiLineString);  
  
/* Create the SQL insert statement to populate the id, name and  
   multilinestring. The question marks are parameter markers that  
   indicate the id, name and water values that will be retrieved at  
   runtime. */  
strcpy (shp_sql, "insert into WATERWAYS (id,name,water)  
values (?,?, db2gse.MLineFromShape (cast(? as blob(1m)),  
db2gse.coordref()..srid(0)))");  
  
/* Allocate memory for the SQL statement handle and associate the  
   statement handle with the connection handle. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Prepare the SQL statement for execution. */  
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);  
  
/* Bind the integer id value to the first parameter. */  
  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);  
/* Bind the varchar name value to the second parameter. */
```

```
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
    SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);

/* Bind the shape to the third parameter. */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

MPointFromShape

MPointFromShape verwendet eine Form des Typs Multipoint (Mehrpunkt) und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrpunktangabe zurück.

Syntax

```
db2gse.MPointFromShape(ShapeMultiPoint (1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_MultiPoint
```

Beispiele

Dieser Codeabschnitt füllt die Tabelle SPECIES_SITINGS eines Biologen aus.

Die Tabelle SPECIES_SITINGS wird mit drei Spalten erstellt. Die Spalten "species" und "genus" kennzeichnen jede Zeile eindeutig, während die Mehrpunktangabe "sitings" die Standorte der Spezies speichert.

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                               genus varchar(32),
                               sitings db2gse.ST_MultiPoint);

/* Create the SQL insert statement to populate the species, genus and
   sitings. The question marks are parameter markers that indicate the
   name and water values that will be retrieved at runtime. */
strcpy (shp_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.MPointFromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Bind the varchar species value to the first parameter. */
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, species_len, 0, species, species_len, &pcbvalue1);

/* Bind the varchar genus value to the second parameter. */
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, genus_len, 0, name, genus_len, &pcbvalue2);

/* Bind the shape to the third parameter. */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, sitings_len, 0, sitings_shape, sitings_len, &pcbvalue3);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

MPolyFromShape

MPolyFromShape verwendet eine Form des Typs Multipolygon und die Kennung eines räumlichen Bezugssystems und gibt ein Multipolygon zurück.

Syntax

```
db2gse.MPolyFromShape(ShapeMultiPolygon Blob(1m), SRID db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiPolygon
```

Beispiele

Dieser Codeabschnitt füllt die Tabelle LOTS aus.

Die Tabelle LOTS speichert die lot_id, die jedes Grundstück eindeutig kennzeichnet und das Multipolygon für das Grundstück, die die Geometrie für die Grundstückslinie enthält.

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );

/* Create the SQL insert statement to populate the lot_id and lot. The
   question marks are parameter markers that indicate the lot_id and lot
   values that will be retrieved at runtime. */
strcpy (shp_sql,"insert into LOTS (lot_id,lot)
values (?, db2gse.MPolyFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Bind the lot_id integer value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* Bind the lot shape to the second parameter. */
pcbvalue2 = lot_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, lot_len, 0, lot_shape, lot_len, &pcbvalue2);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

PointFromShape

PointFromShape verwendet eine Form des Typs Point (Punkt) und die Kennung eines räumlichen Bezugssystems und gibt einen Punkt zurück.

Syntax

```
db2gse.PointFromShape(db2gse.ShapePoint blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Dieser Programmabschnitt füllt die Tabelle HAZARDOUS_SITES aus.

Die Gefahrenstandorte werden in der Tabelle HAZARDOUS_SITES gespeichert, die mit der nachfolgenden Anweisung CREATE TABLE erstellt wird. Die als Punkt definierte Spalte location speichert einen Standort, der das geografische Zentrum jedes Gefahrenstandorts darstellt.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);

/* Create the SQL insert statement to populate the site_id, name and
   location. The question marks are parameter markers that indicate the
   site_id, name and location values that will be retrieved at runtime. */
strcpy (shp_sql,"insert into HAZARDOUS_SITES (site_id, name, location)
values (?,?, db2gse.PointFromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* Bind the site_id integer value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* Bind the name varchar value to the second parameter. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* Bind the location shape to the third parameter. */
pcbvalue3 = location_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, location_len, 0, location_shape, location_len, &pcbvalue3);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

PolyFromShape

PolyFromShape verwendet eine Form des Typs Polygon und die Identität eines räumlichen Bezugssystems und gibt ein Polygon zurück.

Syntax

```
db2gse.PolyFromShape (ShapePolygon Blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Polygon
```

Beispiele

Dieser Programmabschnitt füllt die Tabelle SENSITIVE_AREAS aus. Die Fragezeichen sind Parametermarken für die Werte ID, den Namen, die Größe, den Typ und die Zone, die zur Laufzeit abgerufen werden.

Die Tabelle SENSITIVE_AREAS enthält verschiedene Spalten, die die gefährdeten Institutionen beschreiben, zusätzlich zu der Spalte zone, die die Polygeometrie der Institution beschreibt.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

```
/* Create the SQL insert statement to populate the id, name, size, type and
   zone. The question marks are parameter markers that indicate the
   id, name, size, type and zone values that will be retrieved at runtime. */
strcpy (shp_sql,"insert into SENSITIVE_AREAS (id, name, size, type, zone)
values (?, ?, ?, ?, db2gse.PolyFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");
```

```
/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);
```

```
/* Bind the id integer value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
                      SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);
```

```
/* Bind the name varchar value to the second parameter. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
                      SQL_CHAR, 0, 0, name, 0, &pcbvalue2);
```

```
/* Bind the size float to the third parameter. */
pcbvalue3 = 0;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
                      SQL_REAL, 0, 0, &size, 0, &pcbvalue3);
```

```

/* Bind the type varchar to the fourth parameter. */
pcbvalue4 = type_len;
rc = SQLBindParameter (hstmt, 4, SQL_PARAM_INPUT, SQL_C_CHAR,
    SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);

/* Bind the zone polygon to the fifth parameter. */
pcbvalue5 = zone_len;
rc = SQLBindParameter (hstmt, 5, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, zone_len, 0, zone_shp, zone_len, &pcbvalue5);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);

```

ShapeToSQL

ShapeToSQL konstruiert einen Wert für `db2gse.ST_Geometry` mit seiner Formdarstellung. Der SRID-Wert Null wird automatisch verwendet.

Syntax

```
db2gse.ShapeToSQL(ShapeGeometry blob(1M))
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Der folgende C-Codeausschnitt enthält ODBC-Funktionen, die in SQL-Funktionen des Spatial Extender eingebettet sind; diese Funktionen fügen Daten in die Tabelle LOTS ein. Die Tabelle LOTS wurde mit zwei Spalten erstellt: der Spalte `lot_id`, die jedes Grundstück eindeutig kennzeichnet, und der Multipolygonspalte `lot`, die die Geometrie jedes Grundstücks enthält.

```
CREATE TABLE lots (lot_id integer,  
                   lot db2gse.ST_MultiPolygon);
```

Die Funktion `ShapeToSQL` wandelt Formen in eine Spatial Extender-Geometrie um. Die gesamte Anweisung `INSERT` wird in `shp_sql` kopiert. Die Anweisung `INSERT` enthält Parametermarken zum dynamischen Akzeptieren der `LOT_ID`-Daten und der `LOT`-Daten.

```
/* Create the SQL insert statement to populate the lot id and the  
   lot multipolygon. The question marks are parameter markers that  
   indicate the lot_id and lot values that will be retrieved at  
   run time. */
```

```
strcpy (shp_sql, "insert into lots (lot_id, lot) values (?,  
db2gse.ShapeToSQL(cast(? as blob(1m)))");
```

```
/* Allocate memory for the SQL statement handle and associate the  
   statement handle with the connection handle. */
```

```
rc = SQLAllocStmt (handle, &hstmt);
```

```
/* Prepare the SQL statement for execution. */
```

```
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);
```

```
/* Bind the integer key value to the first parameter. */
```

```
pcbvalue1 = 0;
```

```
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Bind the shape to the second parameter. */
```

```
pcbvalue2 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
  
/* Execute the insert statement. */  
  
rc = SQLExecute (hstmt);
```

ST_Area

ST_Area verwendet ein Polygon oder ein Multipolygon und gibt seine Fläche zurück.

Syntax

```
db2gse.ST_Area(s db2gse.ST_Surface)
db2gse.ST_Surface
db2gse.ST_Polygondb2gse.ST_MultiSurface
db2gse.ST_MultiPolygon
```

Rückgabetyt

Double

Beispiele

Der Stadtplaner benötigt eine Liste der Gebäude. Zum Abrufen der Liste wählt ein GIS-Techniker die Gebäude-ID und die Grundfläche der einzelnen Gebäude aus.

Die Gebäudegrundflächen sind in der Tabelle BUILDINGFOOTPRINTS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird:

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

Für die Anforderung des Stadtplaners verwendet der Techniker die folgende Anweisung SELECT zur Auswahl des eindeutigen Schlüssels, der building_id und der Grundfläche jedes Gebäudes aus der Tabelle BUILDINGFOOTPRINTS:

```
SELECT building_id, db2gse.ST_Area (footprint) "Area"
FROM BUILDINGFOOTPRINTS;
```

Die Anweisung SELECT gibt die folgende Ergebnisgruppe zurück:

building_id	Area
506	+1.407680000000000E+003
1208	+2.557590000000000E+003
543	+1.807860000000000E+003
178	+2.086710000000000E+003
.	.
.	.
.	.

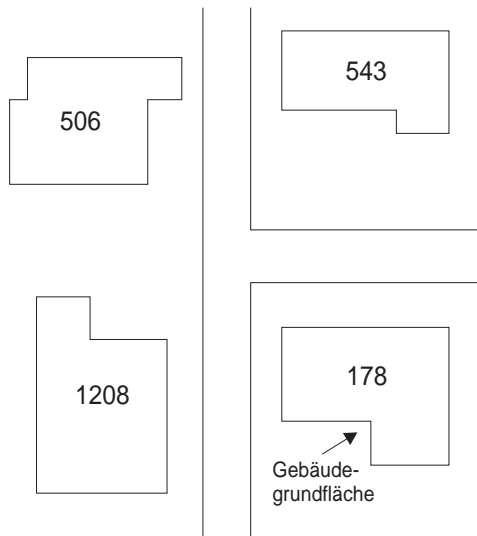


Abbildung 28. Über die Fläche eine Gebäudegrundfläche finden. Vier der fünf Gebäudegrundflächen mit ihren Gebäude-ID-Nummern werden entlang der angrenzenden Straße angezeigt.

Anmerkung: Wie bereits erwähnt, kann ST_Area ein Polygon oder ein Multipolygon als Eingabe verwenden. Als Datentyp für den Eingabeparameter kommen daher folgende Typen in Frage:

- Einer der für Polygone verwendeten Datentypen (db2gse.ST_Surface oder db2gse.ST_Polygon)
- Einer der für Multipolygone verwendeten Datentypen (db2gse.ST_MultiSurface oder db2gse.ST_MultiPolygon)

ST_AsBinary

ST_AsBinary verwendet ein Geometrieobjekt und gibt seine bekannte binäre Darstellung zurück. ST_AsBinary kann keine leere Geometrie als Eingabe verwenden (SQLSTATE 38827).

Syntax

```
db2gse.ST_AsBinary(g db2gse.ST_Geometry)
```

Rückgabetyt

BLOB(1m)

Beispiele

Der folgende Codeabschnitt verdeutlicht, wie die Funktion ST_AsBinary die Flächen-Multipolygone der Tabelle BUILDINGFOOTPRINTS in WKB-Multipolygone umwandelt. Diese Multipolygone werden an die Funktion draw_polygon zur Anzeige übergeben.

```
/* Create the SQL expression. */
strcpy(sqlstmt, "select db2gse.ST_AsBinary (footprint) from BUILDINGFOOTPRINTS
where db2gse.EnvelopesIntersect(footprint, db2gse.ST_PolyFromWKB
(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Prepare the SQL statement. */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* Set the pcbvalue1 length of the shape. */
pcbvalue1 = blob_len;

/* Bind the shape parameter */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* Execute the query */
rc = SQLExecute (hstmt);

/* Assign the results of the query (the Zone polygons) to the
  fetched_binary variable.
  */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* Fetch each polygon within the display window and display it. */
while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
  draw_polygon(fetched_binary);
```

ST_AsText

db2gse.ST_AsText verwendet ein Geometrieobjekt und gibt seine bekannte Textdarstellung zurück.

Syntax

```
db2gse.ST_AsText(g db2gse.ST_Geometry)
```

Rückgabetyt

Varchar(4000)

Beispiele

Im folgenden Szenario wandelt die Funktion db2gse.ST_AsText den Punkt HAZARDOUS_SITES "location" in seine Textbeschreibung um:

```
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(0)));

SELECT site_id, name, cast(db2gse.ST_AsText(location) as varchar(40))
       "Location"
FROM HAZARDOUS_SITES;
```

Die Anweisung SELECT gibt die folgende Ergebnisgruppe zurück:

SITE_ID	Name	Location
102	W. H. Kleenare Chemical Repository	POINT (1020.000000000 324.000000000)

ST_Boundary

ST_Boundary verwendet ein Geometrieobjekt und gibt seine kombinierte Begrenzung als Geometrieobjekt zurück. Punkte und Mehrpunktangaben ergeben immer eine Begrenzung in Form einer leeren Geometrie mit der Dimension 0 (und nicht mit der Dimension 1).

Syntax

```
db2gse.ST_Boundary(g db2gse.ST_Geometry)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Im folgenden Codeabschnitt wird eine Tabelle mit dem Namen BOUNDARY_TEST erstellt. BOUNDARY_TEST enthält zwei Spalten: GEOTYPE, als Varchar definiert, und G1, als die Superklassegeometrie definiert. Die folgenden Anweisungen INSERT fügen jeweils eine der Unterklassegeometrien ein. Die Funktion ST_Boundary ruft die Begrenzungen jeder Unterklasse ab, die in der Geometriespalte G1 gespeichert ist. Beachten Sie, dass die Dimension der resultierenden Geometrie immer um eins niedriger ist als die Eingabegeometrie. Punkte und Mehrpunktangaben führen immer zu einer Begrenzung in einer leeren Geometrie, Dimension 1. Linienfolgen und Mehrlinienfolgen geben eine Mehrpunktbegrenzung, Dimension 0, zurück. Ein Polygon oder ein Multipolygon gibt immer eine Mehrlinienbegrenzung, Dimension 1, zurück.

```
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
Multipoint	POINT EMPTY
Multilinestring	MULTIPOINT (9.55000000 23.75000000, 10.02000000
Multipolygon	MULTILINESTRING ((51.71000000 21.73000000, 73.36000000

6 record(s) selected.

ST_Buffer

ST_Buffer verwendet ein Geometrieobjekt und eine Entfernung und gibt das Geometrieobjekt zurück, das das Quellenobjekt umgibt.

Syntax

```
db2gse.ST_Buffer(g db2gse.ST_Geometry , [measure] Double)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Der zuständige Planer benötigt eine Liste der Gefahrenstandorte, in deren Umkreis von fünf Meilen sensible Bereiche wie Schulen, Krankenhäuser und Pflegeheime liegen. Die sensiblen Bereiche sind in der Tabelle SENSITIVE_AREAS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die Spalte ZONE ist als Polygon definiert; dieses Polygon stellt den Umriss der jeweiligen sensiblen Bereiche dar.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

Die Gefahrenstandorte sind in der Tabelle HAZARDOUS_SITES gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die als Punkt definierte Spalte LOCATION speichert einen Standort, der das geografische Zentrum jedes Gefahrenstandorts darstellt.

```
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)))
```

Die Tabellen SENSITIVE_AREAS und HAZARDOUS_SITES werden über die Funktion db2gse.ST_Overlaps verknüpft. Die Funktion gibt 1 (TRUE) zurück für alle Zeilen SENSITIVE_AREAS, deren Zonenpolygone den Puffer von fünf Meilen um den Punkt HAZARDOUS_SITES schneiden.

```
SELECT sa.name "Sensible Bereiche", hs.name "Gefahrenpunkte"
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Overlaps(sa.zone, db2gse.ST_Buffer (hs.location,(5 * 5280)))
=1;
```

(5 * 5280 ist gleichbedeutend mit fünf Meilen. Dies folgt aus der Tatsache, dass eine Meile 5280 Fuß umfasst, und dass ein Fuß die lineare Einheit des Koordinatensystems ist, zu dem die in der Anweisung VALUES angegebenen Koordinaten gehören).

In Abb. 29 liegen einige der sensiblen Bereiche in diesem Verwaltungsbezirk innerhalb des Fünf-Meilen-Puffers um die Gefahrenstandorte. Beide Fünf-Meilen-Puffer enthalten das Krankenhaus, einer davon enthält die Schule. Das Pflegeheim liegt außerhalb beider Bereiche.

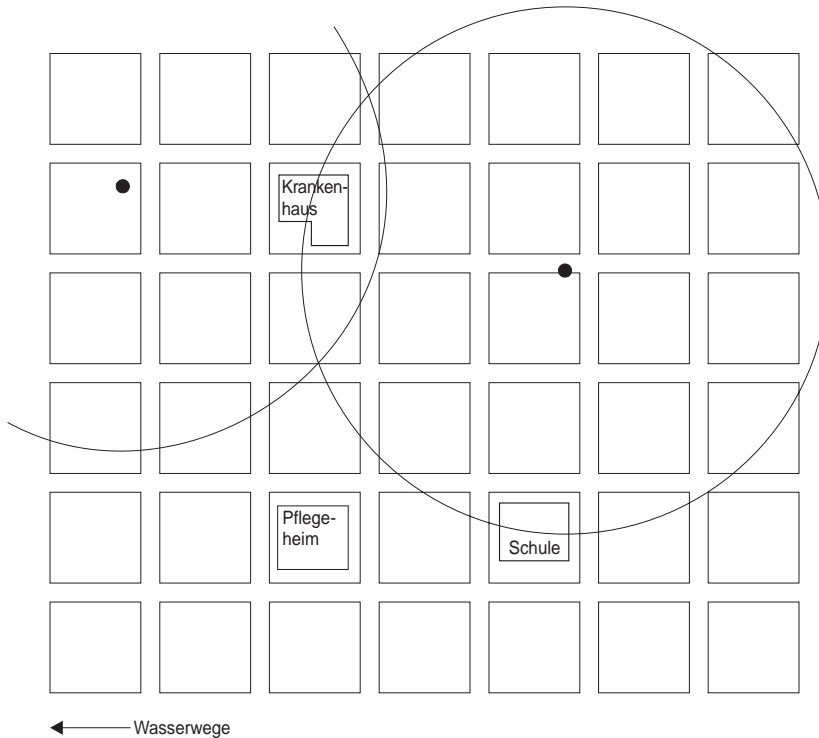


Abbildung 29. Ein Puffer mit einem Fünf-Meilen-Radius wird auf einen Punkt angewendet

ST_Centroid

ST_Centroid verwendet ein Polygon oder ein Multipolygon und gibt seinen geometrischen Mittelpunkt als Punkt zurück.

Syntax

```
db2gse.ST_Centroid(s db2gse.ST_Surface)
db2gse.ST_Centroid(ms db2gse.ST_MultiSurface)
```

Rückgabotyp

Für die Oberfläche: db2gse.ST_Point

Beispiele

Der städtische GIS-Techniker möchte die Multipolygone der Gebäudegrundflächen als einzelne Punkte in einer Grafik zur Bebauungsdichte anzeigen.

Die Gebäudegrundflächen sind in der Tabelle BUILDINGFOOTPRINTS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wurde.

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,
                                   lot_id      integer,
                                   footprint   db2gse.ST_MultiPolygon);
```

Die Funktion ST_Centroid gibt den Mittelpunkt jedes Multipolygons für die Gebäudegrundflächen zurück. Die Funktion AsShape wandelt den Mittelpunkt in eine Form um, die von der Anwendung erkannte externe Darstellung.

```
SELECT building_id,
       CAST(db2gse.AsShape(db2gse.ST_Centroid (footprint)) as blob(1m))
  "Centroid"
FROM BUILDINGFOOTPRINTS;
```

ST_Contains

ST_Contains verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn das erste Objekt das zweite vollständig beinhaltet, andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Contains(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Im nachfolgenden Beispiel werden zwei Tabellen erstellt. Eine Tabelle enthält die Gebäudegrundflächen ("footprints") der Stadt, die andere Tabelle die Grundstücke ("lots"). Der Stadtplaner möchte sicherstellen, dass alle Gebäudegrundflächen vollständig auf dem jeweiligen Grundstück liegen.

In beiden Tabellen speichert der Datentyp Multipolygon die Geometrie der Gebäudegrundflächen und der Grundstücke. Der Datenbank-Designer hat für beide Funktionen Multipolygone ausgewählt. Der Designer hat festgestellt, dass die Grundstücke durch natürliche Merkmale wie beispielsweise Flüsse unterbrochen sein können und dass die Gebäudegrundflächen häufig mehrere Gebäude umfassen.

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,  
                                lot_id integer,  
                                footprint db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );
```

Der Stadtplaner wählt zunächst die Gebäude aus, die sich nicht vollständig auf einem Grundstück befinden.

```
SELECT building_id  
FROM BUILDINGFOOTPRINTS, LOTS  
WHERE db2gse.ST_Contains(lot,footprint) = 0;
```

Der Stadtplaner stellt fest, dass die erste Abfrage eine Liste aller Gebäude-IDs zurückgibt, deren Gebäudegrundfläche außerhalb eines Grundstückspolygons liegt. Er weiß aber auch, dass diese Information kein Hinweis darauf ist, ob den anderen Gebäuden die richtige Grundstücks-ID zugeordnet ist. Diese zweite Abfrage führt eine Datenintegritätsprüfung mit der Spalte lot_id der Tabelle BUILDINGFOOTPRINTS aus.

```
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;
```

In Abb. 30 liegen die mit ihren Gebäude-IDs gekennzeichneten Gebäudegrundflächen innerhalb der jeweiligen Grundstücke. Die Grundstückslinien sind als gepunktete Linien dargestellt. Dies ist zwar in der Abbildung nicht dargestellt; diese Linien reichen jedoch bis zur Mitte der Straße und umfassen die Grundstücke und die darauf enthaltenen Gebäudegrundflächen vollständig.

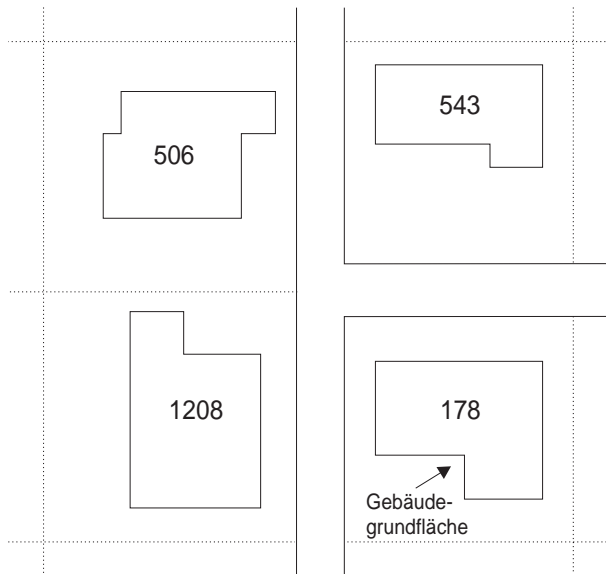


Abbildung 30. Mit `ST_Contains` sicherstellen, dass alle Gebäude vollständig auf dem jeweiligen Grundstück liegen

ST_ConvexHull

ST_ConvexHull verwendet ein Geometrieobjekt und gibt die konvexe Hölse zurück.

Syntax

```
db2gse.ST_ConvexHull(g db2gse.ST_Geometry)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Dieses Beispiel erstellt die Tabelle CONVEXHULL_TEST mit zwei Spalten: GEOTYPE und G1. Die Spalte GEOTYPE, eine Angabe varchar(20), speichert den Namen der Unterklasse einer in G1 gespeicherten Geometrie, die als Geometrie definiert ist.

```
CREATE TABLE CONVEXHULL_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Jede Anweisung INSERT fügt eine Geometrie jedes Unterklassentyps in die Tabelle CONVEXHULL_TEST ein.

```
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)))

```

Die folgende Anweisung SELECT listet den Unterklassennamen auf, der in der Spalte GEOTYPE und der konvexen Hülse gespeichert ist. Die von der Funktion ST_ConvexHull generierte konvexe Hülse wird mit der Funktion ST_AsText in Text umgewandelt. Anschließend wird sie in eine Angabe varchar(256) umgesetzt, da die Standardausgabe von ST_AsText varchar(4000) ist.

```

SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_ConvexHull(G1))
as varchar(256) "The convexhull"
FROM CONVEXHULL_TEST

```

ST_CoordDim

ST_CoordDim gibt die Koordinatendimensionen des Werts ST_Geometry zurück. Eine Erläuterung der Koordinatendimensionen finden Sie im Abschnitt „Punkte“ auf Seite 165.

Syntax

```
db2gse.ST_CoordDim(g1 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Die Tabelle coorddim_test wird mit den Spalten geotype und g1 erstellt. Die Spalte geotype speichert den Namen der in der Geometriespalte g1 gespeicherten Geometrieunterklasse.

```
CREATE TABLE coorddim_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Die INSERT-Anweisungen fügen eine Muster-Unterklasse in die Tabelle coorddim_test ein.

```
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))
)
```

Die Anweisung SELECT listet den Unterklassennamen auf, der in der Spalte geotype mit der Koordinatendimension dieses Geotyps gespeichert ist.

```
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 verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn deren Schnittmenge in einem Geometrieobjekt resultiert, deren Dimension um eins niedriger ist als die maximale Dimension der Quellenobjekte. Das Schnittobjekt enthält Punkte, die in beiden Quellengeometrieobjekten liegen; es ist jedoch nicht mit einem der Quellenobjekte identisch. Andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Crosses(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Die Regierung des Landkreises denkt über eine neue Vorschrift nach, die festlegt, dass alle Lagerstätten für gefährliche Abfälle im Landkreis nicht näher als fünf Meilen an Flussläufen oder Bächen liegen dürfen. Der zuständige GIS-Manager verfügt über eine exakte Darstellung der Flüsse und Bäche, die als Mehrlinienfolgen in der Tabelle WATERWAYS gespeichert sind. Für die Lagerstätten von gefährlichen Abfällen hat er jedoch nur einen einzigen Standortpunkt.

```
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);
```

Es soll festgestellt werden, ob die Landkreisregierung über Einrichtungen informiert werden muss, die die vorgeschlagene Vorschrift nicht erfüllen. Der GIS-Manager muss hierzu einen Bereich um die Gefahrenpunkte definieren und feststellen, ob sich Flüsse oder Bäche mit dem Polygon schneiden. ST_Crosses vergleicht die Bereiche aus HAZARDOUS_SITES mit WATERWAYS. Auf diese Weise werden nur die Datensätze zurückgegeben, bei denen ein Wasserlauf über den vorgeschlagenen Radius um den Gefahrenpunkt verläuft.

```
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;
```

In Abb. 31 schneidet der Fünf-Meilen-Puffer um den Gefahrenpunkt das Wassernetz des Landkreises. Das Netz der Wasserläufe wurde als Mehrlinienfolge definiert. Die Ergebnisgruppe umfasst somit alle Liniensegmente, die Teil der Segmente sind, die den Radius schneiden.

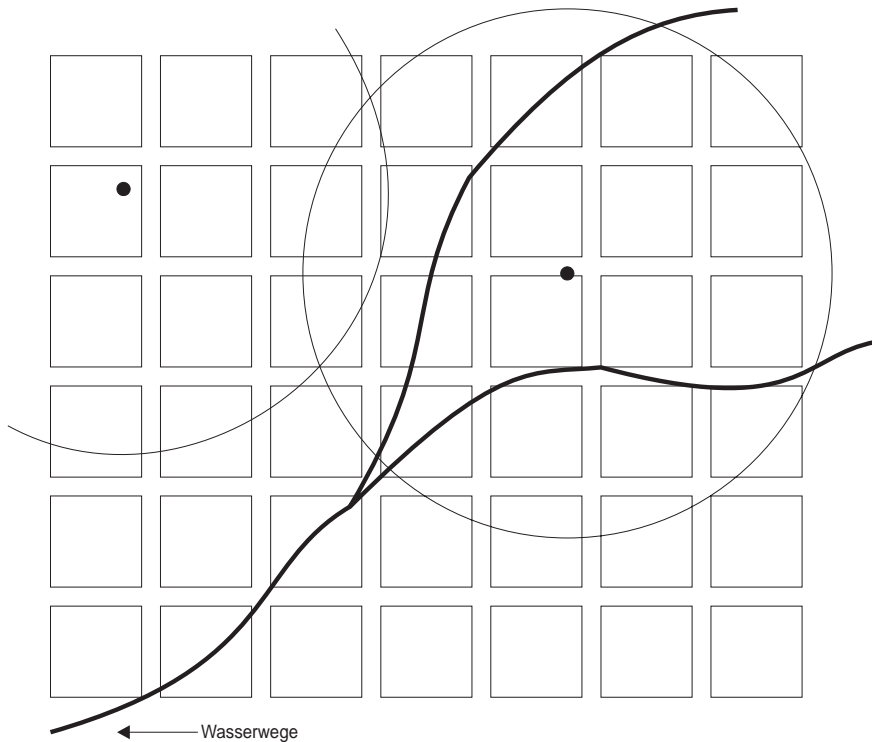


Abbildung 31. Mit `ST_Crosses` die Wasserwege ermitteln, die durch den Lagerungsbereich für gefährliche Abfälle fließen.

ST_Difference

ST_Difference verwendet zwei Geometrieobjekte und gibt ein Geometrieobjekt zurück, das die Differenz der beiden Eingabeobjekte darstellt. Die beiden Eingabegeometrien von ST_Difference müssen dieselbe Dimension aufweisen. Andernfalls wird eine Null zurückgegeben.

Syntax

```
db2gse.ST_Difference(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Der Stadtplaner muss die Gesamtfläche der Grundstücke in der Stadt kennen, auf denen kein Gebäude steht. Das heißt, der Stadtplaner braucht die Summe der Fläche nach Abzug der bebauten Fläche.

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,  
                                lot_id      integer,  
                                footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer,  
                   lot   db2gse.ST_MultiPolygon);
```

Der Stadtplaner erstellt eine Gleichsetzungsverknüpfung zwischen den Tabellen BUILDINGFOOTPRINTS und LOTS über die lot_id. Anschließend verwendet er die Flächensumme der Differenzen zwischen Grundstücke und den Gebäudegrundflächen.

```
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 verwendet ein Geometrieobjekt und gibt seine Dimension zurück.

Syntax

```
db2gse.ST_Dimension(g1 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Die Tabelle DIMENSION_TEST wird mit den Spalten GEOTYPE und G1 erstellt. Die Spalte GEOTYPE speichert den Namen der in der Geometriespalte G1 gespeicherten Geometrieunterklasse.

```
CREATE TABLE DIMENSION_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Die INSERT-Anweisungen fügen eine Muster-Unterklasse in die Tabelle DIMENSION_TEST ein.

```
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)))

```

Die folgende Anweisung SELECT listet den Unterklassennamen auf, der in der Spalte GEOTYPE mit der Dimension dieses Geotyps gespeichert ist.

```

SELECT geotype, db2gse.ST_Dimension(g1) "The dimension"
FROM DIMENSION_TEST

```

Das folgende Ergebnis wird zurückgegeben.

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 verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn die Schnittmenge der beiden Geometrien eine leere Menge ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Disjoint(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Eine Versicherungsgesellschaft möchte die Höhe der Versicherungssumme für die Krankenhäuser, Pflegeheime und Schulen einer Stadt überprüfen. Im Rahmen dieses Prozesses wird auch die Gefährdung beurteilt, die die Lagerstätten für gefährliche Abfälle für die einzelnen Institutionen darstellen. Die Versicherungsgesellschaft möchte nur die Institutionen berücksichtigen, bei denen kein Risiko der Kontaminierung besteht. Der von der Versicherungsgesellschaft eingestellte GIS-Berater wurde angewiesen, alle Institutionen zu ermitteln, die nicht innerhalb eines Radius von fünf Meilen innerhalb einer Lagerstätte mit gefährlichen Abfällen liegen.

Die Tabelle SENSITIVE_AREAS enthält verschiedene Spalten, die die gefährdeten Institutionen beschreiben, zusätzlich zu der Spalte ZONE, die die Polygeometrie der Institution beschreibt.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

In der Tabelle HAZARDOUS_SITES ist die Kennung der Standorte in den Spalten SITE_ID und NAME gespeichert; der tatsächliche geografische Standort wird jeweils in der Spalte LOCATION gespeichert.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

Die folgende Anweisung SELECT listet die Namen aller sensiblen Bereiche auf, die sich nicht innerhalb des Fünf-Meilen-Radius von einer Lagerstätte gefährlicher Abfälle befinden. Die Funktion ST_Intersects könnte die Funktion ST_Disjoint in dieser Abfrage ersetzen, wenn das Ergebnis der Funktion auf 0 statt auf 1 gesetzt wird. ST_Intersects und ST_Disjoint geben genau entgegengesetzte Ergebnisse zurück.

```
SELECT sa.name
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Disjoint(db2gse.ST_Buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

In Abb. 32 werden sensible Standorte mit dem Fünf-Meilen-Radius um die Lagerstätten gefährlicher Abfälle verglichen. Das Pflegeheim ist der einzige sensible Bereich, für den die Funktion `ST_Disjoint` da Ergebnis 1 (TRUE) zurückgibt. Die Funktion `ST_Disjoint` gibt 1 zurück, wenn sich zwei Geometrien in keiner Weise schneiden.

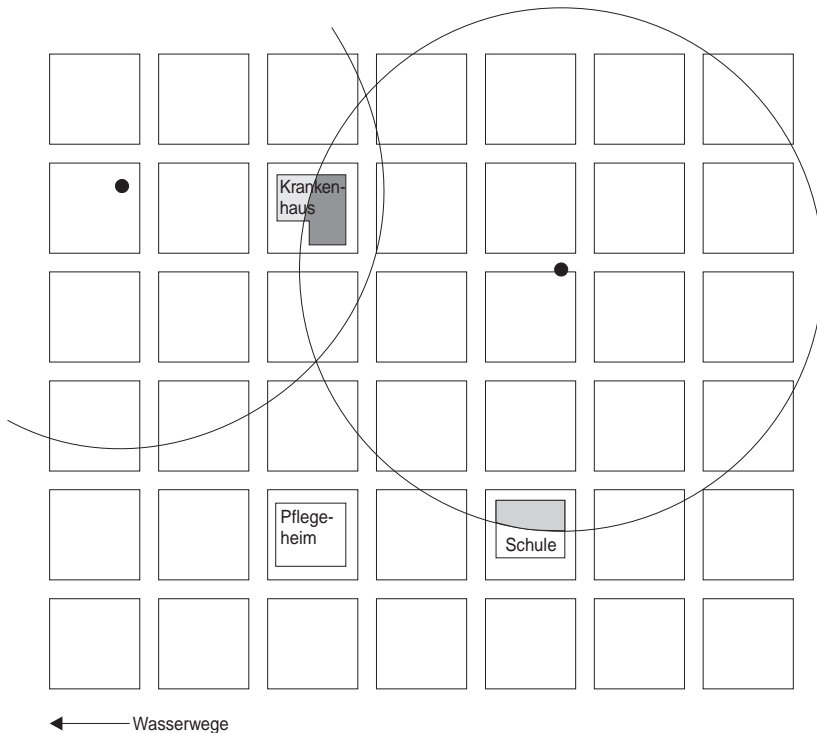


Abbildung 32. Mit `ST_Disjoint` die Gebäude ermitteln, die nicht innerhalb (in der Schnittmenge) eines Gefahrengebiets liegen

ST_Distance

ST_Distance verwendet zwei Geometrieobjekte und gibt die kleinste Entfernung zwischen diesen Objekten zurück.

Syntax

```
db2gse.ST_Distance(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Double

Beispiele

Der Stadtplaner benötigt eine Liste aller Gebäude, deren Entfernung bis zum Grundstücksrand maximal 1 Fuß (30,48 cm) beträgt.

Die Spalte BUILDING_ID der Tabelle BUILDINGFOOTPRINTS kennzeichnet jedes Gebäude eindeutig. Die Spalte LOT_ID kennzeichnet das Grundstück, zu dem das jeweilige Gebäude gehört. Das Multipolygon der Gebäudegrundfläche speichert die Geometrie der Grundfläche jedes Gebäudes.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
lot_id integer,  
footprint db2gse.ST_MultiPolygon);
```

Die Tabelle LOTS speichert die lot-ID, die jedes Grundstück eindeutig kennzeichnet sowie das Multipolygon, das die Geometrie der Grundstückslinie enthält.

```
CREATE TABLE LOTS ( lot_id integer,  
lot db2gse.ST_MultiPolygon);
```

Die Abfrage gibt eine Liste der Gebäude-IDs zurück, die einen Abstand von maximal 1 Fuß zu der jeweiligen Grundstückslinie haben. Die Funktion ST_Distance führt eine räumliche Verknüpfung zwischen den Grundflächen und der Begrenzung der Grundstückspolygone durch. Die Gleichsetzungsverknüpfung zwischen bf.lot_id und LOTS.lot_id stellt jedoch sicher, dass nur die Multipolygone zu dem gleichen Grundstück mit der Funktion ST_Distance verglichen werden.

```
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 verwendet eine Linienfolge und gibt einen Punkt zurück, der den letzten Punkt der Linienfolge darstellt.

Syntax

```
db2gse.ST_Endpoint(c db2gse.ST_Curve)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Die Tabelle ENDPOINT_TEST speichert die ganzzahlige Spalte GID; die jede Zeile und die Linien Spalte LN1, in der Mehrlinienfolgen gespeichert werden, eindeutig kennzeichnet.

```
CREATE TABLE ENDPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Die Anweisungen INSERT fügt Linienfolgen in die Tabelle ENDPOINT_TEST ein. Die erste Zeile hat keine Z-Koordinaten oder Maßangaben; die zweite dagegen schon.

```
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)))
```

Die folgende Anweisung SELECT listet die Spalte GID mit den Ergebnissen der Funktion ST_Endpoint auf. Die Funktion ST_Endpoint generiert eine Punktgeometrie, die von der Funktion ST_AsText in Text umgewandelt wird. Mit der Funktion CAST wird der Standardwert varchar(4000) der Funktion ST_AsText auf varchar(60) verkürzt.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_Endpoint(ln1)) AS varchar(60))
"Endpoint"
FROM ENDPOINT_TEST
```

Das folgende Ergebnis wird zurückgegeben.

```
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 verwendet ein Geometrieobjekt und gibt seinen Zeichenrahmen als Geometrie zurück.

Syntax

```
db2gse.ST_Envelope(g db2gse.ST_Geometry)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die Spalte GEOTYPE in der Tabelle ENVELOPE_TEST speichert den Namen der in der Geometriespalte G1 gespeicherten Geometrieunterklasse.

```
CREATE TABLE ENVELOPE_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen jede Geometrieunterklasse in die Tabelle ENVELOPE_TEST ein.

```
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))

```

Die folgende Anweisung SELECT listet den Namen der Unterklasse neben ihrem Umschlag auf. Da die Funktion ST_Envelope entweder einen Punkt, eine Linienfolge oder ein Polygon zurückgibt, wird ihre Ausgabe mit der Funktion ST_AsText in Text umgewandelt. Die Funktion CAST wandelt das Standardergebnis varchar(4000) der Funktion ST_AsText in ein Ergebnis varchar(280) um.

```

SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_Envelope(g1)) AS varchar(280))
"The envelope"
FROM ENVELOPE_TEST

```

Das folgende Ergebnis wird zurückgegeben.

GEOTYPE	The envelope
Point	POINT (10.02000000 20.01000000)
Linestring 40.01000000)	LINestring (10.01000000 20.01000000, 10.01000000 40.01000000)
Linestring	POLYGON ((10.02000000 20.01000000, 11.92000000 20.01000000, 11.92000000 25.64000000, 10.02000000 25.64000000, 10.02000000 20.01000000))
Polygon	POLYGON ((10.02000000 20.01000000, 25.02000000 20.01000000, 25.02000000 35.64000000, 10.02000000 35.64000000, 10.02000000 20.01000000))
Multipoint	POLYGON ((10.02000000 20.01000000, 11.92000000 20.01000000, 11.92000000 25.64000000, 10.02000000 25.64000000, 10.02000000 20.01000000))
Multilinestring 20.01000000)	LINestring (10.01000000 20.01000000, 50.01000000 20.01000000)
Multilinestring	POLYGON ((9.55000000 20.01000000, 15.36000000 20.01000000, 15.36000000 30.11000000, 9.55000000 30.11000000, 9.55000000 20.01000000))
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 vergleicht zwei Geometrien und gibt 1 (TRUE) zurück, wenn die Geometrien identisch sind; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Equals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Der GIS-Techniker vermutet, dass manche der Daten in der Tabelle BUILDINGFOOTPRINTS irgendwie dupliziert wurden. Der Techniker fragt die Tabelle ab, um festzustellen, ob irgendwelche der Multipolygone der Gebäudegrundflächen gleich sind.

Die Tabelle BUILDINGFOOTPRINTS wird mit der folgenden Anweisung erstellt. Die Spalte BUILDING_ID kennzeichnet die Gebäude eindeutig; die Spalte LOT_ID kennzeichnet das Grundstück zu dem Gebäude, und die Spalte FOOTPRINT speichert die Geometrie des Gebäudes.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
lot_id integer,  
footprint db2gse.ST_MultiPolygon);
```

Die Tabelle BUILDINGFOOTPRINTS ist über das Prädikat ST_Equals mit sich selbst räumlich verknüpft; dieses Prädikat gibt 1 zurück, wenn es zwei gleiche Multipolygone findet. Die Bedingung bf1.building_id <> bf2.building_id ist erforderlich, um den Vergleich mit der gleichen Geometrie zu eliminieren.

```
SELECT bf1.building_id, bf2.building_id  
FROM BUILDINGFOOTPRINTS bf1, BUILDINGFOOTPRINTS bf2  
WHERE db2gse.ST_Equals(bf1.footprint,bf2.footprint) = 1  
and bf1.building_id <> bf2.building_id;
```

ST_ExteriorRing

ST_ExteriorRing verwendet ein Polygon und gibt seinen äußeren Ring als Linienfolge zurück.

Syntax

```
db2gse.ST_ExteriorRing(s db2gse.ST_Polygon)
```

Rückgabetyt

```
db2gse.ST_LineString
```

Beispiele

Ein Ornithologe, der den Vogelbestand auf verschiedenen Südseeinseln untersucht, weiß, dass sich die Zone für die Nahrungssuche einer bestimmten Spezies auf die Küstenlinien beschränkt. Zur Berechnung des Bestands, der auf der Insel genügend Nahrung finden könnte, muss er die Länge der Küstenlinie (den Umfang) kennen. Auf einigen der Inseln gibt es zwar mehrere Seen; deren Ufer werden jedoch ausschließlich von einer aggressiveren Spezies bevölkert. Der Ornithologe braucht daher nur die Länge des äußeren Inselrings zu kennen.

Die Spalten ID und NAME der Tabelle ISLANDS kennzeichnen alle Inseln, und die Spalte LAND des Typs ST_Polygon speichert die Geometrie dieser Inseln.

```
CREATE TABLE ISLANDS (id    integer,
                       name  varchar(32),
                       land  db2gse.ST_Polygon);
```

Die Funktion ST_ExteriorRing extrahiert den äußeren Ring von jedem Inselpolygon als Linienfolge. Die Länge der Linienfolge wird über die Funktion length ermittelt. Die Längen der Linienfolge werden über die Funktion SUM zusammengefasst.

```
SELECT SUM(db2gse.ST_length(db2gse.ST_ExteriorRing (land))) FROM ISLANDS;
```

In Abb. 33 auf Seite 251 stellen die äußeren Ringe der Inseln die ökologische Schnittstelle zwischen den Inseln und dem Meer dar. Auf manchen der Inseln gibt es Seen, die als innere Ringe der Polygone dargestellt sind.

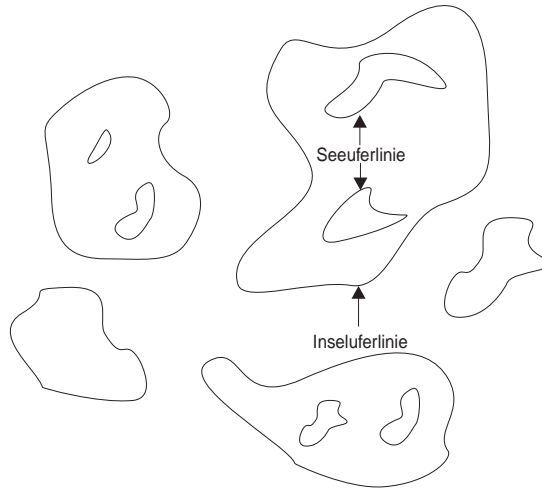


Abbildung 33. Mit `ST_ExteriorRing` die Länge der Küstenlinie einer Insel ermitteln.

ST_GeometryN

ST_GeometryN verwendet eine Gruppe und einen ganzzahligen Index und gibt das *n*te Geometrieobjekt in der Gruppe zurück.

Syntax

```
db2gse.ST_GeometryN(g db2gse.ST_Geometry, n Integer)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Der Stadtplaner muss wissen, ob alle Gebäudegrundflächen innerhalb des ersten Polygons des Multipolygons des Grundstücks liegen.

Die Spalte BUILDING_ID kennzeichnet jede Zeile der Tabelle BUILDING-FOOTPRINTS eindeutig. Die Spalte LOT_ID kennzeichnet das Grundstück, zu dem die einzelnen Gebäude gehören. Die Spalte FOOTPRINT speichert die Geometrie des Gebäudes.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
                                  lot_id         integer,  
                                  footprint      db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer,  
                    lot      db2gse.ST_MultiPolygon);
```

Die Abfrage listet die BUILDINGFOOTPRINTS building_id und die lot_id für alle Gebäudegrundflächen auf, die innerhalb des ersten Grundstückspolygons liegen. Die Funktion ST_GeometryN gibt das erste Grundstückspolygon in der Anordnung der Multipolygone zurück.

```
SELECT bf.building_id,bf.lot_id  
FROM BUILDINGFOOTPRINTS bf,LOTS  
WHERE db2gse.ST_Within(footprint, db2gse.ST_GeometryN (lot,1)) = 1  
      AND bf.lot_id = LOTS.lot_id;
```

ST_GeometryType

ST_GeometryType verwendet ein ST_Geometry-Objekt und gibt seinen Geometrietyp als Zeichenfolge zurück.

Syntax

```
db2gse.ST_GeometryType (g db2gse.ST_Geometry)
```

Rückgabetyt

```
Varchar(4000)
```

Beispiele

Die Tabelle GEOMETRYTYPE_TEST enthält die Geometriespalte G1.

```
CREATE TABLE GEOMETRYTYPE_TEST(g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen jede Geometrieunterklasse in die Spalte G1 ein.

```
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)))
```

Die folgende Anweisung SELECT listet den Geometrietyp jeder Unterklasse auf, die in der Geometriespalte G1 gespeichert ist.

```
SELECT db2gse.ST_GeometryType(g1) "Geometry type" FROM GEOMETRYTYPE_TEST
```

Das folgende Ergebnis wird zurückgegeben.

Geometry type

ST_Point
ST_LineString
ST_Polygon
ST_MultiPoint
ST_MultiLineString
ST_MultiPolygon

6 record(s) selected.

ST_GeomFromText

ST_GeomFromText verwendet eine bekannte Textdarstellung und die Kennung eines räumlichen Bezugssystems und gibt ein Geometrieobjekt zurück.

Syntax

```
db2gse.ST_GeomFromText(geometryTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Die Tabelle GEOMETRY_TEST enthält die ganzzahlige Spalte GID, die jede Zeile eindeutig kennzeichnet, und die Spalte G1, in der die Geometrie gespeichert wird.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Die INSERT-Anweisungen fügen die Daten in die Spalten GID und G1 der Tabelle GEOMETRY_TEST ein. Die Funktion ST_GeomFromText wandelt die Textdarstellung jeder Geometrie in die entsprechende exemplarfähige Spatial Extender-Unterklasse um.

```
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 verwendet eine bekannte binäre Darstellung und die Kennung eines räumlichen Bezugssystems und gibt ein Geometrieobjekt zurück.

Syntax

```
db2gse.ST_GeomFromWKB(WKBGeometry Blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Der folgende C-Codeausschnitt enthält ODBC-Funktionen, die in SQL-Funktionen des Spatial Extender eingebettet sind; diese Funktionen fügen Daten in die Tabelle LOTS ein.

Die Tabelle LOTS wurde mit zwei Spalten erstellt: der Spalte LOT_ID, die jedes Grundstück eindeutig kennzeichnet, und der Multipolygonspalte LOT, die die Geometrie jedes Grundstücks enthält.

```
CREATE TABLE LOTS ( lot_id integer,  
                    lot db2gse.ST_MultiPolygon);
```

Die Funktion ST_GeomFromWKB wandelt WKB-Darstellungen in Spatial Extender-Geometrien um. Die gesamte Anweisung INSERT wird in die Zeichenfolge wkb_sql kopiert. Die Anweisung INSERT enthält Parametermarken zum dynamischen Akzeptieren der LOT_ID-Daten und der LOT-Daten.

```
/* Create the SQL insert statement to populate the lot id and the  
   lot multipolygon. The question marks are parameter markers that  
   indicate the lot_id and lot values that will be retrieved at  
   runtime. */  
strcpy (wkb_sql,"insert into LOTS (lot_id, lot) values (?,  
db2gse.ST_GeomFromWKB  
  
(cast(? as blob(1m)), db2gse.coordref(..srid(0)))");  
  
/* Allocate memory for the SQL statement handle and associate the  
   statement handle with the connection handle. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Prepare the SQL statement for execution. */  
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);  
  
/* Bind the integer key value to the first parameter. */  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);  
  
/* Bind the shape to the second parameter. */  
pcbvalue2 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

ST_InteriorRingN

Gibt den n ten inneren Ring eines Polygons als Linienfolge zurück. Die Ringe sind nicht nach ihrer geometrischen Anordnung geordnet. Sie sind entsprechend den Regeln der internen geometrischen Prüfroutinen angeordnet. Die Reihenfolge der Ringe kann daher nicht vordefiniert werden.

Syntax

ST_InteriorRingN(p ST_Polygon, n Integer)

Rückgabebetyp

db2gse.ST_LineString

Beispiele

Ein Ornithologe, der den Vogelbestand auf verschiedenen Südseeinseln untersucht, weiß, dass sich die Zone für die Nahrungssuche einer bestimmten passiven Spezies auf die Küstenlinie beschränkt. Auf manchen der Inseln gibt es kleine Seen. Die Ufer der Seen werden ausschließlich von einer aggressiveren Spezies bevölkert. Der Ornithologe weiß, dass ab einer bestimmten Uferlänge an den Seen die aggressive Spezies so zahlreich wird, dass sie die passive Spezies an der Küste bedroht. Der Ornithologe will daher die Summe der Uferlängen der inneren Ringe der Inseln wissen.

In Abb. 34 stellen die äußeren Ringe der Inseln die ökologische Schnittstelle zwischen den Inseln und dem Meer dar. Auf manchen der Inseln gibt es Seen, die als innere Ringe der Polygone dargestellt sind.

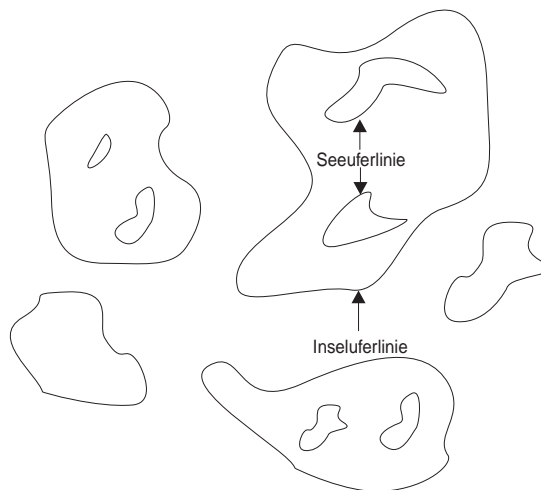


Abbildung 34. Mit ST_InteriorRingN die Länge der Seeufer auf allen Inseln ermitteln

Die Spalten ID und name der Tabelle ISLANDS kennzeichnen alle Inseln, und die Polygonspalte land speichert die Geometrie dieser Inseln.

```
CREATE TABLE ISLANDS (id    integer,
                       name  varchar(32),
                       land  db2gse.ST_Polygon);
```

Das folgende ODBC-Programm extrahiert mit der Funktion ST_InteriorRingN den inneren Ring (lake) aus jedem Inselpolygon als Linienfolge. Der Umfang der zurückgegebenen Linienfolge wird akkumuliert und zusammen mit der ID der Insel angezeigt.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sg.h"
#include "sgerr.h"
#include "sqlcli1.h"

/**          ***
*** Change these constants ***
***          ***/

#define USER_NAME    "sdetest" /* your user name */
#define USER_PASS    "acid.rain" /* your user password */
#define DB_NAME      "mydb" /* database to connect to */

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;

    /* Allocate memory for the ODBC environment handle henv and initialize
```

```

the application. */

    rc = SQLAllocEnv (&henv);
    if (rc != SQL_SUCCESS)
    {
        printf ("SQLAllocEnv failed with %d\n", rc);
        exit(0);
    }

/* Allocate memory for a connection handle within the henv environment. */

    rc = SQLAllocConnect (henv, &handle);
    if (rc != SQL_SUCCESS)
    {
        printf ("SQLAllocConnect failed with %d\n", rc);
        exit(0);
    }

/* Load the ODBC driver and connect to the data source identified
   by the database, user, and password.*/

    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");

/* Allocate memory to the SQL statement handle island_cursor. */

    rc = SQLAllocStmt (handle, &island_cursor);
    check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* Prepare and execute the query to get the island IDs and number of
   lakes (interior rings) */

    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");

/* Bind the island table's ID column to the variable island_id */

    rc = SQLBindCol (island_cursor, 1, SQL_C_SLONG, &island_id, 0, &id_ind);
    check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* Bind the result of numinteriorrings(land) to the num_lakes variable. */

    rc = SQLBindCol (island_cursor, 2, SQL_C_SLONG, &num_lakes, 0, &lake_ind);
    check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

```

```

/* Allocate memory to the SQL statement handle lake_cursor. */
rc = SQLAllocStmt (handle, &lake_cursor);
check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* Prepare the query to get the length of an interior ring. */

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");

/* Bind the lake_number variable as the first input parameter */

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");

/* Bind the island_id as the second input parameter */

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");

/* Bind the result of the Length(db2gse.ST_InteriorRingN(land, cast
(? as integer))) an die Variable lake_perimeter binden */

rc = SQLBindCol (lake_cursor, 1, SQL_C_SLONG, &lake_perimeter, 0,
        &length_ind);
check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* Outer loop, get the island ids and the number of lakes
(interior rings) */

while (SQL_SUCCESS == rc)
{
    /* Fetch an island */

    rc = SQLFetch (island_cursor);

    if (rc != SQL_NO_DATA)
    {
        check_sql_err (NULL, island_cursor, rc, "SQLFetch");

        /* Inner loop, for this island, get the perimeter of all of
        its lakes (interior rings) */

        for (total_perimeter = 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);
    }
}

/* Display the Island id and the total perimeter of its lakes. */

    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 fertig ...\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 verwendet zwei Geometrieobjekte und gibt die Schnittmenge als Geometrieobjekt zurück.

Wenn ST_Intersection ein Polygon und eine Linienfolge als Eingabeparameter erhält und die folgenden Bedingungen erfüllt sind:

- Ein Punkt des Polygons ist der Startpunkt der Linienfolge und
- Das Polygon und die Linienfolge haben keinen weiteren gemeinsamen Punkt,

dann gibt ST_Intersection die Textfolge POINT EMPTY zurück.

Syntax

```
db2gse.ST_Intersection(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabety

```
db2gse.ST_Geometry
```

Beispiele

Der Chef der Feuerwehr muss die Bereiche der Krankenhäuser, Schulen, und Pflegeheime abrufen, die sich mit dem Radius einer möglichen Kontamination durch gefährliche Abfälle schneiden.

Die sensiblen Bereiche sind in der Tabelle SENSITIVE_AREAS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die Spalte ZONE ist als Polygon definiert, in dem der Umriss der jeweiligen sensiblen Bereiche gespeichert ist.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,  
                               name        varchar(128),  
                               size        float,  
                               type        varchar(10),  
                               zone        db2gse.ST_Polygon);
```

Die Gefahrenstandorte sind in der Tabelle HAZARDOUS_SITES gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die als Punkt definierte Spalte LOCATION speichert einen Standort, der das geografische Zentrum jedes Gefahrenstandorts darstellt.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,  
                               name       varchar(128),  
                               location   db2gse.ST_Point);
```

Die Funktion buffer generiert einen Fünf-Meilen-Puffer um die Standorte der Lagerstätten für gefährlichen Abfall. Die Funktion ST_Intersection generierte Polygone aus der Schnittmenge der gepufferten Polygone um die Abfallstandorte und der sensiblen Bereiche.

Die Funktion `ST_Area` gibt die Polygonfläche der Schnittmenge zurück; diese wird mit der Funktion `SUM` als Summe über alle Gefahrenstandorte gebildet. Die Klausel `GROUP BY` weist die Abfrage an, die Schnittmengenflächen nach `site_ID` der Gefahrenstandorte zu akkumulieren.

```
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;
```

In Abb. 35 stehen die Kreise für die gepufferten Polygone um die Gefahrenstandorte. Die Schnittmenge dieser Pufferpolygone mit den Polygonen der sensiblen Bereiche erzeugen drei weitere Polygone. Das Krankenhaus links oben wird zwei Mal geschnitten; die Schule rechts unten nur ein Mal.

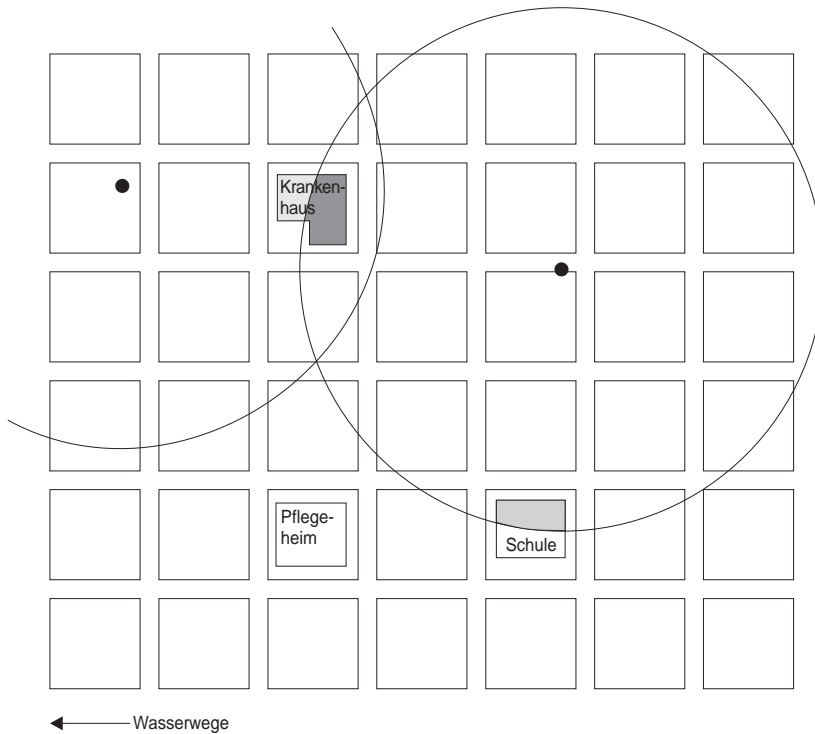


Abbildung 35. Mit `ST_Intersection` feststellen, wie groß der gefährdete Bereich in den Gebäuden ist

ST_Intersects

ST_Intersects verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn die Schnittmenge der beiden Geometrien keine leere Menge ergibt. Andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Intersects(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Der Chef der Feuerwehr benötigt eine Liste aller sensiblen Bereiche innerhalb eines Radius von 5 Meilen um die Lagerstätten für gefährlichen Abfall.

Die sensiblen Bereiche sind in der Tabelle SENSITIVE_AREAS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die Spalte ZONE ist als Polygon definiert, in dem der Umriss der jeweiligen sensiblen Bereiche gespeichert ist.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

Die Gefahrenstandorte sind in der Tabelle HAZARDOUS_SITES gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die als Punkt definierte Spalte LOCATION speichert einen Standort, der das geografische Zentrum jedes Gefahrenstandorts darstellt.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name        varchar(128),
                               location    db2gse.ST_Point);
```

Die Abfrage gibt eine Liste der sensiblen Bereiche und der Namen der Gefahrenstandorte zurück, sie sich mit dem Fünf-Meilen-Puffer um die Gefahrenstandorte schneiden.

```
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 verwendet eine Linienfolge oder eine Mehrlinienfolge und gibt 1 (TRUE) zurück, falls sie geschlossen ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_IsClosed(c db2gse.ST_Curve)
db2gse.ST_IsClosed(mc db2gse.ST_MultiCurve)
```

Rückgabetyt

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle CLOSED_LINESTRING mit einer einzigen Linienspalte.

```
CREATE TABLE CLOSED_LINESTRING (ln1 db2gse.ST_LineString)
```

Die folgenden INSERT-Anweisungen fügen zwei Datensätze in die Tabelle CLOSED_LINESTRING ein. Der erste Datensatz ist keine geschlossene Linie, der zweite Datensatz dagegen schon.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen die Ergebnisse der Funktion ST_IsClosed. Die erste Zeile gibt 0 zurück, weil die Linienfolge nicht geschlossen ist; die zweite Zeile gibt 1 zurück, weil die Linienfolge geschlossen ist.

```
SELECT db2gse.ST_IsClosed(ln1) "Is it closed" FROM CLOSED_LINESTRING
```

```
Is it closed
-----
0
1
```

2 record(s) selected.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle CLOSED_MULTILINESTRING mit einer einzigen Mehrlinienspalte.

```
CREATE TABLE CLOSED_MULTILINESTRING (m1n1 db2gse.ST_MultiLineString)
```

Die folgenden INSERT-Anweisungen fügen zwei Datensätze in CLOSED_MULTILINESTRING ein, einen Datensatz mit einer nicht geschlossenen Mehrlinienfolge und einen Datensatz mit einer geschlossenen Mehrlinienfolge.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen die Ergebnisse der Funktion ST_IsClosed. Die erste Zeile gibt 0 zurück, weil die Mehrlinienfolge nicht geschlossen ist; die zweite Zeile gibt 1 zurück, weil die Mehrlinienfolge geschlossen ist. Eine Mehrlinienfolge gilt als geschlossen, wenn alle ihre Linienfolgen geschlossen sind.

```
SELECT db2gse.ST_IsClosed(mln1) "Is it closed" FROM CLOSED_MULTILINESTRING
```

```
Is it closed
```

```
-----
```

```
0
```

```
1
```

```
2 record(s) selected.
```

ST_IsEmpty

ST_IsEmpty verwendet ein Geometrieobjekt und gibt 1 (TRUE) zurück, falls es leer ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_IsEmpty(g db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle EMPTY_TEST mit zwei Spalten. Die Spalte GEOTYPE speichert den Datentyp der in der Geometriespalte G1 gespeicherten Unterklassen.

```
CREATE TABLE EMPTY_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen zwei Datensätze für die Geometrieunterklassen Punkt, Linienfolge und Polygon ein. Ein Datensatz ist leer, der andere nicht.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen den Geometrietyp der Spalte GEOTYPE und die Ergebnisse der Funktion 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 verwendet eine Linienfolge und gibt 1 (TRUE) zurück, wenn es sich um einen Ring handelt (die Linienfolge also geschlossen und einfach ist); andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_IsRing(c db2gse.ST_Curve)
```

Rückgabotyp

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle RING_LINESTRING mit einer einzigen Linienspalte mit dem Namen LN1.

```
CREATE TABLE RING_LINESTRING (ln1 db2gse.ST_LineString)
```

Die folgenden INSERT-Anweisungen fügen drei Linienfolgen in die Spalte LN1 ein. Die erste Zeile enthält eine Linienfolge, die nicht geschlossen ist und somit keinen Ring darstellt. Die zweite Zeile enthält eine Linienfolge, die geschlossen ist und somit einen Ring darstellt. Die dritte Zeile enthält eine geschlossene Linienfolge, die jedoch nicht einfach ist, da sie ihren eigenen Innenbereich schneidet; sie stellt somit keinen Ring dar.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen die Ergebnisse der Funktion ST_IsRing. Die erste und dritte Zeile geben jeweils 0 zurück, weil ihre Linienfolgen keine Ringe sind; die zweite Zeile gibt dagegen 1 zurück, weil es sich dabei um einen Ring handelt.

```
SELECT db2gse.ST_IsRing(1n1) "Ring" FROM RING_LINestring
```

```
Ring
```

```
-----
```

```
0
```

```
1
```

```
0
```

```
3 record(s) selected.
```

ST_IsSimple

ST_IsSimple verwendet ein Geometrieobjekt und gibt 1 (TRUE) zurück, wenn das Objekt einfach ("simple") ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_IsSimple(g db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle ISSIMPLE_TEST mit zwei Spalten. Die Spalte PID, ein smallint-Element, enthält die eindeutige Kennung für jede Zeile. Die Geometriespalte G1 speichert die einfachen und nicht einfachen Geometriemuster.

```
CREATE TABLE ISSIMPLE_TEST (pid smallint, g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen zwei Datensätze in die Tabelle ISSIMPLE_TEST ein. Der erste Datensatz ist einfach, weil es sich um eine Linienfolge handelt, die ihren Innenbereich nicht schneidet. Der zweite Datensatz ist nicht einfach, weil er seinen Innenbereich schneidet.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen die Ergebnisse der Funktion ST_IsSimple. Der erste Datensatz gibt 1 zurück, weil die Linienfolge einfach ist; der zweite Datensatz gibt 0 zurück, weil die Linienfolge nicht einfach ist.

```
SELECT ST_IsSimple(g1)
FROM ISSIMPLE_TEST
```

```
g1
-----
1 0
```

ST_IsValid

ST_IsValid verwendet eine ST_Geometry und gibt 1 (TRUE) zurück, falls es gültig ist; andernfalls wird 0 (FALSE) zurückgegeben. Eine in eine DB2-Datenbank eingefügte Geometrie ist immer gültig, weil der Spatial Extender ihre räumlichen Daten vor dem Akzeptieren immer überprüft. Andere DBMS-Anbieter überprüfen die Eingabe jedoch unter Umständen nicht, sondern erwarten, dass die Anwendung dies tut.

Syntax

```
db2gse.ST_IsValid(g db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Die Tabelle valid_test wird mit den Spalten geotype und g1 erstellt. Die Spalte geotype speichert den Namen der in der Geometriespalte g1 gespeicherten Geometrieunterklasse.

```
CREATE TABLE valid_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Die INSERT-Anweisungen fügen eine Muster-Unterklasse in die Tabelle valid_test ein.

```
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))
)

```

Die Anweisung SELECT listet den Unterklassenamen auf, der in der Spalte geotype mit der Dimension dieses Geotyps gespeichert ist.

```

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 verwendet eine Linienfolge oder eine Mehrlinienfolge und gibt ihre Länge zurück.

Syntax

```
db2gse.ST_Length(c db2gse.ST_Curve)
db2gse.ST_Length(mc db2gse.ST_MultiCurve)
```

Rückgabetyt

Double

Beispiele

Ein Heimatökologe untersucht das Wanderverhalten des Lachsbestands in den Wasserläufen des Landkreises. Der Ökologe möchte die Länge aller Bäche und Flüsse im Landkreis wissen.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle WATERWAYS. Die Spalten ID und NAME kennzeichnen alle in der Tabelle gespeicherten Bäche und Flüsse. Die Spalte WATER ist eine Mehrlinienfolge, da der Fluss und die Fluss-Systeme häufig eine Zusammenfassung verschiedener Linienfolgen sind.

```
CREATE TABLE WATERWAYS (id integer, name varchar(128),
water db2gse.ST_MultiLineString);
```

Die folgenden Anweisung SELECT verwendet die Funktion ST_Length, um den Namen und die Länge jedes Wasserweges im Landkreis zurückzugeben.

```
SELECT name, db2gse.ST_Length(water) "Length"
FROM WATERWAYS;
```

Abb. 36 auf Seite 277 zeigt das Fluss- und Bachsystem innerhalb der Grenzen des Landkreises.

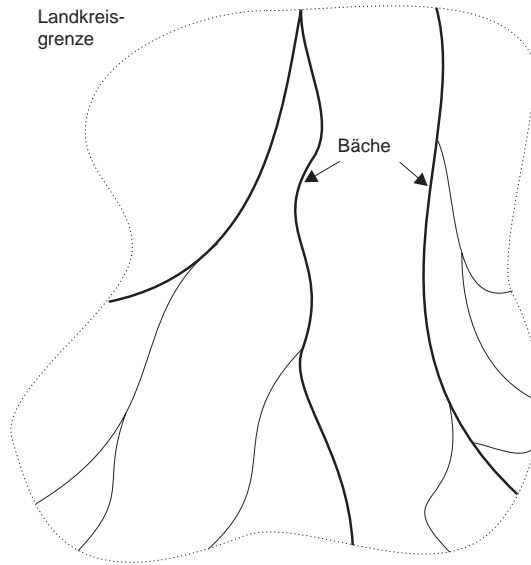


Abbildung 36. Mit `ST_Length` die Gesamtlänge der Wasserwege im Landkreis ermitteln

ST_LineFromText

ST_LineFromText verwendet eine bekannte Textdarstellung des Typs Linienfolge und die Kennung eines räumlichen Bezugssystems und gibt eine Linienfolge zurück.

Syntax

```
db2gse.ST_LineFromText(lineStringTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabetypp

```
db2gse.ST_LineString
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle LINESTRING_TEST mit einer einzigen Linienspalte LN1.

```
CREATE TABLE LINESTRING_TEST (ln1 db2gse.ST_LineString)
```

Die folgende Anweisung INSERT fügt über die Funktion ST_LineFromText eine Linienfolge in die Spalte LN1 ein.

```
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 verwendet eine bekannte binäre Darstellung des Typs Linienfolge und die Kennung eines räumlichen Bezugssystems und gibt eine Linienfolge zurück.

Syntax

```
db2gse.ST_LineFromWKB(WKBLineString Blob(1M), SRID db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_LineString
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle SEWERLINES mit der eindeutigen ID, der Größenklasse und der Geometrie aller Abwasserkanallinien auf.

Die Tabelle SEWERLINES wird mit drei Spalten erstellt. Die erste Spalte, SEWER_ID, kennzeichnet jede Abwasserkanallinie eindeutig. Die zweite Spalte CLASS des Typs Integer kennzeichnet den Typ des Abwasserkanals; normalerweise gibt diese Spalte die Kapazität des Kanals an. Die dritte Spalte, SEWER, des Typs Linienfolge speichert die Geometrie der Abwasserkanallinie.

```
CREATE TABLE SEWERLINES (sewer_id integer,
                          class integer,
                          sewer db2gse.ST_LineString);

/* Create the SQL insert statement to populate the sewer_id, size class
   and the sewer linestring. The question marks are parameter markers that
   indicate the sewer_id, class and sewer geometry values that will be
   retrieved at runtime. */
strcpy (wkb_sql,"insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.ST_LineFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Bind the integer sewer_id value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* Bind the integer class value to the second parameter. */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);
```

```
/* Bind the shape to the third parameter. */  
pcbvalue3 = blob_len;  
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, sewer_wkb, blob_len, &pcbvalue3);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

ST_MLineFromText

ST_MLineFromText verwendet eine bekannte Textdarstellung des Typs Mehrlinienfolge und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrlinienfolge zurück.

Syntax

```
db2gse.ST_MLineFromText(multiLineStringTaggedText String, SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiLineString
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle MLINESTRING_TEST. MLINESTRING_TEST hat zwei Spalten: die smallint-Spalte GID, die die Zeile eindeutig kennzeichnet, und die Mehrlinienspalte ML1.

```
CREATE TABLE ST_MLINESTRING_TEST (gid smallint, ml1 db2gse.ST_MultiLineString)
```

Die folgende Anweisung INSERT fügt die Mehrlinienfolge mit der Funktion ST_MLineFromText ein.

```
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 verwendet eine bekannte binäre Darstellung des Typs Mehrlinienfolge und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrlinienfolge zurück.

Syntax

```
db2gse.ST_MLineFromWKB(WKBMultiLineString Blob(1M), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiLineString
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle WATERWAYS mit einer eindeutigen ID, einem Namen und einer Mehrlinienfolge WATER aus.

Die Tabelle WATERWAYS wird erstellt mit den Spalten ID und NAME, die jeden in der Tabelle gespeicherten Fluss und Wasserlauf kennzeichnen. Die Spalte WATER ist eine Mehrlinienfolge, da der Fluss und die Fluss-Systeme häufig eine Zusammenfassung verschiedener Linienfolgen sind.

```
CREATE TABLE WATERWAYS (id          integer,  
                          name       varchar(128),  
                          water      db2gse.ST_MultiLineString);  
  
/* Create the SQL insert statement to populate the id, name and  
   multilinestring. The question marks are parameter markers that  
   indicate the id, name and water values that will be retrieved at  
   runtime. */  
strcpy (shp_sql, "insert into WATERWAYS (id,name,water)  
values (?,?, db2gse.ST_MLineFromWKB (cast(? as blob(1m)),  
db2gse.coordref())..srid(0))");  
  
/* Allocate memory for the SQL statement handle and associate the  
   statement handle with the connection handle. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Prepare the SQL statement for execution. */  
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);  
  
/* Bind the integer id value to the first parameter. */  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);  
  
/* Bind the varchar name value to the second parameter. */  
pcbvalue2 = name_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,  
SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);
```



```
/* Bind the shape to the third parameter. */  
pcbvalue3 = blob_len;  
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

ST_MPointFromText

ST_MPointFromText verwendet eine bekannte Textdarstellung des Typs Multipoint (Mehrpunkt) und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrpunktangabe zurück.

Syntax

```
db2gse.ST_MPointFromText(multiPointTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_MultiPoint
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle MULTIPOINT_TEST mit einer einzigen Mehrpunktspalte MPT1.

```
CREATE TABLE MULTIPOINT_TEST (mpt1 db2gse.ST_MultiPoint)
```

Die folgende Anweisung INSERT fügt über die Spalte ST_MPointFromText eine Mehrpunktangabe in die Spalte MPT1 ein.

```
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 verwendet eine bekannte binäre Darstellung des Typs Multipoint (Mehrpunkt) und die Kennung eines räumlichen Bezugssystems und gibt eine Mehrpunktangabe zurück.

Syntax

```
db2gse.ST_MPointFromWKB(WKBMultiPoint Blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_MultiPoint
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle SPECIES_SITINGS aus.

Die Tabelle SPECIES_SITINGS wird mit drei Spalten erstellt. Die Spalten SPECIES und GENUS kennzeichnen jede Zeile eindeutig, während die Mehrpunktangabe SITINGS die Standorte der Spezies speichert.

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                               genus varchar(32),
                               sitings db2gse.ST_MultiPoint);

/* Create the SQL insert statement to populate the species, genus and
   sitings. The question marks are parameter markers that
   indicate the species, genus and sitings values that will be retrieved at
   runtime. */
strcpy (wkb_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.ST_MPointFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Bind the varchar species value to the first parameter. */
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, species_len, 0, &species, species_len, &pcbvalue1);
/* Bind the varchar genus value to the second parameter. */
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, genus_len, 0, &name, genus_len, &pcbvalue2);

/* Bind the shape to the third parameter. */
pcbvalue3 = sitings_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, sitings_len, 0, sitings_wkb, sitings_len, &pcbvalue3);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

ST_MPolyFromText

ST_MPolyFromText verwendet eine bekannte Textdarstellung des Typs Multipolygon und die Kennung eines räumlichen Bezugssystems und gibt ein Multipolygon zurück.

Diese Funktion kann kein Multipolygon als Eingabe verwenden, das mehrere Polygone mit denselben Koordinaten beinhaltet.

Syntax

```
db2gse.ST_MPolyFromText(multiPolygonTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiPolygon
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle MULTIPOLYGON_TEST mit einer einzigen Multipolygonspalte MPL1.

```
CREATE TABLE MULTIPOLYGON_TEST (mpl1 db2gse.ST_MultiPolygon)
```

Die folgende Anweisung INSERT fügt über die Funktion ST_MPolyFromText ein Multipolygon in die Spalte MPL1 ein.

```
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 verwendet eine bekannte binäre Darstellung des Typs Multipolygon und die Kennung eines räumlichen Bezugssystems und gibt ein Multipolygon zurück.

Syntax

```
db2gse.ST_MPolyFromWKB(WKBMultiPolygon Blob(1M), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_MultiPolygon
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle LOTS aus.

Die Tabelle LOTS speichert die LOT_ID, die jedes Grundstück eindeutig kennzeichnet und das Multipolygon für das Grundstück, die die Geometrie für die Grundstückslinie enthält.

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );  
  
/* Create the SQL insert statement to populate the lot_id, and lot. The  
   question marks are parameter markers that indicate the lot_id, and lot  
   values that will be retrieved at runtime. */  
strcpy (wkb_sql,"insert into LOTS (lot_id,lot)  
values (?, db2gse.ST_MPolyFromWKB (cast(? as blob(1m)),  
db2gse.coordref(..srid(0)))");  
  
/* Allocate memory for the SQL statement handle and associate the  
   statement handle with the connection handle. */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* Prepare the SQL statement for execution. */  
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);  
  
/* Bind the lot_id integer value to the first parameter. */  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);  
  
/* Bind the lot shape to the second parameter. */  
pcbvalue2 = lot_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
SQL_BLOB, lot_len, 0, lot_wkb, lot_len, &pcbvalue2);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

ST_NumGeometries

ST_NumGeometries verwendet eine Gruppe und gibt die Anzahl der Geometrien in der Gruppe zurück.

Syntax

```
db2gse.ST_NumGeometries(g db2gse.ST_GeomCollection)
```

Rückgabetyt

Integer

Beispiele

Der Stadtplaner muss die tatsächliche Anzahl der Einzelgebäude zu jeder Gebäudegrundfläche kennen.

Die Gebäudegrundflächen sind in der Tabelle BUILDINGFOOTPRINTS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wurde.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
lot_id integer,  
footprint db2gse.ST_MultiPolygon);
```

Die folgende Anweisung SELECT verwendet die Funktion ST_NumGeometries zum Auflisten der BUILDING_ID, die jedes Gebäude und die Anzahl der Gebäude für jede Grundfläche eindeutig kennzeichnet.

```
SELECT building_id, db2gse.ST_NumGeometries (footprint) "Number of buildings"  
FROM BUILDINGFOOTPRINTS;
```

ST_NumInteriorRing

ST_NumInteriorRing verwendet ein Polygon und gibt die Anzahl seiner inneren Ringe zurück.

Syntax

```
db2gse.ST_NumInteriorRing(p db2gse.ST_Polygon)
```

Rückgabetyt

Integer

Beispiele

Eine Ornithologin, die den Vogelbestand auf verschiedenen Südseeinseln untersuchen möchte, weiß, dass sich die Zone für die Nahrungssuche einer bestimmten Spezies auf die Inseln mit Süßwasserseen beschränkt. Sie möchte daher wissen, welche Inseln ein oder mehrere Seen haben.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle ISLANDS. Die Spalten ID und NAME der Tabelle ISLANDS kennzeichnen alle Inseln, und die Polygonspalte LAND speichert die Geometrie dieser Inseln.

```
CREATE TABLE ISLANDS (id integer, name varchar(32), land db2gse.ST_Polygon);
```

Da die inneren Ringe die Seen darstellen, werden mit der Funktion ST_NumInteriorRing nur die Inseln aufgelistet, die mindestens einen inneren Ring haben.

```
SELECT name FROM ISLANDS WHERE db2gse.ST_NumInteriorRing(land) > 0;
```

ST_NumPoints

ST_NumPoints verwendet eine Linienfolge und gibt die Anzahl ihrer Punkte zurück.

Syntax

```
db2gse.ST_NumPoints(l db2gse.ST_LineString)
```

Rückgabetyt

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle NUMPOINTS_TEST. Die Spalte GEOTYPE enthält den in der Geometriespalte G1 gespeicherten Geometrietyp.

```
CREATE TABLE NUMPOINTS_TEST (geotype varchar(12), g1 db2gse.ST_Geometry)
```

Die folgende Anweisung INSERT fügt eine Linienfolge ein.

```
INSERT INTO NUMPOINTS_TEST VALUES( linestring,  
db2gse.ST_LineFromText('linestring (10.02 20.01, 23.73 21.92)',  
db2gse.coordref()..srid(0)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe listet den Geometrietyp und die Anzahl der darin enthaltenen Punkte auf.

```
SELECT geotype, db2gse.ST_NumPoints(g1)  
FROM NUMPOINTS_TEST
```

```
GEOTYPE      Number of points  
-----  
ST_linestring      2
```

1 record(s) selected.

ST_OrderingEquals

ST_OrderingEquals vergleicht zwei Geometrien und gibt 1 (TRUE) zurück, wenn die Geometrien gleich sind und die Koordinaten die gleiche Reihenfolge haben; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_OrderingEquals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabebetyp

Integer

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle LINESTRING_TEST mit den beiden Linienspalten L1 und L2.

```
CREATE TABLE LINESTRING_TEST (lid integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString);
```

Die folgende Anweisung INSERT fügt zwei Linienfolgen ein in L1 und L2, die gleich sind und dieselbe Koordinatenreihenfolge haben.

```
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)));
```

Die folgende Anweisung INSERT fügt zwei Linienfolgen ein in L1 und L2, die gleich sind, aber nicht dieselbe Koordinatenreihenfolge haben.

```
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)));
```

Wie die folgende SELECT-Anweisung und die Ergebnismenge zeigen, gibt die Funktion ST_OrderingEquals folgende Werte zurück:

- 1 (TRUE), wenn die als Eingabe verwendeten Geometrien übereinstimmen und ihre Koordinaten dieselbe Reihenfolge aufweisen
- 0 (FALSE), wenn die als Eingabe verwendeten Geometrien nicht übereinstimmen oder wenn die Koordinaten einer Geometrie eine andere Reihenfolge aufweisen als die der anderen Geometrie

```
SELECT lid, db2gse.ST_OrderingEquals(l1,l2) OrderingEquals FROM linestring_test  
lid OrderingEquals
```

```
---      -----  
1         1  
2         0
```

ST_Overlaps

ST_Overlaps verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn die Schnittmenge der beiden Objekte ein Geometrieobjekt derselben Dimension ergibt, das jedoch mit keinem der Quellenobjekte identisch ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Overlaps(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabetyt

Integer

Beispiele

Der zuständige Planer benötigt eine Liste der Abfall-Gefahrenstandorte, in deren Umkreis von fünf Meilen sensible Bereiche wie Schulen, Krankenhäuser und Pflegeheime liegen.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle SENSITIVE_AREAS. Die Tabelle SENSITIVE_AREAS enthält verschiedene Spalten, die die gefährdeten Institutionen beschreiben, zusätzlich zu der Spalte ZONE, die die Polygoneometrie der Institution beschreibt.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

In der Tabelle HAZARDOUS_SITES ist die Kennung der Standorte in den Spalten SITE_ID und NAME gespeichert; der tatsächliche geografische Standort wird jeweils in der Punktspalte LOCATION gespeichert.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

In der folgenden Anweisung SELECT werden die Tabellen SENSITIVE_AREAS and HAZARDOUS_SITES über die Funktion ST_Overlaps verknüpft. Die Funktion gibt 1 (TRUE) zurück für alle Zeilen in der Tabelle SENSITIVE_AREAS, deren Zonenpolygone den gepufferten Fünf-Meilen-Radius um den HAZARDOUS_SITES-Standort überlappen.

```
SELECT hs.name
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa WHERE db2gse.ST_Overlaps
(buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

In Abb. 37 auf Seite 293 überlappen das Krankenhaus und die Schule den Fünf-Meilen-Radius der beiden Abfallstandorte des Landkreises; das Pflegeheim liegt außerhalb dieses Radius.

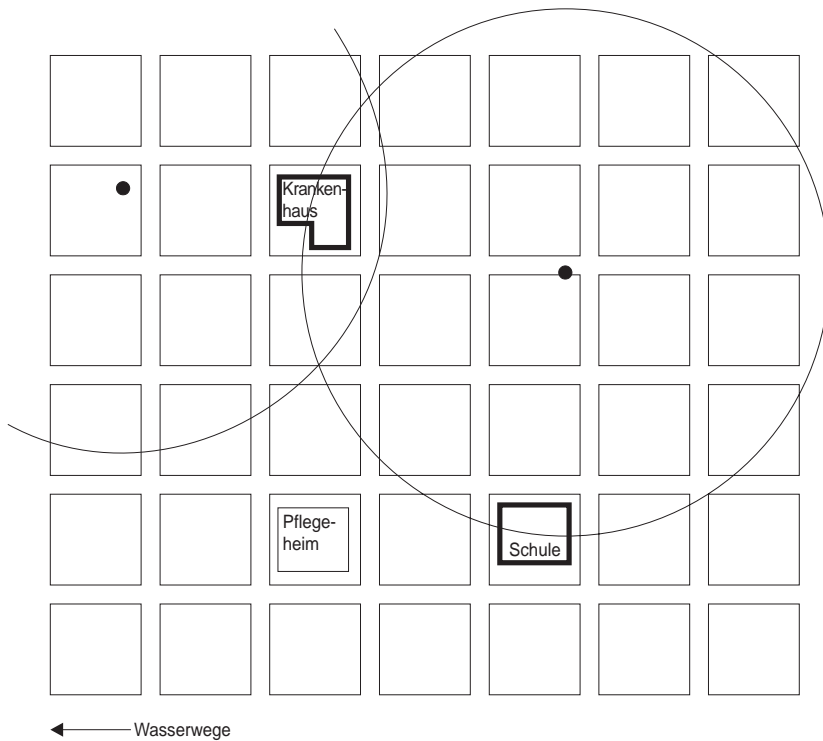


Abbildung 37. Mit `ST_Overlaps` die Gebäude ermitteln, die zumindest teilweise im Gefährdungsbereich der Abfall-Lagerstätten liegen

ST_Perimeter

ST_Perimeter gibt den Umfang eines Polygons zurück.

Syntax

```
db2gse.ST_Perimeter(p db2gse.ST_Polygon)
```

Rückgabetyt

Double

Beispiele

Ein Ornithologe, der das Verhalten der Vögel an den Ufern von Seen studiert, muss die Uferlinie der Seen innerhalb eines bestimmten Gebiets kennen. Die Seen sind als Polygone in der Tabelle WATERBODIES gespeichert; diese Tabelle wurde mit der folgenden Anweisung CREATE TABLE erstellt.

```
CREATE TABLE WATERBODIES (wbid integer,  
                           waterbody db2gse.ST_MultiPolygon);
```

In der folgenden Anweisung SELECT gibt die Funktion ST_Perimeter den Umfang jeder Wasserfläche zurück; die Funktion SUM akkumuliert die Umfangswerte und gibt ihre Summe zurück.

```
SELECT SUM(db2gse.ST_Perimeter(waterbody))  
FROM waterbodies;
```

ST_Point

ST_Point gibt einen ST_Point zurück; als Eingabe verwendet die Funktion eine X-Koordinate, eine Y-Koordinate und einen räumlichen Bezug.

Syntax

```
db2gse.ST_Point(X Double, Y Double, SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle POINT_TEST mit einer einzigen Punktspalte PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

Die Funktion ST_Point wandelt die Punktkoordinaten in eine Punktgeometrie um, bevor die Anweisung INSERT sie in die Spalte PT1 einfügt.

```
INSERT INTO point_test VALUES(  
    db2gse.ST_Point(10.01,20.03, db2gse.coordref()..srid(0))  
)
```

ST_PointFromText

ST_PointFromText verwendet eine bekannte Textdarstellung des Typs Punkt und die Kennung eines räumlichen Bezugssystems und gibt einen Punkt zurück.

Syntax

```
db2gse.ST_PointFromText(pointTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle POINT_TEST mit einer einzigen Punktspalte PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

Bevor die Anweisung INSERT den Punkt in die Spalte PT1 einfügt, wandelt die Funktion ST_PointFromText die Punkttextkoordinaten in das Punktformat um.

```
INSERT INTO POINT_TEST VALUES (  
    db2gse.ST_PointFromText ('point(10.01 20.03)',  
    db2gse.coordref()..srid(0))
```

ST_PointFromWKB

ST_PointFromWKB verwendet eine bekannte binäre Darstellung des Typs Punkt und die Kennung eines räumlichen Bezugssystems und gibt einen Punkt zurück.

Syntax

```
db2gse.ST_PointFromWKB(WKBPoint Blob(1M), SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle HAZARDOUS_SITES aus.

Die Gefahrenstandorte sind in der Tabelle HAZARDOUS_SITES gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wird. Die als Punkt definierte Spalte LOCATION speichert einen Standort, der das geografische Zentrum jedes Gefahrenstandorts darstellt.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);

/* Create the SQL insert statement to populate the site_id, name and
   location. The question marks are parameter markers that indicate the
   site_id, name and location values that will be retrieved at runtime. */

strcpy (wkb_sql,"insert into HAZARDOUS_SITES (site_id, name, location)
values (?,?, db2gse.ST_PointFromWKB(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Bind the site_id integer value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* Bind the name varchar value to the second parameter. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);
```

```
/* Bind the location shape to the third parameter. */  
pcbvalue3 = location_len;  
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, location_len, 0, location_wkb, location_len, &pcbvalue3);  
  
/* Execute the insert statement. */  
rc = SQLExecute (hstmt);
```

ST_PointN

ST_PointN verwendet eine Linienfolge und einen ganzzahligen Index und gibt einen Punkt zurück, der den n-ten Scheitelpunkt im Verlauf der Linienfolge darstellt.

Syntax

```
db2gse.ST_PointN(l db2gse.ST_Curve, n Integer)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle POINTN_TEST mit zwei Spalten: der Spalte GID, die jede Zeile eindeutig kennzeichnet, und der Linienspalte LN1.

```
CREATE TABLE POINTN_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Die folgenden INSERT-Anweisungen fügen zwei Linienwerte ein. Die erste Zeile hat keine Z-Koordinaten oder Maßangaben; die zweite dagegen schon.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe listet die Spalte GID und den zweiten Scheitelpunkt jeder Linienfolge auf. Die erste Zeile gibt einen Punkt zurück, der weder Z-Koordinate noch ein Maß enthält; der aus der zweiten Zeile resultierende Punkt enthält eine Z-Koordinate und ein Maß. Die Funktion ST_PointN gibt Punkte mit einer Z-Koordinate oder einem Maß zurück, falls diese in der Quellenlinie enthalten sind.

```
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

ST_PointOnSurface verwendet ein Polygon oder ein Multipolygon und gibt einen ST_Point zurück.

Syntax

```
db2gse.ST_PointOnSurface(s db2gse.ST_Surface)
db2gse.ST_PointOnSurface(ms db2gse.ST_MultiSurface)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Der Stadtplaner muss einen Kennzeichnungspunkt für jede Gebäudegrundfläche erstellen.

Die Gebäudegrundflächen sind in der Tabelle BUILDINGFOOTPRINTS gespeichert, die mit der folgenden Anweisung CREATE TABLE erstellt wurde.

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

Die Funktion ST_PointOnSurface generiert einen Punkt, der garantiert auf der Gebäudegrundfläche liegt.

```
SELECT db2gse.ST_PointOnSurface(footprint) FROM BUILDINGFOOTPRINTS;
```

ST_PolyFromText

ST_PolyFromText verwendet eine bekannte Textdarstellung des Typs Polygon und die Kennung eines räumlichen Bezugssystems und gibt ein Polygon zurück.

Syntax

```
db2gse.ST_PolyFromText(polygonTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_Polygon
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle POLYGON_TEST mit einer einzigen Polygonspalte.

```
CREATE TABLE POLYGON_TEST (p11 db2gse.ST_Polygon)
```

Die folgende Anweisung INSERT fügt über die Funktion ST_PolyFromText ein Polygon in die Polygonspalte ein.

```
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 verwendet eine bekannte binäre Darstellung des Typs Polygon und die Kennung eines räumlichen Bezugssystems und gibt ein Polygon zurück.

Syntax

```
db2gse.ST_PolyFromWKB(WKBPolygon Blob(1M), SRID db2gse.coordref)
```

Rückgabotyp

```
db2gse.ST_Polygon
```

Beispiele

Der folgende Codeabschnitt füllt die Tabelle SENSITIVE_AREAS aus.

Die Tabelle SENSITIVE_AREAS enthält verschiedene Spalten, die die gefährdeten Institutionen beschreiben, zusätzlich zu der Spalte zone, die die Polygoneometrie der Institution beschreibt.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);

/* Create the SQL insert statement to populate the id, name, size, type and
   zone. The question marks are parameter markers that indicate the id,name,
   size, type and zone values that will be retrieved at runtime. */
strcpy (shp_wkb,"insert into SENSITIVE_AREAS (id, name, size, type, zone)
values (?,?,,?, db2gse.ST_PolyFromWKB (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */
rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Bind the id integer value to the first parameter. */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);

/* Bind the name varchar value to the second parameter. */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* Bind the size float to the third parameter. */
pcbvalue3 = 0;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
SQL_REAL, 0, 0, &size, 0, &pcbvalue3);
```

```
/* Bind the type varchar to the fourth parameter. */
pcbvalue4 = type_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_CHAR,
    SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);

/* Bind the zone polygon to the fifth parameter. */
pcbvalue5 = zone_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, zone_len, 0, zone_wkb, zone_len, &pcbvalue5);

/* Execute the insert statement. */
rc = SQLExecute (hstmt);
```

ST_Polygon

ST_Polygon generiert ein ST_Polygon aus einem ST_LineString.

Syntax

```
db2gse.ST_Polygon(l db2gse.ST_LineString)
```

Rückgabetyt

```
db2gse.ST_Polygon
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle POLYGON_TEST mit einer einzigen Spalte, P1.

```
CREATE TABLE POLYGON_TEST (p1 db2gse.ST_polygon)
```

Die folgende Anweisung INSERT wandelt einen Ring (eine geschlossene, einfache Linienfolge) in ein Polygon um und fügt Sie mit der Funktion ST_LineFromText innerhalb der Funktion ST_Polygon in die Spalte P1 ein.

```
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 vergleicht zwei Geometrien und gibt 1 (TRUE) zurück, wenn die Geometrien die von der Mustermatrix-Zeichenfolge DE-9IM definierten Bedingungen erfüllen; andernfalls wird 0 (FALSE) zurückgegeben. Informationen zu der DE-9IM-Mustermatrix finden Sie im Abschnitt „Prädikatfunktionen“ auf Seite 173.

Syntax

```
db2gse.ST_Relate(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry, pattern-  
Matrix CHAR(9))
```

Rückgabetyt

Integer

Beispiele

Eine DE-9IM-Mustermatrix ist eine Einheit zum Vergleich von Geometrien. Es gibt verschiedene Arten solcher Matrizen. Die Mustermatrix *equals* gibt beispielsweise an, ob zwei Geometrien gleich sind.

In diesem Beispiel wird eine "equals"-Mustermatrix (siehe Tabelle 57) von links nach rechts und von oben nach unten in eine Zeichenfolge („T*F**FFF*“) eingelesen.

Tabelle 57. Equals-Mustermatrix

		b	
	Innenbereich	Begrenzung	Außenbereich
a	Innenbereich T	*	F
	Begrenzung *	*	F
	Außenbereich F	F	*

Als nächstes wird die Tabelle RELATE_TEST mit der folgenden Anweisung CREATE TABLE erstellt.

```
CREATE TABLE RELATE_TEST (rid integer, g1 db2gse.ST_Geometry,  
g2 db2gse.ST_Geometry, g3 db2gse.ST_Geometry);
```

Die folgenden INSERT-Anweisungen fügen eine Muster-Unterklasse in die Tabelle RELATE_TEST ein.

```
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)  
)
```

Die folgende Anweisung `SELECT` und die entsprechende Ergebnisgruppe listet den Unterklassennamen auf, der in der Spalte `GEOTYPE` mit der Dimension dieses Geotyps gespeichert ist.

```
SELECT rid, relate(g1,g2, 'T**FFF*') equals FROM relate_test
```

```
RID      equals
-----  -----
1        1
```

1 record(s) selected.

ST_SRID

ST_SRID verwendet ein Geometrieobjekt und gibt die Kennung seines räumlichen Bezugssystems zurück.

Syntax

```
db2gse.ST_SRID(g1 db2gse.ST_Geometry)
```

Rückgabebetyp

Integer

Beispiele

Die Funktion ST_SRID gibt die Kennung des räumlichen Bezugssystems zurück, das dem ST_Geometry-Wert zugeordnet ist.

Ein Geometrietyp wird beispielweise in einer CREATE TABLE-Anweisung verwendet:

```
CREATE TABLE SRID_TEST(g1 db2gse.ST_Geometry)
```

In der folgenden Anweisung INSERT wird eine Punktgeometrie an der Koordinatenposition 10.01,50.76 in die Geometriespalte G1 eingetragen. Wenn die Punktgeometrie von der Funktion ST_PointFromText erstellt wurde, so hat sie den srid-Wert 1 erhalten.

```
INSERT INTO SRID_TEST  
VALUES (db2gse.ST_PointFromText('point(10.01 50.76)',  
db2gse.coordref(..srid(0)))
```

Die Funktion ST_SRID gibt die Kennung des räumlichen Bezugssystems der gerade eingegebenen Geometrie zurück, wie die folgende SELECT-Anweisung und die entsprechende Ergebnismenge verdeutlichen.

```
SELECT db2gse.ST_SRID(g1) FROM SRID_TEST
```

```
g1  
-----  
0
```

ST_StartPoint

ST_StartPoint verwendet eine Linienfolge und gibt einen Punkt zurück, der den ersten Punkt der Linienfolge darstellt.

Syntax

```
db2gse.ST_StartPoint(c db2gse.ST_Curve)
```

Rückgabetyt

```
db2gse.ST_Point
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle STARTPOINT_TEST. STARTPOINT_TEST enthält zwei Spalten: die Spalte GID, die die Zeilen der Tabelle eindeutig kennzeichnet, und die Linienspalte LN1.

```
CREATE TABLE STARTPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Die folgenden INSERT-Anweisungen fügen die Linienfolgen in die Spalte LN1 ein. Die erste Zeile hat keine Z-Koordinaten oder Maßangaben; die zweite dagegen schon.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe zeigen, wie die Funktion ST_StartPoint den ersten Punkt jeder Linienfolge extrahiert. Die Funktion ST_AsText wandelt den Punkt in sein Textformat um. Der erste Punkt in der Liste hat keine Z-Koordinate und kein Maß; der zweite Punkt hat beides, weil diese Elemente auch in der Quellenlinie enthalten sind.

```
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

ST_SymmetricDiff verwendet zwei Geometrieobjekte und gibt ein Geometrieobjekt zurück, das die symmetrische Differenz der Eingabeobjekte darstellt.

Die Funktion ST_SymmetricDiff gibt die symmetrische Differenz (das Boolesche logische XOR des Raums) zweier sich schneidender Geometrien zurück, die dieselbe Dimension aufweisen. Wenn diese Geometrien übereinstimmen, gibt ST_SymmetricDiff eine leere Geometrie zurück. Wenn sie nicht übereinstimmen, liegt ein Abschnitt einer oder beider Geometrien außerhalb des Schnittbereichs. ST_SymmetricDiff gibt den sich nicht schneidenden Abschnitt bzw. die sich nicht schneidenden Abschnitte als Gruppe, z. B. als Multipolygon zurück.

Wenn ST_SymmetricDiff Geometrien mit unterschiedlicher Dimension als Eingabe erhält, gibt die Funktion eine Null zurück.

Syntax

```
db2gse.ST_SymmetricDiff(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Der zuständige Planer muss den Bereich der sensiblen Bereiche und des Fünf-Meilen-Radius ohne Schnittmenge ermitteln.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle SENSITIVE_AREAS, die verschiedene Spalten zur Beschreibung der gefährdeten Institutionen enthält. Die Tabelle SENSITIVE_AREAS enthält auch die Spalte ZONE, in der die Polygeometrie der Institution gespeichert ist.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

Die folgende Anweisung CREATE TABLE erstellt die Tabelle HAZARDOUS_SITES, in der die Kennung der Standorte in den Spalten SITE_ID und NAME gespeichert wird; der tatsächliche geografische Standort jeder Site wird in der Punktspalte LOCATION gespeichert.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name      varchar(128),
                               location  point);
```

Die Funktion ST_Buffer generiert einen Fünf-Meilen-Puffer um die Standorte der Lagerstätten für gefährlichen Abfall. Die Funktion ST_SymmetricDiff generierte Polygone aus der Schnittmenge der gepufferten Polygone um die Abfallstandorte und der sensiblen Bereiche. Die Funktion ST_Area gibt die Polygonfläche der Schnittmenge für jeden Gefahrenstandort zurück.

```
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
```

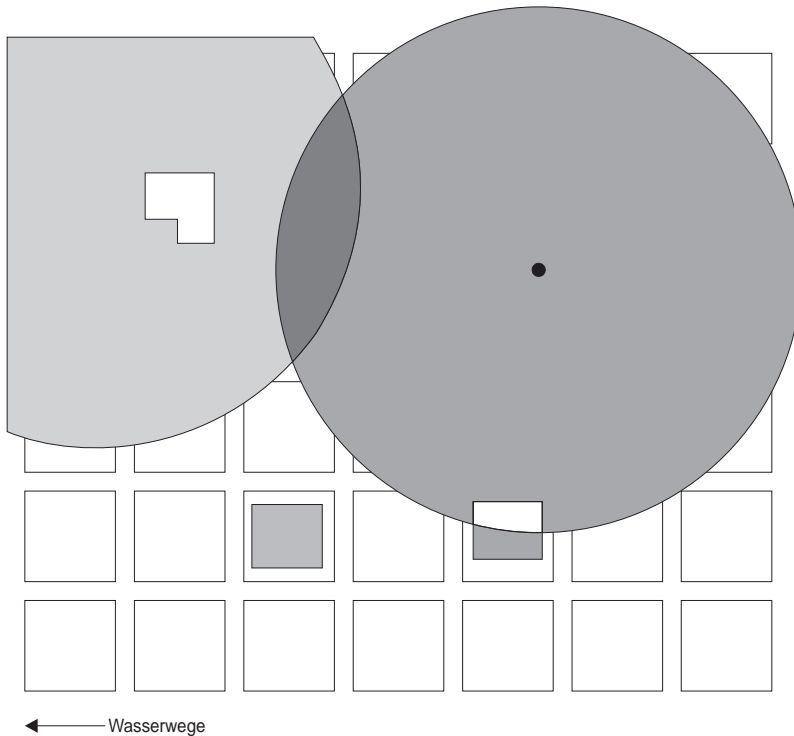


Abbildung 38. Mit ST_SymmetricDiff die Gefahrenbereiche ermitteln, die keine sensiblen Bereiche (bewohnte Gebäude) enthalten

In Abb. 38 führt die symmetrische Differenz aus Gefahrengebieten und den sensiblen Bereichen zur Subtraktion der Schnittbereiche.

ST_Touches

ST_Touches gibt 1 (TRUE) zurück, wenn keiner der gemeinsamen Punkte der beiden Geometrien den Innenbereich beider Geometrien schneidet; andernfalls wird 0 (FALSE) zurückgegeben. Mindestens eine der Geometrien muss eine Linienfolge, ein Polygon, eine Mehrlinienfolge oder ein Multipolygon sein.

Syntax

```
db2gse.ST_Touches(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabotyp

Integer

Beispiele

Der GIS-Techniker muss eine Liste aller Abwasserkanäle, deren Endpunkt einen anderen Abwasserkanal schneiden, zur Verfügung stellen.

Die folgende Anweisung CREATE TABLE erstellt die Tabelle SEWERLINES mit drei Spalten. Die erste Spalte, SEWER_ID, kennzeichnet jede Abwasserkanallinie eindeutig. Die zweite Spalte CLASS des Typs Integer kennzeichnet den Typ des Abwasserkanals; normalerweise gibt diese Spalte die Kapazität des Kanals an. Die dritte Spalte, SEWER, des Typs Linienfolge speichert die Geometrie der Abwasserkanallinie.

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer, sewer  
db2gse.ST_LineString);
```

Die folgende Anweisung SELECT gibt eine sortierte Liste der SEWER_IDS, die einander berühren, zurück.

```
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 ordnet eine Geometrie einem anderen räumlichen Bezugssystem zu als dem momentan zugeordneten.

Syntax

```
db2gse.ST_Transform(g db2gse.ST_Geometry, SRID db2gse.coordref)
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle TRANSFORM_TEST mit den beiden Linienspalten L1 und L2.

```
CREATE TABLE TRANSFORM_TEST (tid integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString)
```

Die folgende Anweisung INSERT fügt eine Linienfolge mit einer SRID von 102 in l1 ein.

```
INSERT INTO TRANSFORM_TEST VALUES (1, db2gse.ST_LineFromText('linestring  
(10.01 40.43, 92.32 29.89)',  
db2gse.coordref()..srid(102)),NULL)
```

Die Funktion ST_Transform wandelt die Linienfolge von L1 aus dem der SRID 102 zugeordneten Koordinatenbezug in den der SRID 105 zugeordneten Koordinatenbezug um. Die folgende Anweisung UPDATE speichert die umgewandelte Linienfolge in Spalte l2.

```
UPDATE TRANSFORM_TEST SET l2 = db2gse.ST_Transform(l1,  
db2gse.coordref()..srid(105))
```

Wenn die zugeordnete Geometrie nicht innerhalb des Koordinatensystems liegt, das dem neuen räumlichen Bezugssystem zugrunde liegt, gibt ST_Transform die Geometrie als Null zurück.

Gehen Sie z. B. von einer ST_Point-Geometrie mit der X-Koordinate 10.01 und der Y-Koordinate 20.02 aus. Nehmen Sie an, dass diese Geometrie einem räumlichen Bezugssystem mit den folgenden Parametern zugeordnet ist:

falsex	0
falsey	0
xyunits	1

Nehmen Sie weiterhin an, dass Sie `ST_Transform` aufrufen, um das räumliche Bezugssystem der `ST_Point`-Geometrie durch ein anderes mit den folgenden Parametern zu ersetzen:

<code>falsex</code>	100
<code>falsey</code>	100
<code>xyunits</code>	1

`ST_Transform` gibt die Geometrie als Null zurück, da die Koordinaten (10.01, 20.02) außerhalb des Koordinatensystems liegen, das dem neuen räumlichen Bezugssystem zugrunde liegt.

ST_Union

ST_Union verwendet zwei Geometrieobjekte und gibt ein Geometrieobjekt zurück, das die Vereinigung der Eingabeobjekte darstellt.

Syntax

```
db2gse.ST_Union(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabotyp

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle SENSITIVE_AREAS, die verschiedene Spalten zur Beschreibung der gefährdeten Institutionen enthält. Die Tabelle SENSITIVE_AREAS enthält auch die Spalte ZONE, in der die Polygoneometrie der Institution gespeichert ist.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

Die folgende Anweisung CREATE TABLE erstellt die Tabelle HAZARDOUS_SITES, in der die Kennung der Standorte in den Spalten SITE_ID und NAME gespeichert wird. Der tatsächliche geografische Standort jeder Site wird in der Punktspalte LOCATION gespeichert.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer, name varchar(128),
                               location db2gse.ST_Point);
```

Die folgende Anweisung SELECT verwendet die Funktion ST_Buffer zum Generieren eines Fünf-Meilen-Puffers um die Standorte der Lagerstätten für gefährlichen Abfall. Die Funktion ST_Union generierte Polygone aus der Vereinigungsmenge der gepufferten Polygone um die Abfallstandorte und der sensiblen Bereiche. Die Funktion ST_Area gibt die Vereinigung der Polygonbereiche zurück.

```
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 verwendet zwei Geometrieobjekte und gibt 1 (TRUE) zurück, wenn das erste Objekt im zweiten vollständig enthalten ist; andernfalls wird 0 (FALSE) zurückgegeben.

Syntax

```
db2gse.ST_Within(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Rückgabebetyp

Integer

Beispiele

Im nachfolgenden Beispiel werden zwei Tabellen erstellt. Die erste Tabelle, BUILDINGFOOTPRINTS, enthält die Gebäudegrundflächen einer Stadt. Die zweite Tabelle, LOTS, enthält die Grundstücke der Stadt. Der Stadtplaner möchte sicherstellen, dass alle Gebäudegrundflächen vollständig auf dem jeweiligen Grundstück liegen.

In beiden Tabellen speichert der Datentyp Multipolygon die Geometrie der Gebäudegrundflächen und der Grundstücke. Der Datenbank-Designer hat für beide Merkmale Multipolygone ausgewählt, da die Grundstücke durch natürliche Merkmale wie beispielsweise Flüsse unterbrochen sein können und da die Gebäudegrundflächen häufig mehrere Gebäude umfassen.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
lot_id integer,  
footprint db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );
```

Mit der folgenden Anweisung SELECT wählt der Stadtplaner zunächst die Gebäude aus, die nicht vollständig auf einem Grundstück liegen.

```
SELECT building_id  
FROM BUILDINGFOOTPRINTS, LOTS  
WHERE db2gse.ST_Within(footprint,lot) ≠ 1;
```

Die erste Abfrage liefert zwar eine Liste aller BUILDING_IDS, deren Gebäudegrundfläche außerhalb eines Grundstückspolygons liegt; sie ermittelt jedoch nicht, ob den übrigen Gebäuden die richtige lot_id zugeordnet wurde. Diese zweite SELECT-Anweisung führt eine Datenintegritätsprüfung mit der Spalte LOT_ID der Tabelle BUILDINGFOOTPRINTS aus.

```
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_WKBToSQL

ST_WKBToSQL konstruiert einen Wert ST_Geometry anhand der bekannten binären Darstellung. Der SRID-Wert Null wird automatisch verwendet.

Syntax

```
db2gse.ST_WKBToSQL(WKBGeometry Blob(1M))
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle LOTS mit zwei Spalten: Die Spalte LOT_ID, die jedes Grundstück eindeutig kennzeichnet, und die Multipolygonspalte LOT, die die Geometrie jedes Grundstücks enthält.

```
CREATE TABLE lots (lot_id    integer,
                   lot      db2gse.ST_MultiPolygon);
```

Der folgende C-Codeausschnitt enthält ODBC-Funktionen, die in SQL-Funktionen des Spatial Extender eingebettet sind; diese Funktionen fügen Daten in die Tabelle LOTS ein.

Die Funktion ST_WKBToSQL wandelt WKB-Darstellungen in Spatial Extender-Geometrien um. Die gesamte Anweisung INSERT wird in eine Zeichenfolge wkb_sql kopiert. Die Anweisung INSERT enthält Parametermarken zum dynamischen Akzeptieren der LOT_ID-Daten und der LOT-Daten.

```
/* Create the SQL insert statement to populate the lot id and the
   lot multipolygon. The question marks are parameter markers that
   indicate the lot_id and lot values that will be retrieved at
   run time. */

strcpy (wkb_sql,"insert into lots (lot_id, lot)
       values(?, db2gse.ST_WKBToSQL(cast(? as blob(1m))))");

/* Allocate memory for the SQL statement handle and associate the
   statement handle with the connection handle. */

rc = SQLAllocStmt (handle, &hstmt);

/* Prepare the SQL statement for execution. */

rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* Bind the integer key value to the first parameter. */

pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
                      SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);
```

```
/* Bind the shape to the second parameter. */  
  
pcbvalue2 = blob_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
  
/* Execute the insert statement. */  
  
rc = SQLExecute (hstmt);
```

ST_WKTTToSQL

ST_WKTTToSQL konstruiert einen Wert ST_Geometry anhand der bekannten Textdarstellung. Der SRID-Wert Null wird automatisch verwendet.

Syntax

```
db2gse.ST_WKTTToSQL(geometryTaggedText Varchar(4000))
```

Rückgabetyt

```
db2gse.ST_Geometry
```

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle GEOMETRY_TEST mit zwei Spalten: die Spalte GID des Typs Integer, die jede Zeile eindeutig kennzeichnet, und die Spalte G1, in der die Geometrie gespeichert wird.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Die folgenden INSERT-Anweisungen fügen die Daten in die Spalten GID und G1 der Tabelle GEOMETRY_TEST ein. Die Funktion ST_WKTTToSQL wandelt die Textdarstellung jeder Geometrie in die entsprechende Spatial Extender-Exemplarunterklasse um.

```
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 verwendet einen Punkt und gibt seine X-Koordinate zurück.

Syntax

```
ST_X(p ST_Point)
```

Rückgabetyt

Double

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle X_TEST mit zwei Spalten: der Spalte GID, die die Zeile eindeutig kennzeichnet, und der Punktspalte PT1.

```
CREATE TABLE X_TEST (gid integer, pt1 db2gse.ST_Point)
```

Die folgenden INSERT-Anweisungen fügen zwei Zeilen ein. Eine der Zeilen ist ein Punkt ohne Z-Koordinate oder Maß. Die andere Spalte hat eine Z-Koordinate und ein Maß.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe listet die Spalte GID und die Double-X-Koordinate der Punkte auf.

```
SELECT gid, db2gse.ST_X(pt1) "The X coordinate" FROM X_TEST
```

GID	The X coordinate
1	+1.0020000000000000E+001
2	+1.0020000000000000E+001

2 record(s) selected.

ST_Y

ST_Y verwendet einen Punkt und gibt seine Y-Koordinate zurück.

Syntax

```
db2gse.ST_Y(p db2gse.ST_Point)
```

Rückgabotyp

Double

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle Y_TEST mit zwei Spalten: der Spalte GID, die die Zeile eindeutig kennzeichnet, und der Spalte PT1.

```
CREATE TABLE Y_TEST (gid integer, pt1 db2gse.ST_Point)
```

Die INSERT-Anweisungen fügen zwei Zeilen ein. Eine der Zeilen ist ein Punkt ohne Z-Koordinate oder Maß. Die andere Spalte hat eine Z-Koordinate und ein Maß.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe listet die Spalte GID und die Double-Y-Koordinate der Punkte auf.

```
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 verwendet einen Punkt und gibt seine Z-Koordinate zurück.

Syntax

```
Z(p db2gse.ST_Point)
```

Rückgabotyp

Double

Beispiele

Die folgende Anweisung CREATE TABLE erstellt die Tabelle Z_TEST mit zwei Spalten: der Spalte GID, die die Zeile eindeutig kennzeichnet, und der Spalte PT1.

```
CREATE TABLE Z_TEST (gid integer, pt1 db2gse.ST_Point)
```

Die folgenden INSERT-Anweisungen fügen zwei Zeilen ein. Eine der Zeilen ist ein Punkt ohne Z-Koordinate oder Maß. Die andere Spalte hat eine Z-Koordinate und ein Maß.

```
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)))
```

Die folgende Anweisung SELECT und die entsprechende Ergebnisgruppe listet die Spalte GID und die Double-Z-Koordinate der Punkte auf. Die erste Zeile ist NULL, da der Punkt keine Z-Koordinate hat.

```
SELECT gid, db2gse.Z(pt1) "The Z coordinate" FROM Z_TEST
```

```
GID          The Z coordinate
-----
1              -
2    +5.000000000000000E+000
```

2 record(s) selected.

Kapitel 15. Koordinatensysteme

Dieses Kapitel enthält Referenzinformationen zum räumlichen Bezugssystem (SRS) und zu den Koordinatenwerten zum Interpretieren der räumlichen Daten.

- „Übersicht über die Koordinatensysteme“
- „Unterstützte lineare Einheiten“ auf Seite 326
- „Unterstützte Winkleinheiten“ auf Seite 326
- „Unterstützte Spheroiden“ auf Seite 327
- „Unterstützte geodätische Fakten“ auf Seite 329
- „Unterstützte primäre Längengrade“ auf Seite 331
- „Unterstützte Kartenprojektionen“ auf Seite 331
- „Unterstützte konische Projektionen“ auf Seite 332
- „Unterstützte azimutale oder planare Projektionen“ auf Seite 332
- „Unterstützte Kartenprojektionsparameter“ auf Seite 332

Übersicht über die Koordinatensysteme

Die bekannte Textdarstellung des räumlichen Bezugssystems bietet eine Standard-Textdarstellung für Informationen zum räumlichen Bezugssystem. Die Definitionen der bekannten Textdarstellung werden entsprechend dem Koordinatensystemmodell der Petrotechnical Open Software Corporation/ European Professional Surveyors Group (POSC/EPSSG) modelliert.

Ein räumliches Bezugssystem ist ein geografisches Koordinatensystem (Breitengrad-Längengrad), ein projiziertes Koordinatensystem (X,Y) oder ein geozentrisches (X,Y,Z) Koordinatensystem. Das Koordinatensystem besteht aus mehreren Objekten. Jedes Objekt hat ein Schlüsselwort in Großbuchstaben (z. B. DATUM oder UNIT), gefolgt von den durch Kommas getrennten Definitionsparametern des Objekts in Klammern. Manche Objekte bestehen aus anderen Objekten, so dass das Ergebnis eine verschachtelte Struktur darstellt.

Anmerkung: Implementierungen können statt der eckigen Klammern [] auch runde Klammern () verwenden und sollten nach Möglichkeit beide Arten von Klammern lesen können.

Die EBNF-Definition (Extended Backus Naur Form) für die Zeichenfolgendarstellung eines Koordinatensystems mit eckigen Klammern lautet wie folgt (siehe Anmerkung oben zur Verwendung der Klammern):

```

<Koordinatensystem> = <projiziertes KS> |
<geografisches KS> | <geozentrisches KS>
<projiziertes KS> = PROJCS["<name>", <geografisches KS>,
<projektion>, {<parameter>,*
<lineare einheit>}]
<projektion> =
PROJECTION["<name>"]
<parameter> = PARAMETER["<name>", <wert>]
<wert> = <zahl>

```

Ein Koordinatensystem eines Datensatzes wird durch das Schlüsselwort PROJCS definiert, wenn die Daten in projizierten Koordinaten stehen (über GEOGCS bei geografischen Koordinaten, über GEOCCS bei geozentrischen Koordinaten). Das Schlüsselwort PROJCS wird gefolgt von allen "Stücken", die das projizierte Koordinatensystem definieren. Das erste Stück jedes Objekts ist immer der Name. Mehrere Objekte folgen dem Namen des projizierten Koordinatensystems: das geografische Koordinatensystem, die Kartenprojektion, ein oder mehrere Parameter und die lineare Maßeinheit. Alle projizierten Koordinatensysteme basieren auf einem geografischen Koordinatensystem; dieser Abschnitt beschreibt daher zunächst die Stücke, die für ein projiziertes Koordinatensystem spezifisch sind. UTM zone 10N im Faktum NAD83 ist beispielsweise wie folgt definiert:

```

PROJCS["NAD_1983_UTM_Zone_10N",
<geographic cs>,
PROJECTION["Transverse_Mercator"],
PARAMETER["False_Easting",500000.0],
PARAMETER["False_Northing",0.0],
PARAMETER["Central_Meridian",-123.0],
PARAMETER["Scale_Factor",0.9996],
PARAMETER["Latitude_of_Origin",0.0],
UNIT["Meter",1.0]]

```

Der Name und verschiedene Objekte definieren wiederum das geografische Koordinatensystemobjekt: das Faktum, der primäre Längengrad und die Winkeleinheit des Maßes.

```

<geografisches KS> = GEOGCS["<name>", <faktum>, <primärer längengrad>,
<winkeleinheit>] <faktum> = DATUM["<name>", <spheroid>]
<spheroid> = SPHEROID["<name>", <halbhauptachse>, <inversionsabflachung>]
<halbhauptachse> = <zahl>
    (die Halbhauptachse wird in Metern gemessen; sie muss > 0 sein.)
<inversionsabflachung> = <zahl>
<primärer längengrad> = PRIMEM["<name>", <längengrad>]
<längengrad> = <zahl>

```

Die Zeichenfolge im geografischen Koordinatensystem für UTM zone 10 in NAD83:

```
GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",  
SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],  
UNIT["Degree",0.0174532925199433]]
```

Das Objekt UNIT kann eine Winkeleinheit oder eine lineare Maßeinheit sein:

```
<winkeleinheit> = <einheit>  
<lineare einheit> = <einheit>  
<einheit> = UNIT["<name>", <umsetzungsfaktor>]  
<umsetzungsfaktor> = <zahl>
```

Der Umsetzungsfaktor gibt die Anzahl der Meter (für eine lineare Einheit) oder die Zahl als Bogenmaß (für eine Winkeleinheit) pro Einheit ein; er muss größer als Null sein.

Die vollständige Zeichenfolgendarstellung für UTM Zone 10N lautet also wie folgt:

```
PROJCS["NAD_1983_UTM_Zone_10N",  
GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],UNIT["Degree",0.0174532925199433]],  
PROJECTION["Transverse_Mercator"],PARAMETER["False_Easting",500000.0],  
PARAMETER["False_Northing",0.0],PARAMETER["Central_Meridian",-123.0],  
PARAMETER["Scale_Factor",0.9996],PARAMETER["Latitude_of_Origin",0.0],  
UNIT["Meter",1.0]]
```

Ein geozentrisches Koordinatensystem ähnelt einem geografischen Koordinatensystem:

```
<geozentrisches KS> = GEOCCS["<name>", <faktum>, <primärer längengrad>,  
<lineare einheit>]
```

Unterstützte lineare Einheiten

Tabelle 58. Unterstützte lineare Einheiten

Einheit	Umsetzungsfaktor
Meter	1,0
Fuß (International)	0,3048
Fuß (US)	12/39,37
Modifizierter amerikanischer Fuß	12,0004584 / 39,37
Clarke's Fuß	12 / 39,370432
Indischer Fuß	12 / 39,370141
Link	7,92 / 39,370432
Link (Benoit)	7,92 / 39,370113
Link (Sears)	7,92 / 39,370147
Chain (Benoit)	792 / 39,370113
Chain (Sears)	792 / 39,370147
Yard (Indian)	36 / 39,370141
Yard (Sears)	36 / 39,370147
Fathom	1,8288
Nautische Meile	1852,0

Unterstützte Winkeleinheiten

Tabelle 59. Unterstützte Winkeleinheiten

Einheit	Umsetzungsfaktor
Bogenmaß	1,0
Dezimalgrad	$\pi/180$
Dezimale Minute	$(\pi/180)/60$
Dezimale Sekunde	$(\pi/180)/36000$
Gon	$\pi/200$
Grad	$\pi/200$

Unterstützte Spheroide

Tabelle 60. Unterstützte Spheroide

Name	Halbhauptachse	Inversionsabflachung
Airy	6377563,396	299,3249646
Modified Airy	6377340,189	299,3249646
Australian	6378160	298,25
Bessel	6377397,155	299,1528128
Modified Bessel	6377492,018	299,1528128
Bessel (Namibia)	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
Clarke 1880 (SGA)	6378249,2	293,46598
Everest 1830	6377276,345	300,8017
Everest 1975	6377301,243	300,8017
Everest (Sarawak and Sabah)	6377298,556	300,8017
Modified Everest 1948	6377304,063	300,8017
Fischer 1960	6378166	298,3
Fischer 1968	6378150	298,3
Modified Fischer (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
International 1909	6378388	297,0
International 1924	6378388	297,0
New International 1967	6378157.5	298,2496
Krasovsky	6378245	298,3

Tabelle 60. Unterstützte Spherioide (Forts.)

Name	Halbhauptachse	Inversionsabflachung
Mercury 1960	6378166	298,3
Modified Mercury 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
South American 1969	6378160	298,25
Southeast Asia	6378155	298,3
Kugel (Radius = 1.0)	1	0
Kugel (Radius = 6371000 m)	6371000	0
Kugel (Radius =6370997 m)	6370997	0
Struve 1860	6378297	294,73
Walbeck	6376896	302,78
War Office	6378300.583	296
WGS 1960	6378165	298,3
WGS 1966	6378145	298,25
WGS 1972	6378135	298,26
WGS 1984	6378137	298,257223563

Unterstützte geodätische Fakten

Tabelle 61. Unterstützte geodätische Fakten

Unterstützte geodätische Fakten	
Adindan	Lissabon
Afgooye	Loma Quintana
Agadez	Lome
Australian Geodetic Datum 1966	Luzon 1911
Australian Geodetic Datum 1984	Mahe 1971
Ain el Abd 1970	Makassar
Amersfoort	Malongo 1987
Aratu	Manoca
Arc 1950	Massawa
Arc 1960	Merchich
Ancienne Triangulation Francaise	Militar-Geografische Institute
Barbados	Mhast
Batavia	Minna
Beduarum	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
Cape	Nouvelle Triangulation Francaise
Carthage	New Zealand Geodetic Datum 1949
Chua	OS (SN) 1980
Conakry 1905	OSGB 1936
Corrego Alegre	OSGB 1970 (SN)
Elfenbeinküste	Padang 1884
Datum 73	Palästina 1923

Tabelle 61. Unterstützte geodätische Fakten (Forts.)

Unterstützte geodätische Fakten	
Deir ez Zor	Pointe Noire
Deutsche Hauptdreiecksnetz	Provisional South American Datum 1956
Douala	Pulkovo 1942
European Datum 1950	Qatar
European Datum 1987	Qatar 1948
Ägypten 1907	Qornoq
European Reference System 1989	RT38
Fahud	Südamerika Datum 1969
Gandajika 1970	Sapper Hill 1943
Garoua	Schwarzeck
Geocentric Datum of Australia 1994	Segora
Guyane Francaise	Serindung
Herat North	Stockholm 1938
Hito XVIII 1963	Sudan
Hu Tzu	Shan Tananarive 1925
Hungarian Datum 1972	Timbalai 1948
Indian 1954	TM65
Indian 1975	TM75
Indonesian Datum 1974	Tokyo
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
La Canoa	WGS 1984
Lake	Yacare
Leigon	Yoff
Liberia 1964	Zanderij

Unterstützte primäre Längengrade

Tabelle 62. Unterstützte primäre Längengrade

Standort	Koordinaten
Greenwich	0° 0' 0"
Bern	7° 26' 22,5" E
Bogota	74° 4' 51,3" W
Brüssel	4° 22' 4,71" E
Ferro	17° 40' 0" W
Jakarta	106° 48' 27,79" E
Lissabon	9° 7' 54,862" W
Madrid	3° 41' 16,58" W
Paris	2° 20' 14,025" E
Rom	12° 27' 8,4" E
Stockholm	18° 3' 29" E

Unterstützte Kartenprojektionen

Tabelle 63. Unterstützte Kartenprojektionen

Zylindrische Projektionen	Pseudozylindrische Projektionen
Behrmann	Craster parabolic
Cassini	Eckert I
Cylindrical equal area	Eckert II
Equirectangular	Eckert III
Gall's stereographic	Eckert IV
Gauss-Kruger	Eckert V
Mercator	Eckert VI
Miller cylindrical	McBryde-Thomas flat polar quartic
Oblique	Mercator (Hotine) Mollweide
Plate-Carée	Robinson
Times	Sinusoidal (Sansom-Flamsteed)
Transverse Mercator	Winkel I

Unterstützte konische Projektionen

Tabelle 64. Unterstützte konische Projektionen

Unterstützte konische Projektionen	
Albers conic equal-area	Chamberlin trimetric
Bipolar oblique conformal conic	Two-point equidistant
Bonne	Hammer-Aitoff equal-area
Equidistant conic	Van der Grinten I
Lambert conformal conic	Verschiedene
Polyconic	Alaska series E
Einfache konische	Alaska Grid (Modified-Stereographic by Snyder)

Unterstützte azimutale oder planare Projektionen

- Azimuthal equidistant
- General vertical near-side perspective
- Gnomonic
- Lambert Azimuthal equal-area
- Orthographic
- Polar-Stereographic
- Stereographic

Unterstützte Kartenprojektionsparameter

Tabelle 65. Unterstützte Kartenprojektionsparameter

Parameter	Beschreibung
central_meridian	Der als Ursprung der X-Koordinaten ausgewählte Längengrad.
scale_factor	Im allgemeinen zur Reduzierung der Verzerrung bei der Kartenprojektion verwendet.
standard_parallel_1	Ein Breitengrad, der normalerweise keine Verzerrung aufweist. Auch für "maßstabgerechter Breitengrad" verwendet.
standard_parallel_2	Ein Längengrad, der normalerweise keine Verzerrung aufweist.

Tabelle 65. Unterstützte Kartenprojektionsparameter (Forts.)

Parameter	Beschreibung
longitude_of_center	Der Längengrad, der den zentralen Punkt der Kartenprojektion definiert.
latitude_of_center	Der Breitengrad, der den zentralen Punkt der Kartenprojektion definiert.
longitude_of_origin	Der Längengrad, der als Ursprung der X-Koordinaten ausgewählt wurde.
latitude_of_origin	Der Breitengrad, der als Ursprung der Y-Koordinaten ausgewählt wurde.
false_easting	Wird zu den X-Koordinaten addiert. Wird verwendet, um positive Werte zu erhalten.
false_northing	Wird zu den Y-Koordinaten addiert. Wird verwendet, um positive Werte zu erhalten.
azimuth	Der Winkel östlich von Nord, der die Mittellinie einer schiefen Projektion definiert.
longitude_of_point_1	Der Längengrad des ersten für eine Kartenprojektion erforderlichen Punkts.
latitude_of_point_1	Der Breitengrad des ersten für eine Kartenprojektion erforderlichen Punkts.
longitude_of_point_2	Der Längengrad des zweiten für eine Kartenprojektion erforderlichen Punkts.
latitude_of_point_2	Der Breitengrad des zweiten für eine Kartenprojektion erforderlichen Punkts.
longitude_of_point_3	Der Längengrad des dritten für eine Kartenprojektion erforderlichen Punkts.
latitude_of_point_3	Der Breitengrad des dritten für eine Kartenprojektion erforderlichen Punkts.
landsat_number	Die Nummer eines Landsat-Satelliten.
path_number	Die Umlaufbahnnummer für einen bestimmten Satelliten.
perspective_point_height	Die Höhe des perspektivischen Punkts der Kartenprojektion über der Erde.
fipszone	Zonennummer des State Plane-Koordinatensystems.
zone	UTM-Zonennummer.

Kapitel 16. Dateiformate für räumliche Daten

Dieses Kapitel dokumentiert die bekannten Darstellungen des Spatial Extender. Die Darstellungen werden als *bekannt* (well-known) bezeichnet, da sie vom Open GIS Consortium (OGC) definiert werden und nicht für den Spatial Extender spezifisch sind. Drei Arten räumlicher Werte sind wichtig, um das Importieren und Exportieren räumlicher Daten zu verstehen:

- Die bekannten OGC-Textdarstellungen (WKT)
- Die bekannten Binärdarstellungen des OGC (WKB)
- Die ESRI-Formdarstellungen

Die bekannten OGC-Textdarstellungen

Spatial Extender hat verschiedene Funktionen, die Geometrien aus Textbeschreibungen generieren:

ST_GeomFromText

Erstellt eine Geometrie aus einer Textdarstellung eines beliebigen Geometrietyps.

ST_PointFromText

Erstellt einen Punkt aus einer Textdarstellung eines Punkts.

ST_LineFromText

Erstellt eine Linienfolge aus der Textdarstellung einer Linienfolge.

ST_PolyFromText

Erstellt ein Polygon aus der Textdarstellung eines Polygons.

ST_MPointFromText

Erstellt eine Mehrpunktangabe aus der Textdarstellung einer Mehrpunktangabe.

ST_MLineFromText

Erstellt eine Mehrlinienfolge aus der Textdarstellung einer Mehrlinienfolge.

ST_MPolyFromText

Erstellt ein Multipolygon aus der Textdarstellung eines Multipolygons.

Die Textdarstellung ist eine ASCII-Zeichenfolge, die den Austausch der Geometrie in ASCII-Textform ermöglicht. Sie können diese Funktionen in einer Sprache der dritten oder vierten Generation (3GL oder 4GL) verwenden, da sie keine Definition spezieller Programmstrukturen erfordern. Die Funktion `ST_AsText` setzt eine vorhandene Geometrie um in eine Textdarstellung.

Jeder Geometrie hat eine bekannte Textdarstellung, die zur Konstruktion neuer Exemplare des Typs und zum Umsetzen vorhandener Exemplare in Textform für die alphanumerische Anzeige verwendet werden kann.

Die bekannte Textdarstellung einer Geometrie ist wie folgt definiert: die Notation {}* gibt 0 oder mehr Wiederholungen der Zeichen innerhalb der geschweiften Klammern an; die geschweiften Klammern erscheinen nicht in der Liste der Ausgabezeichen.

```

<Geometrie-Befehltext> :=
|
| <Punkt-Befehltext>
| <Linienfolgen-Befehltext>
| <Polygon-Befehltext>
| <Mehrpunktangaben-Befehltext>
| <Mehrlinienfolgen-Befehltext>
| <Multipolygon-Befehltext>

<Punkt-Befehltext> :=
POINT (<Punkttext>)

<Linienfolgen-Befehltext> :=
LINESTRING (<Linienfolgentext>)

<Polygon-Befehltext> :=
POLYGON (<Polygontext>)

<Mehrpunktangaben-Befehltext> :=
MULTIPOINT (<Mehrpunktangabentext>)

<Mehrlinienfolgen-Befehltext> :=
MULTILINESTRING (<Mehrlinienfolgentext>)

<Multipolygon-Befehltext> :=
MULTIPOLYGON (<Multipolygontext>)

<Punkttext> := EMPTY
|
| <Punkt>
| Z (<PunktZ>)
| M (<PunktM>)
| ZM (<PunktZM>)

<Punkt> := <x> <y>
<x> := Literal mit doppelter Genauigkeit
<y> := Literal mit doppelter Genauigkeit
<PunktZ> := <x> <y> <z>
<x> := Literal mit doppelter Genauigkeit
<y> := Literal mit doppelter Genauigkeit
<z> := Literal mit doppelter Genauigkeit
<PunktM> := <x> <y> <m>
<x> := Literal mit doppelter Genauigkeit
<y> := Literal mit doppelter Genauigkeit
<m> := Literal mit doppelter Genauigkeit
<PunktZM> := <x> <y> <z> <m>
<x> := Literal mit doppelter Genauigkeit

```

```

<y> := Literal mit doppelter Genauigkeit
<z> := Literal mit doppelter Genauigkeit
<m> := Literal mit doppelter Genauigkeit

<Linienfolgentext> := EMPTY
| ( <Punkttext > {,
<Punkttext> }* )
| Z ( <PunktZ-Text > {, <PunktZ-Text> }* )
| M ( <PunktM-Text > {, <PunktM-Text> }* )
| ZM ( <PunktZM-Text > {, <PunktZM-Text> }* )

<Polygontext> := EMPTY
| ( <Linienfolgentext > {,<Linienfolgentext > }* )

<Mehrpunktangabentext> := EMPTY
| ( <Punkttext > {, <Punkttext > }* )

<Mehrlinienfolgentext> := EMPTY
| ( <Linienfolgentext > {,< Linienfolgentext>}* )

<Multipolygontext> := EMPTY
| ( < Polygontext > {, < Polygontext > }* )

```

Die Basisfunktionssyntax lautet:

```
funktion (<textbeschreibung>,<SRID db2gse.coordref>)
```

Die SRID, die Kennung des räumlichen Bezugssystems, und der Primärschlüssel zu der Tabelle SPATIAL_REFERENCES kennzeichnen das räumliche Bezugssystem der Geometrie, das in der Tabelle SPATIAL_REFERENCES gespeichert ist. Bevor eine Geometrie in eine räumliche Spalte eingefügt wird, muss ihre SRID mit der SRID der räumlichen Spalte übereinstimmen.

Die Textbeschreibung besteht aus drei in einfachen Anführungszeichen eingeschlossenen Basiskomponenten. Beispiel.

```
<'Geometriotyp'> ['Koordinatentyp' ] ['Koordinatenliste']
```

Die Angaben haben folgende Bedeutung:

Geometriotyp

Eine der folgenden Angaben: Punkt, Linienfolge, Polygon, Mehrpunktangabe, Mehrlinienfolge oder Multipolygon.

Koordinatentyp

Gibt an, ob die Geometrie Z-Koordinaten oder Maße hat oder nicht. Lassen Sie dieses Argument leer, wenn die Geometrie weder Z-Koordinaten noch Maße hat. Andernfalls stellen Sie den Koordinatentyp auf Z ein für Geometrien mit Z-Koordinaten, auf M für Geometrien mit Maßen oder auf ZM für Geometrien mit beiden Merkmalen.

Koordinatenliste

Definiert die Scheitelpunkte der Geometrie. Koordinatenlisten werden durch Kommas getrennt und in Klammern eingeschlossen. Geometrien mit mehreren Komponenten erfordern Gruppen von Klammern, in die die einzelnen Teile der Komponenten eingeschlossen werden. Wenn die Geometrie leer ist, ersetzt das Schlüsselwort EMPTY die Koordinate.

Tabelle 66 zeigt eine vollständige Liste der Beispiele aller möglichen Textdarstellungen.

Tabelle 66. Geometrietypen und ihre Textdarstellung

Geometriotyp	Textbeschreibung	Kommentar
point	point empty	Leerer Punkt
point	point z empty	Leerer Punkt mit Z-Koordinate
point	point m empty	Leerer Punkt mit Maß
point	point zm empty	Leerer Punkt mit Z-Koordinate und Maß
point	point (10.05 10.28)	Punkt
point	point z (10.05 10.28 2.51)	Punkt mit Z-Koordinate
point	point m (10.05 10.28 4.72)	Punkt mit Maß
point	point zm (10.05 10.28 2.51 4.72)	Punkt mit Z-Koordinate und Maß
linestring	linestring empty	Leere Linienfolge
linestring	linestring z empty	Leere Linienfolge mit Z-Koordinaten
linestring	linestring m empty	Leere Linienfolge mit Maß
linestring	linestring zm empty	Leere Linienfolge mit Z-Koordinaten und Maß
linestring	linestring (10.05 10.28 , 20.95 20.89)	Linienfolge
linestring	linestring z (10.05 10.28 3.09, 20.95 31.98 4.72, 21.98 29.80 3.51)	Linienfolge mit Z-Koordinaten
linestring	linestring m (10.05 10.28 5.84, 20.95 31.98 9.01, 21.98 29.80 12.84)	Linienfolge mit Maß
linestring	linestring zm ()	Linienfolge mit Z-Koordinaten und Maß
polygon	polygon empty	Leeres Polygon

Tabelle 66. Geometrietypen und ihre Textdarstellung (Forts.)

Geometrietyp	Textbeschreibung	Kommentar
polygon	polygon z empty	Leeres Polygon mit Z-Koordinaten
polygon	polygon m empty	Leeres Polygon mit Maß
polygon	polygon zm empty	Leeres Polygon mit Z-Koordinaten und Maß
polygon	polygon ((10 10, 10 20, 20 20, 20 15, 10 10))	Polygon
polygon	polygon z (())	Polygon mit Z-Koordinaten
polygon	polygon m (())	Polygon mit Maß
polygon	polygon zm (())	Polygon mit Z-Koordinaten und Maß
multipoint	multipoint empty	Leere Mehrpunktangabe
multipoint	multipoint z empty	Leere Mehrpunktangabe mit Z-Koordinaten
multipoint	multipoint m empty	Leere Mehrpunktangabe mit Maß
multipoint	multipoint zm empty	Leere Mehrpunktangabe mit Z-Koordinaten und Maß
multipoint	multipoint empty	Leere Mehrpunktangabe
multipoint	multipoint (10 10, 20 20)	Mehrpunktangabe mit zwei Punkten
multipoint	multipoint z (10 10 2, 20 20 3)	Mehrpunktangabe mit Z-Koordinaten
multipoint	multipoint m (10 10 4, 20 20 5)	Mehrpunktangabe mit Maß
multipoint	multipoint zm (10 10 2 4, 20 20 3 5)	Mehrpunktangabe mit Z-Koordinaten und Maß
multilinestring	multilinestring empty	Leere Mehrlinienfolge
multilinestring	multilinestring z empty	Leere Mehrlinienfolge mit Z-Koordinaten
multilinestring	multilinestring m empty	Leere Mehrlinienfolge mit Maß
multilinestring	multilinestring zm empty	Leere Mehrlinienfolge mit Z-Koordinaten mit Maß
multilinestring	multilinestring (())	Mehrlinienfolge

Tabelle 66. Geometrietypen und ihre Textdarstellung (Forts.)

Geometrietyp	Textbeschreibung	Kommentar
multilinestring	multilinestring z (())	Mehrlinienfolge mit Z-Koordinaten
multilinestring	multilinestring m (())	Mehrlinienfolge mit Maß
multilinestring	multilinestring zm (())	Mehrlinienfolge mit Z-Koordinaten mit Maß
multipolygon	multipolygon empty	Leeres Multipolygon
multipolygon	multipolygon z empty	Leeres Multipolygon mit Z-Koordinaten
multipolygon	multipolygon m empty	Leeres Multipolygon mit Maß
multipolygon	multipolygon z	Leeres Multipolygon mit Z-Koordinaten und Maß
multipolygon	multipolygon ((()))	Multipolygon
multipolygon	multipolygon z ((()))	Multipolygon mit Z-Koordinaten
multipolygon	multipolygon 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)))	Multipolygon mit Maß
multipolygon	multipolygon zm ((()))	Multipolygon mit Z-Koordinaten und Maß

Die bekannten OGC-Binärdarstellungen (WKB)

Spatial Extender hat verschiedene Funktionen, die Geometrien aus Binärdarstellungen generieren:

ST_GeomFromWKB

Erstellt eine Geometrie aus einer bekannten Binärdarstellung eines beliebigen Geometrietyps.

ST_PointFromWKB

Erstellt einen Punkt aus einer bekannten Binärdarstellung eines Punkts.

ST_LineFromWKB

Erstellt eine Linienfolge aus einer bekannten Binärdarstellung einer Linienfolge.

ST_PolyFromWKB

Erstellt ein Polygon aus einer bekannten Binärdarstellung eines Polygons.

ST_MPointFromWKB

Erstellt eine Mehrpunktangabe aus einer bekannten Binärdarstellung einer Mehrpunktangabe.

ST_MLineFromWKB

Erstellt eine Mehrlinienfolge aus einer bekannten Binärdarstellung einer Mehrlinienfolge.

ST_MPolyFromWKB

Erstellt ein Multipolygon aus einer bekannten Binärdarstellung eines Multipolygons.

Die bekannte Binärdarstellung ist ein fortlaufender Byte-Datenstrom. Sie ermöglicht den Austausch der Geometrie zwischen einem ODBC-Client und einer SQL-Datenbank in binärer Form. Da diese Geometriefunktionen die Definition von C-Programmstrukturen für die Zuordnung der Binärdarstellung erfordert, sind diese Funktionen für Sprachen der dritten Generation (3GL) konzipiert. Für Umgebungen mit Sprachen der vierten Generation (4GL) sind diese Funktionen nicht geeignet. Die Funktion `ST_AsBinary` setzt eine vorhandene Geometrie um in eine Binärdarstellung.

Die bekannte Binärdarstellung für die Geometrie ergibt sich durch die Serialisierung eines Geometrieexemplars als Folge numerischer Typen. Diese Typen werden aus der Gruppe (unsigned integer, double) gezogen, und jeder numerische Typ wird als Folge von Bytes serialisiert. Die Typen werden mit einer der beiden rein definierten binären Standarddarstellungen für numerische Typen (NDR, XDR) serialisiert. Ein den serialisierten Bytes vorangestellter Ein-Byte-Befehl beschreibt die spezifische Binärcodierung (NDR oder XDR) für einen Geometrie-Bytestrom. Der einzige Unterschied zwischen den beiden Codierungen der Geometrie ist die Byteanordnung: Die XDR-Codierung entspricht Big Endian; die NDR-Codierung entspricht Little Endian.

Definition numerischer Typen

Ein *unsigned integer* (ganze Zahl ohne Vorzeichen) ist ein 32-Bit-Datentyp (4 Byte), der eine nicht negative Ganzzahl im Bereich [1, 4294967295] codiert.

Ein *double* (Gleitkommazahl mit doppelter Genauigkeit) ist ein 64-Bit-Datentyp (8 Byte) mit doppelter Genauigkeit, der eine Zahl mit doppelter Genauigkeit mit dem IEEE 754 Double Precision Format codiert.

Diese Definitionen gelten sowohl für XDR als auch für NDR.

XDR-Codierung (Big Endian) numerischer Typen

Die XDR-Darstellung eines unsigned integer ist Big Endian (das erste Byte ist das signifikanteste Byte).

Die XDR-Darstellung eines double ist Big Endian (das erste Bit ist das Vorzeichenbit).

NDR-Codierung (Little Endian) numerischer Typen

Die XDR-Darstellung eines unsigned integer ist Little Endian (das erste Byte ist das am wenigsten signifikante Byte).

Die XDR-Darstellung eines double ist Little Endian (das letzte Bit ist das Vorzeichenbit).

Umsetzung zwischen NDR und XDR

Die Umsetzung zwischen den Datentypen NDR und XDR für unsigned integers und doubles ist eine einfache Operation. Sie umfasst die Umkehrung der Byteanordnung innerhalb jedes unsigned integer oder double im Bytestrom.

Beschreibung der WKBBGeometry-Byteströme

Dieser Abschnitt beschreibt die bekannte Binärdarstellung für die Geometrie. Der Grundbaustein ist der Bytestrom für einen Punkt, der aus zwei doubles besteht. Die Byteströme für andere Geometrien werden mit den Byteströmen für bereits definierte Geometrien erstellt.

```
// Basic Type definitions
// byte : 1 byte
// uint32 : 32 bit unsigned integer (4 bytes)
// double : double precision number (8 Byte)

// Building Blocks : Point, LinearRing

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
    Point    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;
    }
};

```

Die folgende Abbildung zeigt eine NDR-Darstellung.

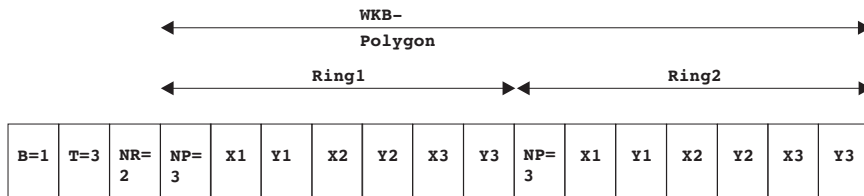


Abbildung 39. Darstellung im NDR-Format. (B=1) des Typs Polygon (T=3) mit 2 linears (NR=2), jeder Ring hat 3 Punkte (NP=3).

Aussagen zur WKB-Darstellung

Die bekannte Binärdarstellung für die Geometrie ist für die Darstellung der Exemplare der Geometrietypen konzipiert; dies wird im Geometrie-Objektmodell und in der OpenGIS Abstract-Spezifikation beschrieben.

Diese Aussagen haben folgende Implikation für Ringe, Polygone und Multipolygone:

Lineare Ringe

Ringe sind einfach und geschlossen; dies bedeutet, dass lineare Ringe sich nicht selbst schneiden können.

Polygone

Keine zwei linearen Ringe innerhalb der Begrenzung eines Polygons können einander schneiden. Die linearen Ringe in der Begrenzung eines Polygons können sich in maximal einem einzigen Punkt berühren.

Multipolygone

Die Innenbereiche von zwei Polygonen, die Elemente eines Multipolygons sind, können sich nicht schneiden. Die Begrenzungen zweier beliebiger Polygone, die Elemente eines Multipolygons sind, können sich nur in einer endlichen Anzahl von Punkten schneiden.

Die ESRI-Formdarstellungen

Spatial Extender hat verschiedene Funktionen, die Geometrien aus ESRI-Formdarstellungen generieren. Zusätzlich zu den von der bekannten binären GIS-Darstellung unterstützten zweidimensionalen Darstellungen unterstützt die ESRI-Formdarstellung auch wahlfreie Z-Koordinaten und Maße. Die folgenden Funktionen generieren Geometrien aus einer ESRI-Form:

GeometryFromShape

Erstellt eine Geometrie aus einer Formdarstellung eines beliebigen Geometrietyps.

PointFromShape

Erstellt einen Punkt aus einer Formdarstellung eines Punkts.

LineFromShape

Erstellt eine Linienfolge aus einer Formdarstellung einer Linienfolge.

PolyFromShape

Erstellt ein Polygon aus einer Formdarstellung eines Polygons.

MPointFromShape

Erstellt eine Mehrpunktangabe aus einer Formdarstellung einer Mehrpunktangabe.

MLineFromShape

Erstellt eine Mehrlinienfolge aus einer Formdarstellung einer Mehrlinienfolge.

MPolyFromShape

Erstellt ein Multipolygon aus einer Formdarstellung eines Multipolygons.

Die allgemeine Syntax dieser Funktionen ist gleich. Das erste Argument ist die Formdarstellung, die als BLOB-Datentyp (großes Binärobjekt) eingegeben wird. Das zweite Argument ist die Kennung des räumlichen Bezugs, die der Geometrie zugeordnet werden soll. Die Funktion `GeometryFromShape` hat die folgende Syntax:

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), cr db2gse.coordref)
```

Da diese Formfunktionen die Definition von C-Programmstrukturen für die Zuordnung der Binärdarstellung erfordern, sind diese Funktionen für Sprachen der dritten Generation (3GL) konzipiert und für Umgebungen der vierten Generation nicht geeignet. Die Funktion `AsShape` setzt den Geometriewert um in eine ESRI-Formdarstellung.

Der Formtyp 0 kennzeichnet eine Nullform ohne geometrische Daten für die Form.

Wert	Formtyp
0	Nullform (Null Shape)
1	Punkt (Point)
3*	Mehrfachlinie (PolyLine)
5	Polygon
8	Mehrpunktangabe (MultiPoint)
11	PunktZ (PointZ)
13	MehrfachlinieZ (PolyLineZ)

Wert	Formtyp
15	PolygonZ (PolygonZ)
18	MehrpunktangabeZ (MultiPointZ)
21	PunktM (PointM)
23	MehrfachlinieM (PolyLineM)
25	PolygonM (PolygonM)
28	MehrpunktangabeM (MultiPointM)

Anmerkung: * Die in dieser Liste nicht angegebenen Formtypen (2, 4, 6 usw.) sind zur späteren Verwendung reserviert.

Formtypen im XY-Raum

Punkt

Ein Punkt besteht aus einem Paar Koordinaten mit doppelter Genauigkeit in der Anordnung X, Y.

Tabelle 67. Punkt-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	1	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	J	J	Double	1	Little

Mehrpunktangabe

Eine Mehrpunktangabe besteht aus einer Gruppe von Punkten. Der Begrenzungsrahmen wird in der Anordnung Xmin, Ymin, Xmax, Ymax gespeichert.

Tabelle 68. Mehrpunkt-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	8	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little

Mehrfachlinie

Eine Mehrfachlinie ist eine geordnete Gruppe von Scheitelpunkten, die aus einem oder mehreren Teilen bestehen. Ein Teil ist eine verbundene Sequenz von zwei oder mehr Punkten. Punkte können miteinander verbunden sein oder auch nicht. Teile können sich schneiden oder auch nicht.

Da diese Spezifikation aufeinanderfolgende Punkte mit identischen Koordinaten zulässt, müssen diese Fälle beim Lesen der Formdatei berücksichtigt werden. Andererseits sind degenerierte Teile mit einer Länge Null nicht zulässig.

Die Felder für eine Mehrfachlinie sind:

Box Der Begrenzungsrahmen der Mehrfachlinie, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts

Die Anzahl der Teile in der Mehrfachlinie.

NumPoints

Die gesamte Anzahl von Punkten für alle Teile.

Parts Eine Feldgruppe von Längen-NumParts. Jede Mehrfachlinie speichert den Index ihres ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jedes Teil in der Mehrfachlinie werden von Ende zu Ende gespeichert. Die Punkte für Teil 2 folgen den Punkten für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jedes Teil. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Teilen.

Tabelle 69. Mehrfachlinien-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	3	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Anmerkung: $X = 44 + 4 * \text{NumParts}$.

Polygon

Ein Polygon besteht aus einem oder mehreren Ringen. Ein Ring ist eine verbundene Folge von vier oder mehr Punkten, die eine geschlossene Schleife bilden, die sich nicht selbst schneidet. Ein Polygon kann mehrere äußere Ringe enthalten. Die Anordnung der Scheitelpunkt oder die Ausrichtung eines Rings gibt an, welche Seite des Rings der Innenbereich des Polygons ist. Der Bereich, der beim Entlanggehen am Ring in Richtung der Scheitelpunkte auf der rechten Seite liegt, ist der Innenbereich des Polygons. Scheitelpunkte von Polygonen, die Löcher in Polygonen definieren, sind gegen den Uhrzeigersinn

angeordnet. Scheitelpunkte für einen einzelnen Polygonring sind daher immer im Uhrzeigersinn angeordnet. Die Ringe eines Polygons werden als Teile bezeichnet.

Da diese Spezifikation aufeinanderfolgende Punkte mit identischen Koordinaten zulässt, müssen diese Fälle beim Lesen der Formdatei berücksichtigt werden. Andererseits sind degenerierte Teile mit einer Länge oder Fläche Null nicht zulässig.

Die Felder für ein Polygon sind:

Box Der Begrenzungsrahmen des Polygons, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts

Die Anzahl der Ringe in dem Polygon.

NumPoints

Die gesamte Anzahl von Punkten für alle Ringe.

Parts Eine Feldgruppe von Längen-NumParts. Speichert für jeden Ring den Index seines ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jeden Ring in dem Polygon werden von Ende zu Ende gespeichert. Die Punkte für Ring 2 folgen den Punkten für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jeden Ring. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Ringen.

Wichtiger Hinweis zu Polygonformen:

- Die Ringe sind geschlossen (der erste und letzte Scheitelpunkt eines Rings MÜSSEN identisch sein).
- Die Reihenfolge der Ringe in der Punktfeldgruppe ist nicht von Bedeutung.
- In einer Formdatei gespeicherte Polygone müssen rein sein. Ein reines Polygon ist wie folgt definiert:
 - Es schneidet sich nicht selbst. Dies bedeutet, dass ein Segment, das zu einem Ring gehört, kein Segment schneidet, das zu einem anderen Ring gehört. Die Ringe eines Polygons können sich an den Schnittpunkten berühren, aber nicht entlang ihrer Segmente. Kolineare Segmente werden als schneidend betrachtet.
 - Die Innenseite des Polygons befindet sich auf der "richtigen" Seite der Linie, durch die es definiert ist. Der Bereich, der beim Entlanggehen am Ring in Richtung der Scheitelpunkte auf der rechten Seite liegt, ist der Innenbereich des Polygons. Scheitelpunkte für einen einzelnen Polygonring sind daher immer im

Uhrzeigersinn angeordnet. Ringe, die Löcher in diesen Polygonen definieren, sind gegen den Uhrzeigersinn angeordnet.

"Nicht reine" Polygone liegen vor, wenn die Ringe, die Löcher in dem Polygon definieren, ebenfalls im Uhrzeigersinn angeordnet sind; dies bewirkt ein Überlappen der Innenbereiche.

Ein Beispielexemplar eines Polygons:

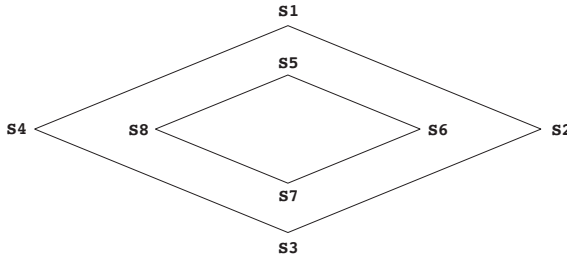


Abbildung 40. Ein Polygon mit einem Loch und acht Scheitelpunkten

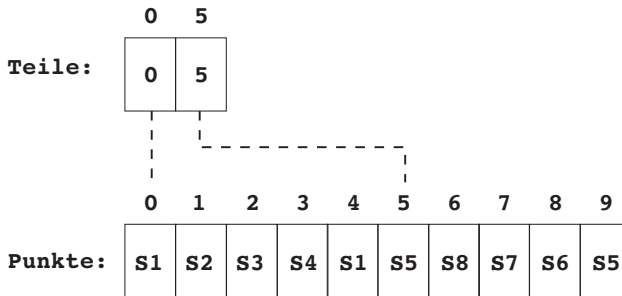


Abbildung 41. Inhalt des Polygon-Bytestroms. NumParts ist 2 und NumPoints ist 10. Beachten Sie, dass die Anordnung der Punkte für die Innenfläche (das Loch) umgekehrt ist.

Tabelle 70. Polygon-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	5	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little

Anmerkung: $X = 44 + 4 * \text{NumParts}$.

Maß-Formtypen in XY-Räumen

PunktM

Ein PunktM besteht aus einem Paar Koordinaten mit doppelter Genauigkeit in der Anordnung X, Y sowie einem Maß M.

Tabelle 71. PunktM-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	21	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	J	J	Double	1	Little
Byte 20	M	M	Double	1	Little

MehrpunktangabeM

Die Felder für eine MehrpunktangabeM sind:

Box Der Begrenzungsrahmen der MehrpunktangabeM, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumPoints

Die Anzahl der Punkte.

Points Eine Feldgruppe von Punkten von Längen-NumPoints.

NumMs

Die Anzahl der folgenden Maße. NumMs kann nur zwei Werte haben: Null, wenn diesem Feld keine Maße folgen, oder gleich NumPoints, wenn Maße vorhanden sind.

M Range

Minimum und Maximum für das Maß für MehrpunktangabeM, gespeichert in der Reihenfolge Mmin, Mmax.

M Array

Eine Feldgruppe von Maßen von Längen-NumPoints.

Tabelle 72. MehrpunktangabeM-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	28	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little
Byte X	NumMs	NumMs	Integer	1	Little
Byte X+4*	Mmin	Mmin	Double	1	Little

Tabelle 72. MehrpunktangabeM-Bytestrom, Inhalt (Forts.)

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte X+12*	Mmax	Mmax	Double	1	Little
Byte X+20*	Marray	Marray	Double	NumPoints	Little

Anmerkungen:

1. $X = 40 + (16 * \text{NumPoints})$
2. * wahlfrei

MehrfachlinieM

Eine Formdatei-MehrfachlinieM besteht aus einem oder mehreren Teilen. Ein Teil ist eine verbundene Sequenz von zwei oder mehr Punkten. Teile können miteinander verbunden sein oder auch nicht. Teile können sich schneiden oder auch nicht.

Die Felder für eine MehrfachlinieM sind:

Box Der Begrenzungsrahmen der MehrfachlinieM, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts

Die Anzahl der Teile in der MehrfachlinieM.

NumPoints

Die gesamte Anzahl von Punkten für alle Teile.

Parts Eine Feldgruppe von Längen-NumParts. Speichert für jeden Teil den Index seines ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jedes Teil in der MehrfachlinieM werden von Ende zu Ende gespeichert. Die Punkte für Teil 2 folgen den Punkten für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jedes Teil. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Teilen.

NumMs

Die Anzahl der folgenden Maße. NumMs kann nur zwei Werte haben: Null, wenn diesem Feld keine Maße folgen, oder gleich NumPoints, wenn Maße vorhanden sind.

M Range

Minimum und Maximum für das Maß für MehrfachlinieM, gespeichert in der Reihenfolge Mmin, Mmax.

M Array

Eine Feldgruppe von Längen-NumPoints. Die Maße für jedes Teil in

der MehrfachlinieM werden von Ende zu Ende gespeichert. Die Maße für Teil 2 folgen den Maßen für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jedes Teil. Es gibt keine Begrenzer in der Maßfeldgruppe zwischen den Teilen.

Tabelle 73. MehrfachlinieM-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	13	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	NumMs	NumMs	Integer	1	Little
Byte Y+4*	Mmin	Mmin	Double	1	Little
Byte Y+12*	Mmax	Mmax	Double	1	Little
Byte Y+20*	Marray	Marray	Double	NumPoints	Little

Anmerkungen:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$.
2. * wahlfrei

PolygonM

Ein PolygonM besteht aus einer Anzahl von Ringen. Ein Ring ist eine geschlossene Schleife, die sich nicht selbst schneidet. Beachten Sie, dass Schnittmengen im XY-Raum berechnet werden, *nicht* im XYM-Raum. Ein PolygonM kann mehrere äußere Ringe enthalten. Die Ringe eines PolygonM werden als Teile bezeichnet.

Die Felder für ein PolygonM sind:

Box Der Begrenzungsrahmen des PolygonM, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts
Die Anzahl der Ringe in dem PolygonM.

NumPoints
Die gesamte Anzahl von Punkten für alle Ringe.

Parts Eine Feldgruppe von Längen-NumParts. Speichert für jeden Ring den Index seines ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jeden Ring in dem PolygonM werden von Ende zu Ende gespeichert. Die Punkte für Ring 2 folgen den Punkten für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jeden Ring. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Ringen.

NumMs

Die Anzahl der folgenden Maße. NumMs kann nur zwei Werte haben: Null, wenn diesem Feld keine Maße folgen, oder gleich NumPoints, wenn Maße vorhanden sind.

M Range

Minimum und Maximum für das Maß für das PolygonM, gespeichert in der Reihenfolge Mmin, Mmax.

M Array

Eine Feldgruppe von Längen-NumPoints. Die Maße für jeden Ring in dem PolygonM werden von Ende zu Ende gespeichert. Die Maße für Ring 2 folgen den Maßen für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startmaßes für jeden Ring. Es gibt keine Begrenzer in der Maßfeldgruppe zwischen den Ringen.

Wichtige Hinweise zu PolygonM-Formen:

- Die Ringe sind geschlossen (der erste und letzte Scheitelpunkt eines Rings müssen identisch sein).
- Die Reihenfolge der Ringe in der Punktfeldgruppe ist nicht von Bedeutung.

Tabelle 74. PolygonM-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	15	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	NumMs	NumMs	Integer	1	Little
Byte Y+4*	Mmin	Mmin	Double	1	Little
Byte Y+12*	Mmax	Mmax	Double	1	Little
Byte Y+20*	Marray	Marray	Double	NumPoints	Little

Anmerkungen:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$.
2. * wahlfrei

Formtypen im XYZ-Raum

PunktZ

Ein PunktZ besteht aus einem Triplet Koordinaten mit doppelter Genauigkeit in der Anordnung X, Y, Z sowie einem Maß M.

Table 75. PunktZ-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	11	Integer	1	Little
Byte 4	X	X	Double	1	Little
Byte 12	J	J	Double	1	Little
Byte 20	Z	Z	Double	1	Little
Byte 28	Measure	M	Double	1	Little

MehrpunktangabeZ

Ein MehrpunktangabeZ stellt eine PunktZ-Gruppe dar wie folgt:

- Der Begrenzungsrahmen wird in der Anordnung Xmin, Ymin, Xmax, Ymax gespeichert.
- Die Z-Begrenzung wird in der Anordnung Zmin, Zmax gespeichert. Die M-Begrenzung wird in der Anordnung Mmin, Mmax gespeichert.

Table 76. MehrpunktangabeZ-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	18	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumPoints	NumPoints	Integer	1	Little
Byte 40	Points	Points	Point	NumPoints	Little
Byte X	Zmin	Zmin	Double	1	Little
Byte X+8	Zmax	Zmax	Double	1	Little
Byte X+16	Zarray	Zarray	Double	NumPoints	Little
Byte Y	NumMs	NumMs	Integer	1	Little
Byte Y+4*	Mmin	Mmin	Double	1	Little
Byte Y+12*	Mmax	Mmax	Double	1	Little
Byte Y+20*	Marray	Marray	Double	NumPoints	Little

Anmerkungen:

1. $X = 40 + (16 * \text{NumPoints}); Y = X + 16 + (8 * \text{NumPoints})$
2. * wahlfrei

MehrfachlinieZ

Eine MehrfachlinieZ besteht aus einem oder mehreren Teilen. Ein Teil ist eine verbundene Sequenz von zwei oder mehr Punkten. Teile können miteinander verbunden sein oder auch nicht. Teile können sich schneiden oder auch nicht.

Die Felder für eine MehrfachlinieZ sind:

Box Der Begrenzungsrahmen der MehrfachlinieZ, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts

Die Anzahl der Teile in der MehrfachlinieZ.

NumPoints

Die gesamte Anzahl von Punkten für alle Teile.

Parts Eine Feldgruppe von Längen-NumParts. Speichert für jeden Teil den Index seines ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jedes Teil in der MehrfachlinieZ werden von Ende zu Ende gespeichert. Die Punkte für Teil 2 folgen den Punkten für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jedes Teil. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Teilen.

Z Range

Die Minimum- und Maximum-Z-Werte für die MehrfachlinieZ, gespeichert in der Anordnung Zmin, Zmax.

Z Array

Eine Feldgruppe von Längen-NumPoints. Die Z-Werte für jedes Teil in der MehrfachlinieZ werden von Ende zu Ende gespeichert. Die Z-Werte für Teil 2 folgen den Z-Werten für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jedes Teil. Es gibt keine Begrenzer in der Z-Feldgruppe zwischen den Teilen.

NumMs

Die Anzahl der folgenden Maße. NumMs kann nur zwei Werte haben: Null, wenn diesem Feld keine Maße folgen, oder gleich NumPoints, wenn Maße vorhanden sind.

M Range

Minimum und Maximum für das Maß für die MehrfachlinieZ, gespeichert in der Reihenfolge Mmin, Mmax.

M Array

Eine Feldgruppe von Längen-NumPoints. Die Maße für jedes Teil in der MehrfachlinieZ werden von Ende zu Ende gespeichert. Die Maße für Teil 2 folgen den Maßen für Teil 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startmaßes für jedes Teil. Es gibt keine Begrenzer in der Maßfeldgruppe zwischen den Teilen.

Tabelle 77. MehrfachlinieZ-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	13	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	Zmin	Zmin	Double	1	Little
Byte Y+8	Zmax	Zmax	Double	1	Little
Byte Y+16	Zarray	Zarray	Double	NumPoints	Little
Byte Z	NumMs	NumMs	Integer	1	Little
Byte Z+4*	Mmin	Mmin	Double	1	Little
Byte Z+12*	Mmax	Mmax	Double	1	Little
Byte Z+20*	Marray	Marray	Double	NumPoints	Little

Anmerkungen:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$, $Z = Y + 16 + (8 * \text{NumPoints})$
2. * wahlfrei

PolygonZ

Ein PolygonZ besteht aus einer Anzahl von Ringen. Ein Ring ist eine geschlossene Schleife, die sich nicht selbst schneidet. Ein PolygonZ kann mehrere äußere Ringe enthalten. Die Ringe eines PolygonZ werden als Teile bezeichnet.

Die Felder für ein PolygonZ sind:

Box Der Begrenzungsrahmen des PolygonZ, gespeichert in der Anordnung Xmin, Ymin, Xmax, Ymax.

NumParts

Die Anzahl der Ringe in dem PolygonZ.

NumPoints

Die gesamte Anzahl von Punkten für alle Ringe.

Parts Eine Feldgruppe von Längen-NumParts. Speichert für jeden Ring den Index seines ersten Punkts in der Punktfeldgruppe. Feldgruppenindizes sind auf Null bezogen.

Points Eine Feldgruppe von Längen-NumPoints. Die Punkte für jeden Ring in dem PolygonZ werden von Ende zu Ende gespeichert. Die Punkte für Ring 2 folgen den Punkten für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startpunkts für jeden Ring. Es gibt keine Begrenzer in der Punktfeldgruppe zwischen den Ringen.

Z Range

Die Minimum- und Maximum-Z-Werte für den Bogen, gespeichert in der Anordnung Zmin, Zmax.

Z Array

Eine Feldgruppe von Längen-NumPoints. Die Z-Werte für jeden Ring in dem PolygonZ werden von Ende zu Ende gespeichert. Die Z-Werte für Ring 2 folgen den Z-Werten für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Start-Z-Werts für jeden Ring. Es gibt keine Begrenzer in der Z-Werte-Feldgruppe zwischen den Ringen.

NumMs

Die Anzahl der folgenden Maße. NumMs kann nur zwei Werte haben: Null, wenn diesem Feld keine Maße folgen, oder gleich NumPoints, wenn Maße vorhanden sind.

M Range

Minimum und Maximum für das Maß für das PolygonZ, gespeichert in der Reihenfolge Mmin, Mmax.

M Array

Eine Feldgruppe von Längen-NumPoints. Die Maße für jeden Ring in dem PolygonZ werden von Ende zu Ende gespeichert. Die Maße für Ring 2 folgen den Maßen für Ring 1, usw. Die Teilefeldgruppen enthalten den Feldgruppenindex des Startmaßes für jeden Ring. Es gibt keine Begrenzer in der Maßfeldgruppe zwischen den Ringen.

Wichtige Hinweise zu PolygonZ-Formen:

- Die Ringe sind geschlossen (der erste und letzte Scheitelpunkt eines Rings MÜSSEN identisch sein).
- Die Reihenfolge der Ringe in der Punktfeldgruppe ist nicht von Bedeutung.

Tabelle 78. PolygonZ-Bytestrom, Inhalt

Position	Feld	Wert	Typ	Anzahl	Anordnung
Byte 0	Shape Type	15	Integer	1	Little
Byte 4	Box	Box	Double	4	Little
Byte 36	NumParts	NumParts	Integer	1	Little
Byte 40	NumPoints	NumPoints	Integer	1	Little
Byte 44	Parts	Parts	Integer	NumParts	Little
Byte X	Points	Points	Point	NumPoints	Little
Byte Y	Zmin	Zmin	Double	1	Little
Byte Y+8	Zmax	Zmax	Double	1	Little
Byte Y+16	Zarray	Zarray	Double	NumPoints	Little
Byte Z	NumMs	NumMs	Integer	1	Little
Byte Z+4*	Mmin	Mmin	Double	1	Little
Byte Z+12*	Mmax	Mmax	Double	1	Little
Byte Z+20*	Marray	Marray	Double	NumPoints	Little

Teil 3. Schlussteil

Bemerkungen

Möglicherweise bietet IBM die in dieser Dokumentation beschriebenen Produkte, Services oder Funktionen in anderen Ländern nicht an. Informationen über die gegenwärtig im jeweiligen Land verfügbaren Produkte und Services sind beim IBM Ansprechpartner erhältlich. Hinweise auf IBM Lizenzprogramme oder andere IBM Produkte bedeuten nicht, dass nur Programme, Produkte oder Dienstleistungen von IBM verwendet werden können. An Stelle der IBM Produkte, Programme oder Dienstleistungen können auch andere ihnen äquivalente Produkte, Programme oder Dienstleistungen verwendet werden, solange diese keine gewerblichen oder anderen Schutzrechte der IBM verletzen. Die Verantwortung für den Betrieb der Produkte, Programme oder Dienstleistungen in Verbindung mit Fremdprodukten und Fremddienstleistungen liegt beim Kunden, soweit nicht ausdrücklich solche Verbindungen erwähnt sind.

Für in diesem Handbuch beschriebene Erzeugnisse und Verfahren kann es IBM Patente oder Patentanmeldungen geben. Mit der Auslieferung dieses Handbuchs ist keine Lizenzierung dieser Patente verbunden. Lizenzanfragen sind schriftlich an IBM Europe, Director of Licensing, 92066 Paris La Defense Cedex, France, zu richten. Anfragen an obige Adresse müssen auf englisch formuliert werden.

Trotz sorgfältiger Bearbeitung können technische Ungenauigkeiten oder Druckfehler in dieser Veröffentlichung nicht ausgeschlossen werden. Die Angaben in diesem Handbuch werden in regelmäßigen Zeitabständen aktualisiert. Die Änderungen werden in Überarbeitungen bekanntgegeben. IBM kann jederzeit Verbesserungen und/oder Änderungen an den in dieser Veröffentlichung beschriebenen Produkten und/oder Programmen vornehmen.

Verweise in diesen Informationen auf Web-Sites anderer Anbieter dienen lediglich als Benutzerinformationen und stellen keinerlei Billigung des Inhalts dieser Web-Sites dar. Das über diese Web-Sites verfügbare Material ist nicht Bestandteil des Materials für dieses IBM Produkt. Die Verwendung dieser Web-Sites geschieht auf eigene Verantwortung.

Werden an IBM Informationen eingesandt, können diese beliebig verwendet werden, ohne dass eine Verpflichtung gegenüber dem Einsender entsteht.

Lizenznehmer des Programms, die Informationen zu diesem Produkt wünschen mit der Zielsetzung: (i) den Austausch von Informationen zwischen unabhängigen, erstellten Programmen und anderen Programmen (einschließlich des vorliegenden Programms) sowie (ii) die gemeinsame Nutzung der ausgetauschten Informationen zu ermöglichen, wenden sich an folgende Adresse:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Die Bereitstellung dieser Informationen kann unter Umständen von bestimmten Bedingungen - in einigen Fällen auch von der Zahlung einer Gebühr - abhängig sein.

Die Lieferung des im Handbuch aufgeführten Lizenzprogramms sowie des zugehörigen Lizenzmaterials erfolgt im Rahmen der Allgemeinen Geschäftsbedingungen der IBM, der Internationalen Nutzungsbedingungen der IBM für Programmpakete oder einer äquivalenten Vereinbarung.

Alle in diesem Dokument enthaltenen Leistungsdaten stammen aus einer gesteuerten Umgebung. Die Ergebnisse, die in anderen Betriebsumgebungen erzielt werden, können daher erheblich von den hier erzielten Ergebnissen abweichen. Einige Daten stammen möglicherweise von Systemen, deren Entwicklung noch nicht abgeschlossen ist. Eine Garantie, dass diese Daten auch in allgemein verfügbaren Systemen erzielt werden, kann nicht gegeben werden. Darüber hinaus wurden einige Daten unter Umständen durch Extrapolation berechnet. Die tatsächlichen Ergebnisse können abweichen. Benutzer dieses Dokuments sollten die entsprechenden Daten in ihrer spezifischen Umgebung prüfen.

Informationen über Produkte anderer Hersteller als IBM wurden von den Herstellern dieser Produkte zur Verfügung gestellt, bzw. aus von ihnen veröffentlichten Ankündigungen oder anderen öffentlich zugänglichen Quellen entnommen. IBM hat diese Produkte nicht getestet und übernimmt im Hinblick auf Produkte anderer Hersteller keine Verantwortung für einwandfreie Funktion, Kompatibilität oder andere Ansprüche. Fragen zu den Leistungsmerkmalen von Produkten anderer Anbieter sind an den jeweiligen Anbieter zu richten.

Aussagen über Pläne und Absichten der IBM unterliegen Änderungen oder können zurückgenommen werden und repräsentieren nur die Ziele der IBM.

Diese Veröffentlichung enthält Beispiele für Daten und Berichte des alltäglichen Geschäftsablaufes. Sie sollen nur die Funktionen des Lizenzprogrammes illustrieren; sie können Namen von Personen, Firmen, Marken oder Produkten enthalten. Alle diese Namen sind frei erfunden, Ähnlichkeiten mit tatsächlichen Namen und Adressen sind rein zufällig.

COPYRIGHT-LIZENZ:

Diese Veröffentlichung enthält Beispielanwendungsprogramme, die in Quellsprache geschrieben sind. Sie dürfen diese Beispielprogramme kostenlos kopieren, ändern und verteilen, wenn dies zu dem Zweck geschieht, Anwendungsprogramme zu entwickeln, verwenden, vermarkten oder zu verteilen, die mit der Anwendungsprogrammierschnittstelle konform sind, für die diese Beispielprogramme geschrieben werden. Die in diesem Handbuch aufgeführten Beispiele sollen lediglich der Veranschaulichung und zu keinem anderen Zweck dienen. Diese Beispiele wurden nicht unter allen denkbaren Bedingungen getestet.

Kopien oder Teile der Beispielprogramme bzw. daraus abgeleiteter Code müssen folgenden Copyrightvermerk beinhalten:

© (Name Ihrer Firma) (Jahr). Teile des vorliegenden Codes wurden aus Beispielprogrammen der IBM Corp. abgeleitet. © Copyright IBM Corp. _Jahr/Jahre angeben_. Alle Rechte vorbehalten.

Marken

Folgende Namen sind in gewissen Ländern Marken der International Business Machines Corporation.

ACF/VTAM	IBM
AISPO	IMS
AIX	IMS/ESA
AIX/6000	LAN DistanceMVS
AIXwindows	MVS/ESA
AnyNet	MVS/XA
APPN	Net.Data
IBM System AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RS/6000
DataPropagator	IBM System /370
DataRefresher	SP
DB2	SQL/DS
DB2 Connect	SQL/400
DB2 Extender	System/370
DB2 OLAP Server	IBM System /390
DB2 Universal Database	SystemView
Distributed Relational	VisualAge
Database Architecture	VM/ESA
DRDA	VSE/ESA
eNetwork	VTAM
Extended Services	WebExplorer
FFST	WIN-OS/2
First Failure Support Technology	

Folgende Namen sind in gewissen Ländern Marken oder eingetragene Marken anderer Unternehmen:

Microsoft, Windows und Windows NT sind Marken oder eingetragene Marken von Microsoft Corporation.

Java und alle auf Java basierenden Marken und Logos sowie Solaris sind in gewissen Ländern Marken von Sun Microsystems, Inc.

Tivoli und NetView sind in gewissen Ländern Marken von Tivoli Systems Inc.

UNIX ist eine eingetragene Marke und wird ausschließlich von der X/Open Company Limited lizenziert.

Andere Namen von Unternehmen, Produkten oder Dienstleistungen können Marken anderer Unternehmen sein.

Index

A

- Abfragen
 - Arten verwendeter räumlicher Funktionen 69
 - Beispielprogramm 79
 - räumliche Indizes nutzen 71
 - räumliche Prädikatfunktionen verwenden 71
 - Schnittstellen zum Übergeben 9, 69
- Abstandsfaktoren
 - angeben 39, 42
- AIX
 - wo Bezugsdaten gespeichert werden 33
 - wo werden Makrodefinitionen für Konstanten gespeichert 83
- Aktivieren der Datenbanken für räumliche Operationen
 - Beschreibung 10, 34
 - DB2-Steuerzentrale Menüauswahlen 35
- Anwendungen
 - gespeicherte Prozeduren 83
 - Richtlinien zum Schreiben 73
- ArcExplorer
 - als Schnittstelle verwenden 9, 69
- ArcExplorer Java Version 3.0
 - herunterladen 30
- AsShape 161, 195, 200
- Attributive Daten 5
- Auslöser
 - automatische Geocodierung aktivieren
 - db2gse.gse_enable_autogc 91
 - automatische Geocodierung inaktivieren
 - db2gse.gse_disable_autogc 86
 - zum Aufrufen des Geocodierers verwenden 45, 54
- Außenbereich 158, 163
- Automatische Geocodierung 54
- Azimutale Projektionen 332

B

- B-Baumstruktur-Indizes 148
- Begrenzung 158, 163
- Beispielprogramm
 - Beschreibung 73

Bezugsdaten 33

C

- Create Spatial Index, Fenster 67
- Create Spatial Layer, Fenster
 - zum Registrieren einer Sichtspalte als Schicht 51
 - zum Registrieren einer Tabellenspalte als Schicht 48
- Create Spatial Reference, Fenster 41, 42

D

- Datenbanken
 - aktivieren für räumliche Operationen
 - Beispielprogramm 74
 - Beschreibung 34
 - DB2-Steuerzentrale Menüauswahlen 35
 - db2gse.gse_enable_db 95
 - inaktivieren der Unterstützung räumlicher Operationen
 - Beispielprogramm 74
 - db2gse.gse_disable_db 89
- Datenelemente 6
- DB2-Steuerzentrale
 - Create Spatial Index, Fenster 67
 - Create Spatial Layer, Fenster
 - zum Registrieren einer Sichtspalte als Schicht 51
 - zum Registrieren einer Tabellenspalte als Schicht 48
 - Create Spatial Reference, Fenster 41, 42
 - Export Spatial Data, Fenster 65
 - Import Spatial Data, Fenster 61, 63, 64
 - Run Geocoder, Fenster 57, 58
 - Spatial Extender aufrufen über 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 95
- db2gse.gse_enable_idx 96

- db2gse.gse_enable_sref 98
- db2gse.gse_export_shape 101
- db2gse.gse_import_sde 103
- db2gse.gse_import_shape 105
- db2gse.gse_register_gc 108
- db2gse.gse_register_layer 110
- db2gse.gse_run_gc 117
- db2gse.gse_unregist_gc 119
- db2gse.gse_unregist_layer 120
- DB2GSE.SPATIAL_GEOCODER 145
- DB2GSE.SPATIAL_REF_SYS 145
- Dimension 164

E

- EBNF (Extended Backus Naur) 324
- EnvelopesIntersect 179, 201
- ESRI-Formdarstellungen
 - Beschreibung 344
 - zugeordnete räumliche Funktionen 194
- Export Spatial Data, Fenster 65

F

- Falsches M
 - angeben 39, 43
- Falsches X
 - angeben 39, 42
- Falsches Y
 - angeben 39, 42
- Falsches Z
 - angeben 39, 43
- Fehlerbehebung, Hinweise
 - Beispielprogramm 28
 - runGseDemo verwenden 28
- Fehlernachrichten 123
- Formen
 - im XY-Raum 346
 - im XYZ-Raum 354

G

- Genauigkeit
 - beibehalten für räumliche Bezugssysteme 37
 - Geocodierung 15, 55
- Geocodierer
 - automatische Geocodierung aktivieren
 - Beispielprogramm 78
 - Beschreibung 45, 54
 - Create Spatial Layer, Fenster 48

- Geocodierer (*Forts.*)
 - db2gse.gse_enable_autogc 91
 - automatische Geocodierung inaktivieren
 - Beispielprogramm 78
 - db2gse.gse_disable_autogc 86
 - Run Geocoder, Fenster 58
- im Stapelbetrieb ausführen
 - Beispielprogramm 76, 78
 - Beschreibung 54
 - db2gse.gse_run_gc 117
 - Run Geocoder, Fenster 57
- Katalogsicht
 - DB2GSE.SPATIAL_GEOCODER 145
- Nicht Standard-Geocodierer
 - Beschreibung 53
 - db2gse.gse_register_gc zum Registrieren verwenden 108
 - db2gse.gse_unregist_gc verwenden zum Aufheben der Registrierung 119
- Standard-Geocodierer 53
- Geocodierung
 - Beschreibung 7, 53
 - Genauigkeit 15
 - schrittweise steigend 54
 - Stapelbetrieb 54
- Geodätische Fakten 329
- GEOGCS, Schlüsselwort 324, 325
- Geografische Merkmale
 - Beschreibung 3
 - durch Daten dargestellt 4
 - zugeordnete Datentypen 46
- Geografisches Informationssystem (GIS)
 - Beschreibung 3
 - erstellen 10
 - verwenden 11
- Geometrien
 - Beschreibung 157
 - Entsprechung zu Typen von räumlichen Daten 160
 - Indizes für räumliche Gitter 150
 - Linienfolgen 159, 166
 - Mehrlinienfolgen 160, 170
 - Mehrpunktangaben 159, 169
 - Merkmale
 - Außenbereich 158, 163
 - Begrenzung 158, 163
 - Dimension 164
 - Innenbereich 158, 163
 - Klasse 160
 - Maße 161
 - Umschlag 149
- Geometrien (*Forts.*)
 - Merkmale (*Forts.*)
 - X-Koordinaten 161
 - Y-Koordinaten 161
 - Z-Koordinaten 161
 - Multipolygone 160, 171
 - Polygone 159, 168
 - Punkte 159, 165
 - GeometryFromShape 194, 203
 - geozentrisches Koordinatensystem 325
 - Gespeicherte Prozeduren
 - db2gse.gse_disable_autogc 86
 - db2gse.gse_disable_db 89
 - db2gse.gse_disable_sref 90
 - db2gse.gse_enable_autogc 91
 - db2gse.gse_enable_db 95
 - db2gse.gse_enable_idx 96
 - db2gse.gse_enable_sref 98
 - db2gse.gse_export_shape 101
 - db2gse.gse_import_sde 103
 - db2gse.gse_import_shape 105
 - db2gse.gse_register_gc 108
 - db2gse.gse_register_layer 110
 - db2gse.gse_run_gc 117
 - db2gse.gse_unregist_gc 119
 - db2gse.gse_unregist_layer 120
 - Gitterindizes 67
- I**
 - Import Spatial Data, Fenster 61, 63, 64
 - Informationsnachrichten 123
 - Innenbereich 158, 163
 - Installieren von Spatial Extender
 - Hardware- und Softwarevoraussetzungen 18
 - überprüfen 26
 - Is3d 161, 205
 - IsMeasured 162, 206
- J**
 - Java 2 Runtime Environment (JRE) v1.2.2 30
- K**
 - Kartenprojektionen 331
 - Kartenprojektionsparameter 332
 - Katalogsichten
 - DB2GSE.COORD_REF_SYS 143
 - DB2GSE.GEOMETRY_COLUMNS 144
 - DB2GSE.SPATIAL_GEOCODER 145
 - DB2GSE.SPATIAL_REF_SYS 145
 - Klasse 160
- Konische Projektionen 332
- Koordinaten
 - Beschreibung 6
 - X-Koordinaten
 - Beschreibung 35
 - Merkmale von Geometrien 161
 - Y-Koordinaten
 - Beschreibung 36
 - Merkmale von Geometrien 161
 - Z-Koordinaten
 - Beschreibung 36
 - Merkmale von Geometrien 161
- Koordinatensysteme 323
 - Beschreibung 6, 35
 - DB2GSE.COORD_REF_SYS
 - Katalogsicht 143
 - räumliche Bezugssysteme ableiten aus 36
- L**
 - Lineare Einheiten 326
 - Lineare Ringe 344
 - LineFromShape 195, 207
 - Linienfolgen 159, 166, 167
 - LocateAlong 190, 209
 - LocateBetween 191, 211
- M**
 - M 166, 213
 - M-Einheiten
 - angeben 40, 43
 - Maß-Formtypen in XY-Räumen 350
 - Maße
 - Beschreibung 36, 161
 - Merkmale von Geometrien 161
 - Maßstabsfaktoren
 - angeben 39, 42
 - MehrfachlinieM-Bytestrom, Inhalt 352
 - Mehrfachlinien-Bytestrom, Inhalt 347
 - MehrfachlinieZ-Bytestrom, Inhalt 356
 - Mehrlinienfolgen 160, 170
 - Mehrpunkt-Bytestrom, Inhalt 346
 - MehrpunktangabeM-Bytestrom, Inhalt 350
 - Mehrpunktangaben 159, 169
 - MehrpunktangabeZ-Bytestrom, Inhalt 354
 - MLine FromShape 214
 - MLineFromShape 195
 - Mosaik 191

MPointFromShape 195, 216
MPolyFromShape 195, 217
Multipolygone 160, 171
Mustermatrizen 174

N

Nachrichten 123
NDR-Codierung 341, 342

P

Planare Projektionen 332
Plattenspeicherplatz, erforderlicher 19
PointFromShape 166, 195, 218
PolyFromShape 195, 219
Polygon-Bytestrom, Inhalt 349
Polygone 159, 168
PolygonM-Bytestrom, Inhalt 353
PolygonZ-Bytestrom, Inhalt 358
POSC/EPSB.Koordinatensystemmodell 323
Primäre Längengrade 331
Primäre Längengrade 331
PROJCS, Schlüsselwort 324
Projektionen
 azimutal 332
 Karte
 Parameter 332
 Typen 331
 konisch 332
 planar 332
Punkt-Bytestrom, Inhalt 346
Punkte 159, 165
PunktM-Bytestrom, Inhalt 350
PunktZ-Bytestrom, Inhalt 354

Q

Quellendaten 5

R

Räumliche Bezugssysteme
 Beschreibung 11
 DB2GSE.SPATIAL_REF_SYS
 Katalogsicht 145
 erstellen
 Beispielprogramm 74
 Beschreibung 35
 Create Spatial Reference,
 Fenster 40
 db2gse.gse_enable_sref 98
 freigeben
 Beispielprogramm 74
 db2gse.gse_disable_sref 90
 Parameter angeben
 Abstandsfaktoren 39, 42
 falsches M 39, 43
 falsches X 39, 42
 falsches Y 39, 42

Räumliche Bezugssysteme (*Forts.*)
 Parameter angeben (*Forts.*)
 falsches Z 39, 43
 M-Einheiten 40, 43
 Maßstabsfaktoren 39, 42
 XY-Einheiten 39, 42
 Z-Einheiten 40, 43

Räumliche Daten

 abgeleitet aus attributiven
 Daten 7
 abgeleitet aus anderen räumli-
 chen Daten
 Beschreibung 7
 räumliche Funktionen, die die
 Daten ableiten 186

Dateiformate

 ESRI-Formdarstellungen 194,
 344
 WKB-Darstellungen (bekannte
 Binärdarstellungen) 193,
 340
 WKT-Darstellungen (bekann-
 ter Text) 192, 335

exportieren

 Beispielprogramm 80
 Beschreibung 59
 db2gse.gse_export_shape 101
 Export Spatial Data, Fens-
 ter 64

importieren

 Beispielprogramm 76
 Beschreibung 8, 59
 db2gse.gse_import_sde 103
 db2gse.gse_import_shape 105
 Import Spatial Data, Fens-
 ter 60, 63

 Wesen 6

Räumliche Funktionen

 .ST_Area 223
 AsShape 195, 200
 EnvelopesIntersect 179, 201
 GeometryFromShape 194
 Is3d 161, 205
 IsMeasured 162, 206
 kategorisiert nach ausgeführten
 Operationen 69
 LineFromShape 195, 207
 LocateAlong 190, 209
 LocateBetween 191, 211
 M 166, 213
 MLine FromShape 214
 MLineFromShape 195
 MPointFromShape 195, 216
 MPolyFromShape 195, 217
 PointFromShape 166, 195, 218

Räumliche Funktionen (*Forts.*)

 PolyFromShape 195, 219
 Prädikate 71
 räumliche Indizes nutzen 71
 ShapeToSQL 194, 221
 ST_Area 169, 172
 ST_AsBinary 194, 225
 ST_AsText 193, 226
 ST_Boundary 163, 227
 ST_Buffer 189, 229
 ST_Centroid 169, 172, 231
 ST_Contains 184, 232
 ST_Convexhull 234
 ST_ConvexHull 191
 ST_CoordDim 162, 236
 ST_Crosses 181, 238
 ST_Difference 187, 240
 ST_Dimension 164, 241
 ST_Disjoint 176, 243
 ST_Distance 186, 245
 ST_Endpoint 167, 246
 ST_Envelope 164, 247
 ST_Equals 175, 249
 ST_ExteriorRing 169, 250
 ST_GeometryN 170, 252
 ST_GeometryType 160, 253
 ST_GeomFromText 192, 255, 335
 ST_GeomFromWKB 193, 257,
 340
 ST_InteriorRingN 169, 259
 ST_Intersection 186, 264
 ST_Intersects 178, 266
 ST_IsClosed 167, 170, 267
 ST_IsEmpty 163, 269
 ST_IsRing 167, 271
 ST_IsSimple 163, 273
 ST_IsValid 161, 274
 ST_Length 167, 170, 276
 ST_LineFromText 192, 278, 335
 ST_LineFromWKB 193, 279, 340
 ST_MLineFromText 193, 281,
 335
 ST_MLineFromWKB 194, 282,
 341
 ST_MPointFromText 192, 284,
 335
 ST_MPointFromWKB 194, 285,
 341
 ST_MPolyFromText 193, 286,
 335
 ST_MPolyFromWKB 194, 287,
 341
 ST_NumGeometries 170, 288
 ST_NumInteriorRing 169, 289
 ST_NumPoints 167, 290

- Räumliche Funktionen (*Forts.*)
 - ST_OrderingEquals 176, 291
 - ST_Overlaps 180, 292
 - ST_Perimeter 169, 294
 - ST_Point 166, 295
 - ST_PointFromText 166, 296, 335
 - ST_PointFromWKB 166, 193, 297, 340
 - ST_PointN 167, 299
 - ST_PointOnSurface 169, 300
 - ST_PolyFromText 192, 301, 335
 - ST_PolyFromWKB 194, 302, 340
 - ST_Polygon 192, 304
 - ST_Relate 186, 305
 - ST_SRID 307
 - ST_StartPoint 167, 308
 - ST_SymmetricDiff 189, 309
 - ST_Touches 179, 311
 - ST_Transform 312
 - ST_Union 188, 314
 - ST_Within 183, 315
 - ST_WKBTtoSQL 193, 316
 - ST_WKTTtoSQL 192, 318
 - ST_X 166, 319
 - ST_Y 166, 320
- Typen
 - Datenaustausch 192
 - Funktionen, die Beziehungen zwischen Geometrien zeigen 173
 - Funktionen, die Geometrien generieren 186
 - Funktionen, die Geometrien vergleichen 173
 - Merkmale von Geometrien zugeordnet 160
 - Prädikatfunktionen 173
 - Zuordnung zu exemplarfähigen Geometrien 165
- Z 166
- Räumliche Indizes 147
 - erstellen
 - Beispielprogramm 77
 - Create Spatial Index, Fenster 67
 - db2gse.gse_enable_idx 96
 - Gittergröße festlegen 68, 154
 - Gitterindizes 67
 - verwenden 153
 - wie sie erstellt werden 149
- Räumliche Indizez
 - nutzen 71
- Räumliche Informationen
 - abrufen und analysieren
 - Arten verwendeter räumlicher Funktionen 69
 - Beispielprogramm 79
 - räumliche Indizes nutzen 71
 - räumliche Prädikatfunktionen verwenden 71
 - zu verwendende Schnittstellen 9, 69
 - Beschreibung 3
- Räumliche Spalten 53
- Run Geocoder, Fenster 57, 58
- S**
- Schichten
 - Beschreibung 12
 - db2gse.gse_unregist_layer verwenden zum Aufheben der Registrierung 120
 - Katalogsicht
 - DB2GSE.GEOMETRY_COLUMNS 144
 - Sichtspalten registrieren
 - Beispielprogramm 79
 - Create Spatial Layer, Fenster 51
 - db2gse.gse_register_layer 110
 - Tabellenspalten registrieren als
 - Beispielprogramm 76
 - Create Spatial Layer, Fenster 48
 - db2gse.gse_register_layer 110
- Schnittstellen zu Spatial Extender 9
- Schrittweise steigende Geocodierung 54
- ShapeToSQL 194, 221
- Spatial Extender
 - Anwendungen
 - gespeicherte Prozeduren 83
 - Richtlinien zum Schreiben 73
 - Aufgaben, Zusammenfassung
 - Beispielprogramm 73
 - Szenario 13
 - Übersicht 10
 - von gespeicherten Prozeduren ausgeführt 84
 - aufrufen über die DB2-Steuerzentrale 32
 - Beispielprogramm
 - Beschreibung 73
 - Fehler, Warnungen und Informationsnachrichten 123
 - gespeicherte Prozeduren 83
- Spatial Extender (*Forts.*)
 - installieren
 - Hardware- und Softwarevoraussetzungen 18
 - Katalogsichten 143
 - Konfiguration 17
 - Räumliche Funktionen 197
 - Ressourcen
 - Bezugsdaten 33
 - für räumliche Operationen 34
 - Zusammenfassung 33
 - Schnittstellen zu 9
 - Zweck 3
 - Spheroide 327
 - SRID (Kennung des räumlichen Bezugssystems) 337
 - ST_Area 169, 172, 223
 - ST_AsBinary 194, 225
 - ST_AsText 193, 226
 - ST_Boundary 163, 227
 - ST_Buffer 189, 229
 - ST_Centroid 169, 172, 231
 - ST_Contains 184, 232
 - ST_Convexhull 234
 - ST_ConvexHull 191
 - ST_CoordDim 162, 236
 - ST_Crosses 181, 238
 - ST_Difference 187, 240
 - ST_Dimension 164, 241
 - ST_Disjoint 176, 243
 - ST_Distance 186, 245
 - ST_Endpoint 167, 246
 - ST_Envelope 164, 247
 - ST_Equals 175, 249
 - ST_ExteriorRing 169, 250
 - ST_GeometryN 170, 252
 - ST_GeometryType 160, 253
 - ST_GeomFromText 192, 255, 335
 - ST_GeomFromWKB 193, 257, 340
 - ST_InteriorRingN 169, 259
 - ST_Intersection 186, 264
 - ST_Intersects 178, 266
 - ST_IsClosed 167, 170, 267
 - ST_IsEmpty 163, 269
 - ST_IsRing 167, 271
 - ST_IsSimple 163, 273
 - ST_IsValid 161, 274
 - ST_Length 167, 170, 276
 - ST_LineFromText 192, 278, 335
 - ST_LineFromWKB 193, 279, 340
 - ST_MLineFromText 193, 281, 335
 - ST_MLineFromWKB 194, 282, 341
 - ST_MPointFromText 192, 284, 335
 - ST_MPointFromWKB 194, 285, 341

ST_MPolyFromText 193, 286, 335
ST_MPolyFromWKB 194, 287, 341
ST_NumGeometries 170, 288
ST_NumInteriorRing 169, 289
ST_NumPoints 167, 290
ST_OrderingEquals 176, 291
ST_Overlaps 180, 292
ST_Perimeter 169, 294
ST_Point 166, 295
ST_PointFromText 166, 296, 335
ST_PointFromWKB 166, 193, 297, 340
ST_PointN 167, 299
ST_PointOnSurface 169, 300
ST_PolyFromText 192, 301, 335
ST_PolyFromWKB 194, 302, 340
ST_Polygon 192, 304
ST_Relate 186, 305
ST_SRID 307
ST_StartPoint 167, 308
ST_SymmetricDiff 189, 309
ST_Touches 179, 311
ST_Transform 312
ST_Union 188, 314
ST_Within 183, 315
ST_WKBToSQL 193, 316
ST_WKTToSQL 192, 318
ST_X 166, 319
ST_Y 166, 320
Standard-Geocodierer 53
Stapelbetriebs-Geocodierung 54
Szenario der Aufgaben 13

T

Typen von räumlichen Daten 45
 Beschreibung 45
 Entsprechung zu Geometrien 160

U

Umschlag 149
UNIT, Schlüsselwort 325

W

Warnungen 123
Windows NT
 wo Bezugsdaten gespeichert werden 33
 wo werden Makrodefinitionen für Konstanten gespeichert 83
Winkeleinheiten 326
WKB-Darstellungen (bekannte Binär-
darstellungen)
 Beschreibung 340
 zugeordnete räumliche Funktionen 193

WKBGeometry 342
WKBGeometry-Byteströme 342
WKT-Darstellungen (bekannter Text)
 Beschreibung 335
 zugeordnete räumliche Funktionen 192

X

X-Koordinaten
 Beschreibung 35
 Merkmale von Geometrien 161

XDR-Codierung 341, 342

XY-Einheiten
 angeben 39, 42

Y

Y-Koordinaten
 Beschreibung 36
 Merkmale von Geometrien 161

Z

Z 166
Z-Einheiten
 angeben 40, 43
Z-Koordinaten
 Beschreibung 36
 Merkmale von Geometrien 161

Kontaktaufnahme mit IBM

Bei technischen Problemen lesen Sie bitte die entsprechenden Korrekturmaßnahmen im Handbuch *Troubleshooting Guide* und führen Sie diese aus, bevor Sie sich mit der IBM Kundenunterstützung in Verbindung setzen. Mit Hilfe dieses Handbuchs können Sie Informationen sammeln, die die DB2-Kundenunterstützung zur Fehlerbehebung verwenden kann.

Wenn Sie weitere Informationen benötigen oder eines der DB2 Universal Database-Produkte bestellen möchten, setzen Sie sich mit einem IBM Ansprechpartner in einer lokalen Geschäftsstelle oder einem IBM Software-Vertriebspartner in Verbindung.

Telefonische Unterstützung erhalten Sie unter der folgenden Nummer:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.

Produktinformationen

Telefonische Unterstützung erhalten Sie über folgende Nummern:

- Unter 0180 3/313 233 erreichen Sie Hallo IBM, wo Sie Antworten zu allgemeinen Fragen erhalten.
- Unter 0180/55 090 können Sie Handbücher telefonisch bestellen.

<http://www.ibm.com/software/data/>

Auf den DB2-World Wide Web-Seiten erhalten Sie aktuelle DB2-Informationen wie Neuigkeiten, Produktbeschreibungen, Schulungspläne und vieles mehr.

<http://www.ibm.com/software/data/db2/library/>

Mit **DB2 Product and Service Technical Library** können Sie auf häufig gestellte Fragen, Berichtigungen, Handbücher und aktuelle technische DB2-Informationen zugreifen.

Anmerkung: Diese Informationen stehen möglicherweise nur auf Englisch zur Verfügung.

<http://www.elink.ibm.com/pbl/pbl/>

Auf der Web-Site für die Bestellung internationaler Veröffentlichungen (International Publications) finden Sie Informationen zum Bestellverfahren.

<http://www.ibm.com/education/certify/>

Das 'Professional Certification Program' auf der IBM Web-Site stellt Zertifizierungstestinformationen für eine Reihe von IBM Produkten, u. a. auch DB2, zur Verfügung.

[ftp.software.ibm.com](ftp://software.ibm.com)

Melden Sie sich als *anonymous* an. Im Verzeichnis `/ps/products/db2` finden Sie Demo-Versionen, Berichtigungen, Informationen und Tools zu DB2 und vielen zugehörigen Produkten.

<comp.databases.ibm-db2>, <bit.listserv.db2-1>

Über diese Internet-Newsgroups können DB2-Benutzer Ihre Erfahrungen mit den DB2-Produkten austauschen.

Für Compuserve: GO IBMDB2

Geben Sie diesen Befehl ein, um auf IBM DB2 Family Forums zuzugreifen. Alle DB2-Produkte werden über diese Foren unterstützt.

In Anhang A des Handbuchs *IBM Software Support Handbook* finden Sie Informationen dazu, wie Sie sich mit IBM in Verbindung setzen können. Rufen Sie die folgende Web-Seite auf, um auf dieses Dokument zuzugreifen:

<http://www.ibm.com/support/>. Wählen Sie anschließend die Verbindung zum IBM Software Support Handbook am unteren Rand der Seite aus.

Anmerkung: In einigen Ländern sollten sich die IBM Vertragshändler an die innerhalb ihrer Händlerstruktur vorgesehene Unterstützung wenden, nicht an die IBM Unterstützungsfunktion.

Antwort

IBM DB2 Spatial Extender
Benutzer- und Referenzhandbuch
Version 7

IBM Form SC12-2894-01

Anregungen zur Verbesserung und Ergänzung dieser Veröffentlichung nehmen wir gerne entgegen. Bitte informieren Sie uns über Fehler, ungenaue Darstellungen oder andere Mängel.

Zur Klärung technischer Fragen sowie zu Liefermöglichkeiten und Preisen wenden Sie sich bitte entweder an Ihre IBM Geschäftsstelle, Ihren IBM Geschäftspartner oder Ihren Händler.

Unsere Telefonauskunft "HALLO IBM" (Telefonnr.: 01803/31 32 33) steht Ihnen ebenfalls zur Klärung allgemeiner Fragen zur Verfügung.

Kommentare:

Danke für Ihre Bemühungen.

Sie können ihre Kommentare betr. dieser Veröffentlichung wie folgt senden:

- Als Brief an die Postanschrift auf der Rückseite dieses Formulars
- Als E-Mail an die folgende Adresse: ibmterm@de.ibm.com

Name

Adresse

Firma oder Organisation

Rufnummer

E-Mail-Adresse

Antwort
SC12-2894-01



IBM Deutschland GmbH
SW TSC Germany

70548 Stuttgart



Teilenummer: CT7M2DE

SC12-2894-01



(1P) P/N: CT7M2DE

