

IBM DB2 Spatial Extender



Guida di riferimento per l'utente

Versione 7

IBM DB2 Spatial Extender



Guida di riferimento per l'utente

Versione 7

Prima di utilizzare questo prodotto e le relative informazioni, consultare la sezione "Informazioni particolari" a pagina 321.

Questo documento contiene informazioni di proprietà dell'IBM. Viene fornito con un accordo di licenza ed è protetto dalle leggi sul copyright. Le informazioni contenute in questa pubblicazione non includono alcuna garanzia sul prodotto e tutte le istruzioni fornite in questo manuale non vanno interpretate in tale senso.

Ordinare le pubblicazioni mediante il rappresentante di zona.

Quando si inviano informazioni all'IBM, si garantisce il diritto non esclusivo all'IBM di utilizzo o distribuzione di queste informazioni nel modo ritenuto più opportuno senza alcun obbligo nei confronti dell'utente.

© Copyright International Business Machines Corporation 1998, 2000. Tutti i diritti riservati.

Indice

Figure vii

Tabelle ix

Informazioni relative alla pubblicazione . . . xi

Destinatari del manuale xi

Convenzioni xi

Modalità di invio commenti xi

Parte 1. Utilizzo di DB2 Spatial Extender 1

Capitolo 1. Informazioni relative a DB2

Spatial Extender 3

Utilizzo di DB2 Spatial Extender 3

Dati che rappresentano funzioni geografiche . 4

Modalità di rappresentazione delle funzioni geografiche nei dati 4

Struttura dei dati spaziali 6

Provenienza dei dati spaziali 6

Modalità di creazione e utilizzo di DB2 Spatial

Extender GIS 8

Interfacce per DB2 Spatial Extender e le funzionalità associate 9

Attività da eseguire per la creazione e l'utilizzo di DB2 Spatial Extender GIS . . 10

Scenario: Una società di assicurazione aggiorna il GIS 12

Capitolo 2. Installazione di DB2 Spatial Extender 17

Configurazione di DB2 Spatial Extender . . 17

Requisiti di sistema 17

Sistemi operativi supportati 18

Software database richiesto 18

Requisiti spazio su disco 18

Installazione di DB2 Spatial Extender . . . 19

Informazioni preliminari 19

Installazione di DB2 Spatial Extender sui sistemi Windows NT 19

Installazione di DB2 Spatial Extender sui sistemi AIX 19

Verifica dell'installazione 20

Considerazioni successive all'installazione . . 21

Scaricamento ArcExplorer 21

Esecuzione del programma di utilità dell'istanza DB2 (db2iupdt) 21

Novità 22

Capitolo 3. Impostazione delle risorse . . . 23

Inventario delle risorse 23

Dati di riferimento 23

Risorse che abilitano un database per le operazioni spaziali 24

Abilitazione di un database per le operazioni spaziali 24

Creazione di un sistema di riferimento spaziale 25

Informazioni relative ai sistemi di riferimento spaziali e di coordinate . . . 25

Creazione di un sistema di riferimento spaziale dal Centro di controllo 29

Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione 33

Informazioni relative ai tipi di dati spaziali . 33

Tipi di dati per funzioni di unità singola . 34

Tipi di dati per funzioni a più unità . . . 35

Un tipo di dati per tutte le funzioni . . . 36

Definizione di una colonna spaziale per una tabella, registrazione della colonna come struttura e abilitazione di un geocoder alla gestione della colonna. 36

Registrazione della colonna della vista come struttura 38

Capitolo 5. Riempimento delle colonne spaziali 41

Utilizzo di funzioni geocoder 41

Informazioni sul processo di geocoding . . 41

Esecuzione del geocoder in modalità batch . 43

Importazione ed esportazione di dati . . . 45

Informazioni su importazione ed esportazione 45

Importazione di dati in una nuova tabella o in una tabella esistente. 46

Importazione dei dati in una tabella esistente 48

Esportazione dei dati in un file shape	49	indici con struttura ad albero B	120
Capitolo 6. Creazione di indici spaziali	51	Modalità di creazione di un indice spaziale	120
Utilizzo del Centro di controllo per creare un		Modalità di generazione di un indice	
indice spaziale	51	spaziale	121
Determinazione della dimensione delle celle		Istruzioni sull'utilizzo di un indice spaziale	125
di griglia	52	Selezione della dimensione delle celle di	
		griglia	126
		Selezione del numero di livelli	126
Capitolo 7. Richiamo e analisi delle		Capitolo 13. Geometrie e funzioni spaziali	
informazioni spaziali	53	associate	129
Metodi di esecuzione dell'analisi spaziale	53	Informazioni sulle geometrie	129
Creazione di un'interrogazione spaziale.	53	Proprietà delle geometrie e delle funzioni	
Funzioni spaziali e SQL	53	associate	131
Predicati spaziali e SQL	54	Classe.	132
		Coordinate X e Y	132
		Coordinate Z	132
		Misure	132
		Interno, limite ed esterno	133
		Semplice o non semplice	133
		Vuoto o non vuoto	133
		Inviluppo	134
		Dimensione.	134
		SRID (Spatial reference system identifier)	135
		Geometrie istanziabili e funzioni associate	135
		Punti	135
		Stringhe lineari	136
		Poligoni	138
		Multipunti	139
		Multistringhe lineari	140
		Multipoligoni	141
		Le funzioni che mostrano le relazioni e i	
		confronti, generano geometrie e convertono i	
		formati dei valori	142
		Le funzioni che mostrano le relazioni e le	
		similitudini tra elementi geografici	143
		Le funzioni che generano nuove	
		geometrie da quelle esistenti	155
		Le funzioni che convertono il formato dei	
		valori di una geometria	160
Capitolo 8. Scrittura di applicazioni per		Capitolo 14. Funzioni spaziali per le	
DB2 Spatial Extender	57	interrogazioni SQL	165
Utilizzo del programma di esempio	57	AsBinaryShape.	166
Passaggi del programma di esempio	57	GeometryFromShape.	167
		EnvelopesIntersect	168
		Is3d	170
		IsMeasured	171
		LineFromShape	172
		LocateAlong	174
Parte 2. Materiale di riferimento 65			
Capitolo 9. Procedure memorizzate	67		
db2gse.gse_disable_autogc	70		
db2gse.gse_disable_db	72		
db2gse.gse_disable_sref	73		
db2gse.gse_enable_autogc	74		
db2gse.gse_enable_db	77		
db2gse.gse_enable_idx	78		
db2gse.gse_enable_sref	81		
db2gse.gse_export_shape.	84		
db2gse.gse_import_sde	86		
db2gse.gse_import_shape	89		
db2gse.gse_register_gc	91		
db2gse.gse_register_layer	93		
db2gse.gse_run_gc	100		
db2gse.gse_unregist_gc	103		
db2gse.gse_unregist_layer	104		
Capitolo 10. Messaggi.	107		
Capitolo 11. Viste del catalogo.	115		
DB2GSE.COORD_REF_SYS	115		
DB2GSE.GEOMETRY_COLUMNS	116		
DB2GSE.SPATIAL_GEOCODER	116		
DB2GSE.SPATIAL_REF_SYS	117		
Capitolo 12. indici spaziali	119		
Frammento di programma di esempio	119		

LocateBetween	176	ST_OrderingEquals	255
M	178	ST_Overlaps	256
MLine FromShape	179	ST_Perimeter	258
MPointFromShape	181	ST_PointFromText	259
MPolyFromShape	182	ST_PointFromWKB	260
PointFromShape	183	ST_Point	261
PolyFromShape	184	ST_PointN	262
ShapeToSQL	186	ST_PointOnSurface	263
ST_Area	188	ST_PolyFromText	264
ST_AsBinary	190	ST_PolyFromWKB	265
ST_AsText	191	ST_Polygon	267
ST_Boundary	192	ST_Relate	268
ST_Buffer	194	ST_SRID	270
ST_Centroid	196	ST_StartPoint	271
ST_Contains	197	ST_SymmetricDiff.	272
ST_ConvexHull	199	ST_Touches	274
ST_CoordDim	201	ST_Transform	275
ST_Crosses	203	ST_Union	276
ST_Difference	205	ST_Within	277
ST_Dimension	206	ST_WKBToSQL	278
ST_Disjoint	208	ST_WKTToSQL	280
ST_Distance	210	ST_X	281
ST_Endpoint	211	ST_Y	282
ST_Envelope	212	Z	283
ST_Equals	214		
ST_ExteriorRing	215	Capitolo 15. Sistemi di coordinate	285
ST_GeometryFromText	217	Panoramica dei sistemi di coordinate	285
ST_GeomFromWKB	219	Unità lineari supportate	287
ST_GeometryN	221	Unità angolari supportate	288
ST_GeometryType	222	Sferoidi supportati	288
ST_InteriorRingN	224	Dati geodetici supportati	290
ST_Intersection	229	Meridiani principali supportati	292
ST_Intersects	231	Proiezioni della carta supportate	292
ST_IsClosed	232	Proiezioni coniche	293
ST_IsEmpty	234	Proiezioni planari e azimutali	293
ST_IsRing	236	Parametri di proiezione della carta	293
ST_IsSimple	237		
ST_IsValid	238	Capitolo 16. Formati file dei dati spaziali	295
ST_Length	240	Le rappresentazioni WKT OGC	295
ST_LineFromText	242	Le rappresentazioni WKB (well-known	
ST_LineFromWKB	243	binary) OGC	300
ST_MLineFromText	245	Definizioni del tipo numerico	301
ST_MLineFromWKB	246	Codifica XDR (Big Endian) dei tipi	
ST_MPointFromText	248	numerici	301
ST_MPointFromWKB	249	Codifica NDR (Little Endian) dei tipi	
ST_MPolyFromText	250	numerici	302
ST_MPolyFromWKB	251	Conversione tra NDR e XDR	302
ST_NumGeometries	252	Descrizione dei flussi di byte	
ST_NumInteriorRing	253	WKBGeometry	302
ST_NumPoints	254		

Asserzioni relative alla rappresentazione	
WKB	304
Le rappresentazioni in formato ESRI	304
Tipi di formato nell'area XY	306
Tipi di formato misura nell'area XY.	310
Tipi di forma nell'area XYZ	314

Parte 3. Appendici 319

Informazioni particolari	321
Marchi	323

Indice analitico 325

Come ottenere ulteriori informazioni dalla	
IBM	331
Informazioni sul prodotto	331

Figure

1. Riga di tabella che rappresenta una funzione geografica; riga di tabella i cui dati di indirizzo rappresentano una funzione geografica	5
2. Tabelle con colonne spaziali aggiunte	5
3. Tabelle che includono dati spaziali derivati da origini dati	7
4. Tabella che include nuovi dati spaziali derivati da dati spaziali esistenti.	8
5. Impostazione client-server	17
6. Gerarchia dei tipi di dati spaziali	34
7. Applicazione di un livello di griglia 10.0e0	122
8. Effetti dell'aggiunta di livelli di griglia 30.0e0 e 60.0e0	124
9. Gerarchia delle geometrie supportate da DB2 Spatial Extender.	130
10. Oggetti della stringa lineare	138
11. Poligoni	138
12. Multistringhe lineari	141
13. Multipoligoni	142
14. ST_Equals	145
15. ST_Disjoint	146
16. ST_Touches	148
17. ST_Overlaps.	149
18. Within.	152
19. ST_Contains.	154
20. Distanza minima tra due città	155
21. ST_Intersection.	156
22. ST_Difference	157
23. ST_Union.	157
24. ST_Buffer.	158
25. LocateAlong.	159
26. LocateBetween	160
27. ST_ConvexHull.	160
28. Utilizzo dell'area per la ricerca della piantina di un edificio	189
29. Un buffer con un raggio di cinque miglia viene applicato ad un punto	195
30. Utilizzo di ST_Contains per verificare che tutti gli edifici siano contenuti all'interno dei relativi lotti	198
31. Utilizzo di ST_Crosses per la ricerca dei corsi d'acqua che attraversano una zona a rischio	204
32. Utilizzo di ST_Disjoint per la ricerca degli edifici che non sono situati all'interno di un'area a rischio.	209
33. Utilizzo di ST_ExteriorRing per stabilire la lunghezza di una linea di costa dell'isola	216
34. Utilizzo di ST_InteriorRingN per stabilire la lunghezza delle linee costiere dei laghi di ciascuna isola	224
35. Utilizzo di ST_Intersection per determinare la dimensione dell'area a rischio di ogni edificio	230
36. Utilizzo di ST_Length per determinare la lunghezza totale dei corsi d'acqua di un paese	241
37. Utilizzare ST_Overlaps per determinare gli edifici che si trovano almeno parzialmente all'interno di una discarica a rischio	257
38. Utilizzare ST_SymmetricDiff per determinare le aree di discariche a rischio che non contengono aree protette (edifici abitati)	273
39. Rappresentazione nel formato NDR	304
40. Un poligono con un foro e otto vertici	309
41. Contenuto del flusso di byte del poligono	309

Tabella

1. Requisiti minimi software	18	19. I parametri di output relativi alla procedura memorizzata db2gse.gse_export_shape..	85
2. Requisiti spazio su disco	18	20. I parametri di input relativi alla procedura memorizzata db2gse.gse_import_sde.	87
3. Operazioni e funzioni spaziali	53	21. I parametri di output relativi alla procedura memorizzata db2gse.gse_import_sde.	87
4. Regole per l'impostazione dell'indice programma di esempio DB2 Spatial Extender	55 58	22. I parametri di input relativi alla procedura memorizzata db2gse.gse_import_shape.	89
6. I parametri di input relativi alla procedura memorizzata db2gse.gse_disable_autogc.	70	23. I parametri di output relativi alla procedura memorizzata db2gse.gse_import_shape.	90
7. I parametri di output relativi alla procedura memorizzata db2gse.gse_disable_autogc.	71	24. I parametri di input relativi alla procedura memorizzata db2gse.gse_register_gc.	91
8. I parametri di output relativi alla procedura memorizzata db2gse.gse_disable_db.	72	25. I parametri di output relativi alla procedura memorizzata db2gse.gse_register_gc.	92
9. I parametri di input relativi alla procedura memorizzata db2gse.gse_disable_sref.	73	26. I parametri di input relativi alla procedura memorizzata db2gse.gse_register_layer..	93
10. I parametri di output relativi alla procedura memorizzata db2gse.gse_disable_sref.	73	27. I parametri di output per la procedura memorizzata db2gse.gse_register_layer.	99
11. I parametri di input relativi alla procedura memorizzata db2gse.gse_enable_autogc.	74	28. I parametri di input relativi alla procedura memorizzata db2gse.gse_run_gc.	100
12. I parametri di output relativi alla procedura memorizzata db2gse.gse_enable_autogc.	76	29. I parametri di output relativi alla procedura memorizzata db2gse.gse_run_gc.	102
13. I parametri di output relativi alla procedura memorizzata db2gse.gse_enable_db..	77	30. I parametri di input relativi alla procedura memorizzata db2gse.gse_unregist_gc.	103
14. I parametri di input relativi alla procedura memorizzata db2gse.gse_enable_idx.	78	31. I parametri di output relativi alla procedura memorizzata db2gse.gse_unregist_gc.	103
15. I parametri di output per la procedura memorizzata db2gse.gse_enable_idx.	79	32. I parametri di input relativi alla procedura memorizzata db2gse.gse_unregist_layer.	105
16. I parametri di input relativi alla procedura memorizzata db2gse.gse_enable_sref.	81	33. I parametri di output relativi alla procedura memorizzata db2gse.gse_unregist_layer.	105
17. I parametri di output relativi alla procedura memorizzata db2gse.gse_enable_sref.	83		
18. I parametri di input relativi alla procedura memorizzata db2gse.gse_export_shape..	84		

34. Colonne della vista della vista di catalogo DB2GSE.COORD_REF_SYS	115	59. Sferoidi supportati	288
35. Colonne nella vista del catalogo DB2GSE.GEOMETRY_COLUMNS	116	60. Dati geodetici supportati	290
36. Colonne della vista del catalogo DB2GSE.SPATIAL_GEOCODER	116	61. Meridiani principali supportati	292
37. Colonne della vista del catalogo DB2GSE.SPATIAL_REF_SYS	117	62. Proiezioni della carta supportate	292
38. Le voci della cella di griglia 10.0e0 per le geometrie di esempio	122	63. Proiezioni coniche	293
39. Intersezioni delle geometrie nell'indice a tre livelli	125	64. Parametri di proiezione della carta	293
40. Matrice per ST_Within	144	65. Tipi di geometria e delle relative rappresentazioni di testo.	298
41. Matrice per l'uguaglianza	145	66. Contenuto del flusso di byte del punto	306
42. Matrice per ST_Disjoint	147	67. Contenuto del flusso di byte del multipunto	306
43. Matrice per ST_Intersects (1)	147	68. Contenuto del flusso di byte del Multiriga	307
44. Matrice per ST_Intersects (2)	147	69. Contenuto del flusso di byte del poligono	309
45. Matrice per ST_Intersects (3)	147	70. Contenuto del flusso di byte di PuntoM	310
46. Matrice per ST_Intersects (4)	148	71. Contenuto del flusso di byte di MultipuntoM	310
47. Matrice per ST_Touches (1)	149	72. Contenuto del flusso di byte di MultirigaM	312
48. Matrice per ST_Touches (2)	149	73. Contenuto del flusso di byte di PoligonoM	313
49. Matrice per ST_Touches (3)	149	74. Contenuto del flusso di byte di PuntoZ	314
50. Matrice per ST_Overlaps (1)	150	75. Contenuto del flusso di byte di MultipuntoZ	314
51. Matrice per ST_Overlaps (2)	150	76. Contenuto del flusso di byte MultirigaZ	316
52. Matrice per ST_Crosses (1)	151	77. Contenuto del flusso di byte di PoligonoZ	318
53. Matrice per ST_Crosses (2)	151		
54. Matrice per ST_Within	153		
55. Matrice per ST_Contains	154		
56. Matrice modello equals	268		
57. Unità lineari supportate	287		
58. Unità angolari supportate	288		

Informazioni relative alla pubblicazione

Questo manuale è suddiviso in due parti. La prima parte contiene informazioni concettuali su DB2 Spatial Extender e fornisce descrizioni relative all'installazione, configurazione, gestione e programmazione di DB2 Spatial Extender sui sistemi Windows NT e AIX. La seconda parte comprende informazioni di riferimento relative alle procedure memorizzate, alle geometrie, alle funzioni, ai messaggi e alle viste di catalogo utilizzate con DB2 Spatial Extender.

Destinatari del manuale

Questo descrive ai responsabili come impostare un ambiente spaziale e ai programmatori di applicazioni come sviluppare le applicazioni con i dati spaziali.

Convenzioni

Questo manuale utilizza le seguenti convenzioni stilistiche:

Grassetto

Indica i comandi e i controlli della GUI (graphical user interface), come i nomi di campi, di cartelle e le scelte di menu.

Carattere monospace

Indica esempi di codifica o del testo da immettere.

Corsivo

Indica le variabili da sostituire con un valore. Il corsivo indica inoltre i titoli dei manuali ed enfatizza le parole.

MAIUSCOLO

Indica le parole chiave SQL e i nomi di oggetti (ad esempio, tabelle, viste e server).

Modalità di invio commenti

Il feedback dell'utente consente la fornitura di informazioni sulla qualità. Inviare eventuali commenti relativi a tale pubblicazione o ad altri documenti DB2. E' possibile inviare i commenti utilizzando una delle seguenti modalità:

- Inviare i commenti dal Web. E' possibile accedere al modulo in linea per i commenti dei lettori di IBM Data Management fornito all'indirizzo:
<http://www.ibm.com/software/data/rcf>

- Inviare i commenti via posta elettronica all'indirizzo `comments@vnet.ibm.com`. E' necessario includere il nome del prodotto, il numero di versione del prodotto e il nome e il numero parte del manuale (se applicabile). Se si inviano commenti su un testo specifico, è necessario includere riferimenti alla posizione del testo nel manuale (ad esempio, il titolo della sezione o del capitolo, un numero di tabella, un numero di pagina o il titolo di un argomento dell'aiuto).

Parte 1. Utilizzo di DB2 Spatial Extender

Capitolo 1. Informazioni relative a DB2 Spatial Extender

Questo capitolo fornisce informazioni preliminari su DB2 Spatial Extender descrivendone l'utilizzo, i dati elaborati e illustrandone il relativo impiego. Alla fine del capitolo è riportata una guida rapida delle successive sezioni.

Utilizzo di DB2 Spatial Extender

E' possibile utilizzare DB2 Spatial Extender per creare un GIS (*geographic information system*): un complesso di oggetti, dati e applicazioni che consentono di generare e analizzare informazioni spaziali sulle funzioni geografiche. *Funzioni geografiche* include gli oggetti comprendenti la superficie terrestre e tutto ciò che contiene. Questi elementi costituiscono l'ambiente naturale (ad esempio i fiumi, le foreste, le colline, i deserti) e l'ambiente culturale (città, abitazioni, edifici adibiti a uffici, segnalazioni, ecc.).

Le *Informazioni spaziali* includono i seguenti fattori:

- L'ubicazione di funzioni geografiche in base ai rispettivi contesti (ad esempio zone della città con ospedali e cliniche oppure gli edifici abitativi prossimi alle zone sismiche)
- Le vie delle funzioni geografiche sono collegate (ad esempio l'informazione che un determinato sistema di fiumi è racchiuso in una specifica regione oppure che alcuni ponti di quella regione attraversano il sistema dei fiumi)
- Le misurazioni applicate a una o più funzioni geografiche (ad esempio la distanza tra un edificio adibito a uffici e la relativa linea di lotto oppure la lunghezza del perimetro di una riserva di uccelli)

Le informazioni spaziali, anche se combinate con un output RDBMS (relational database management system), forniscono un supporto per l'esecuzione di progetti e per le decisioni relative alle politiche aziendali. Si supponga, ad esempio che un'unità sociale locale debba verificare i richiedenti e i destinatari di assistenza che correntemente vivono nell'area di applicazione dei servizi. DB2 Spatial Extender può derivare tali informazioni dall'ubicazione dell'area di riferimento e dagli indirizzi dei richiedenti e dei destinatari.

Oppure si supponga che il proprietario di una catena di ristoranti voglia aprire un'attività nelle città vicine. Per stabilire dove è possibile aprire nuovi ristoranti, il proprietario avrà bisogno delle seguenti informazioni: Dove è presente una maggiore concentrazione di clientela che di solito frequenta il ristorante. Dove sono ubicate le autostrade principali. Dove è presente un tasso di criminalità minore. Dove sono ubicati ristoranti che potrebbero

rappresentare eventuali concorrenti. DB2 Spatial Extender può produrre informazioni spaziali visive per rispondere a tali domande e l'RDBMS sottostante può generare etichette e testo che descrivono le immagini.

Questo manuale comprende numerosi altri esempi di utilizzo di DB2 Spatial Extender, in particolare il "Capitolo 7. Richiamo e analisi delle informazioni spaziali" a pagina 53, il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57 e il "Capitolo 14. Funzioni spaziali per le interrogazioni SQL" a pagina 165.

Dati che rappresentano funzioni geografiche

Questa sezione fornisce una panoramica dei dati generati, memorizzati e gestiti per richiamare informazioni spaziali. Argomenti trattati:

- Modalità di rappresentazione delle funzioni geografiche nei dati
- Struttura dei dati spaziali
- Procedure per la produzione di dati spaziali

Modalità di rappresentazione delle funzioni geografiche nei dati

In DB2 Spatial Extender, una funzione geografica può essere rappresentata da una riga di una tabella o di una vista o da una parte di questa riga. Si considerino ad esempio due delle funzioni geografiche riportate nel capitolo "Utilizzo di DB2 Spatial Extender" a pagina 3, edifici adibiti a uffici e abitazioni. Nella Figura 1 a pagina 5, ciascuna riga della tabella BRANCHES rappresenta una filiale della banca. Come variante, ciascuna riga della tabella CUSTOMERS nella Figura 1 a pagina 5 rappresenta globalmente un cliente della banca. Tuttavia parte di ciascuna riga, in particolare le celle contenenti l'indirizzo del cliente, può essere considerata come rappresentativa dell'abitazione del cliente.

BRANCHES

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

CUSTOMERS

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

Figura 1. Riga di tabella che rappresenta una funzione geografica; riga di tabella i cui dati di indirizzo rappresentano una funzione geografica. La riga di dati contenuta nella tabella BRANCHES rappresenta una filiale della banca. Le celle per i dati di indirizzo nella tabella CUSTOMERS rappresenta l'abitazione di un cliente. I nomi e gli indirizzi riportati nelle tabelle sono fittizi.

Le tabelle della Figura 1 contengono dati che identificano e descrivono le filiali della banca e i clienti. Questi dati sono denominati *dati di attributi*

Una serie secondaria di dati di attributo—i valori che indicano gli indirizzi delle filiali e dei clienti—possono essere ritradotti in valori che forniscono informazioni spaziali. Ad esempio nella Figura 1, l'indirizzo di una filiale è 92467 Airzone Blvd., San Jose CA 95141. L'indirizzo di un cliente è 9 Concourt Circle, San Jose CA 95141. DB2 Spatial Extender può tradurre questi indirizzi in valori che indicano l'ubicazione della filiale e dell'abitazione del cliente rispetto alle aree circostanti. La Figura 2 mostra le tabelle BRANCHES e CUSTOMERS con le nuove colonne che conterranno tali valori.

BRANCHES

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

CUSTOMERS

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

Figura 2. Tabelle con colonne spaziali aggiunte. In ciascuna tabella, la colonna LOCATION conterrà le coordinate corrispondenti agli indirizzi.

Gli indirizzi e altri identificativi dello stesso tipo che si utilizzano come punto di partenza per le informazioni spaziali vengono denominati *dati di origine*. Poiché i valori derivati da tali dati forniscano informazioni spaziali, questi

valori vengono denominati *dati spaziali*. Nella seguente sezione si riporta una descrizione dei dati spaziali e un'introduzione ai tipi di dati ad essa associati.

Struttura dei dati spaziali

La maggior parte dei dati spaziali è composta da coordinate. Una *coordinata* corrisponde a un numero che denota una posizione relativa a un punto di riferimento. Ad esempio le latitudini rappresentano coordinate che denotano posizioni relative all'equatore. Le longitudini rappresentano coordinate che denotano posizioni relative al meridiano di Greenwich. Pertanto la posizione dello Yellowstone National Park è definita dalla relativa latitudine (44.45 gradi a nord dell'equatore) e la longitudine (110.40 gradi ad ovest del meridiano di Greenwich).

Latitudine, longitudine, i relativi punti di riferimento e altri parametri associati, vengono denominati complessivamente *sistema di coordinate*. Esistono anche sistemi di coordinate basati su valori diversi dalla latitudine e la longitudine. Questi sistemi di coordinate dispongono di proprie misure per le posizioni, i punti di riferimento e ulteriori parametri distintivi.

La voce di dati spaziali più semplice comprende due coordinate che definiscono la posizione di una singola funzione geografica. Una *voce di dati* corrisponde al valore o ai valori che occupano la cella di una tabella relazionale. Una voce di dati spaziali più estesa comprende numerose coordinate che definiscono un percorso lineare formato, ad esempio, da un fiume o una strada. Un terzo tipo comprende coordinate che definiscono il perimetro di un'area; ad esempio, l'estremità di una terra o di un flusso. Questi e altri tipi di dati spaziali supportati da DB2 Spatial Extender vengono descritti in modo più dettagliato in "Capitolo 13. Geometrie e funzioni spaziali associate" a pagina 129.

Ciascuna voce di dati spaziali corrisponde a un'istanza del tipo di dati spaziali. Il tipo di dati per due coordinate che contrassegnano un'ubicazione è ST_Point; il tipo di dati per coordinate che definiscono percorsi lineari è ST_LineString; il tipo di dati per coordinate che definiscono i perimetri è ST_Polygon. Questi insieme ad altri tipi di dati per i dati spaziali sono tipi strutturati appartenenti a una singola gerarchia. Per una panoramica della gerarchia vedere il "Informazioni relative ai tipi di dati spaziali" a pagina 33.

Provenienza dei dati spaziali

Come è possibile richiamare dati spaziali:

- Derivandoli dai dati di attributo
- Derivandoli da altri dati spaziali
- Importandoli

Utilizzo di dati di attributo come origini dati

DB2 Spatial Extender può derivare dati spaziali dai dati di attributo, come indirizzi (come descritto nel “Modalità di rappresentazione delle funzioni geografiche nei dati” a pagina 4). Questo processo è denominato *geocoding*. Per visualizzare la sequenza, vedere la Figura 2 a pagina 5 come immagine “precedente” e la Figura 3 come immagine “successiva”. La Figura 2 a pagina 5 mostra la tabella BRANCHES e la tabella CUSTOMERS contenenti una colonna vuota per i dati spaziali. DB2 Spatial Extender può utilizzare queste tabelle per eseguire un geocode degli indirizzi per richiamare le coordinate corrispondenti agli indirizzi, quindi inserirle nelle colonne. La Figura 3 illustra questo risultato.

BRANCHES

ID	NAME	ADDRESS	CITY	STATE	ZIP	LOCATION
937	Aizone-Mulien	92467 Aizone Blvd	San Jose	CA	95141	1653 3094

CUSTOMERS

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

Figura 3. Tabelle che includono dati spaziali derivati da origini dati. La colonna LOCATION della tabella CUSTOMERS contiene coordinate che il geocoder ha derivato dall'indirizzo delle colonne ADDRESS, CITY, STATE e ZIP. Allo stesso modo, La colonna LOCATION della tabella BRANCHES contiene coordinate che il geocoder ha derivato dall'indirizzo delle colonne ADDRESS, CITY, STATE e ZIP di questa tabella. L'esempio illustrato è fittizio; le coordinate raffigurate non sono reali.

DB2 Spatial Extender utilizza una funzione denominata *geocoder*, per convertire i dati di attributo in dati spaziali e per inserirli nelle colonne di tabella. Per ulteriori informazioni relative ai geocoder, vedere la sezione “Informazioni sul processo di geocoding” a pagina 41.

Utilizzo di altri dati spaziali come origini dati

I dati spaziali possono essere generati non solo da dati di attributi, ma anche da altri dati spaziali. Si supponga ad esempio che nella tabella BRANCHES sono definite tutte le filiali di una banca e che la banca richieda il numero di clienti ubicati nel raggio di cinque miglia da ciascuna filiale. Prima che questa informazione venga richiamata dal database, la banca dovrà fornire il database con la definizione della zona che si sviluppa nel raggio di cinque miglia dalla filiale. E' possibile creare tale definizione utilizzando la funzione ST_Buffer di DB2 Spatial Extender. ST_Buffer, utilizzando le coordinate di ciascuna filiale come input, può generare le coordinate che delineano i perimetri della zona prescelta. La Figura 4 a pagina 8 mostra la tabella BRANCHES con le informazioni fornite da ST_Buffer.

BRANCHES

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

Figura 4. Tabella che include nuovi dati spaziali derivati da dati spaziali esistenti. Le coordinate della colonna SALES_AREA vengono derivate dalla funzione ST_Buffer dalle coordinate della colonna LOCATION. Come per la colonna LOCATION, le colonne presenti in SALES_AREA non sono reali, ma simulate.

Oltre a ST_Buffer, DB2 Spatial Extender fornisce numerose altre funzioni che derivano nuovi dati spaziali da dati spaziali esistenti. Per ulteriori informazioni su ST_Buffer e queste altre funzioni, vedere la sezione “Le funzioni che generano nuove geometrie da quelle esistenti” a pagina 155.

Importazione di dati spaziali

Per richiamare dati spaziali è possibile utilizzare un terzo metodo: importarli dai file che presentano uno dei formati supportati da DB2 Spatial Extender. Per una descrizione di questi formati, vedere la sezione “Capitolo 16. Formati file dei dati spaziali” a pagina 295. Questi file contengono dati che di solito vengono applicati alle cartine: dati di censimento, segnalazione di zone soggette a inondazioni, di zone sismiche, e così via. Utilizzando tali dati in combinazione con i dati spaziali prodotti dall’utente, sarà possibile aumentare le informazioni geografiche disponibili. Ad esempio per determinare i rischi che potrebbero verificarsi per una comunità di abitanti, un ente per i lavori pubblici potrà utilizzare ST_Buffer per definire la zona circostante la comunità. L’ente di lavori pubblici potrà quindi importare i dati relativi alle zone soggette a inondazioni e le zone sismiche per determinare se la zona è a rischio.

Modalità di creazione e utilizzo di DB2 Spatial Extender GIS

E’ possibile creare un DB2 Spatial Extender GIS impostando DB2 Spatial Extender e sviluppando progetti GIS in ambienti combinati DB2 Spatial Extender e il DB2 RDBMS sottostante. E’ possibile utilizzare GIS implementando questi progetti; generando quindi e analizzando le informazioni spaziali e tradizionali che tali progetti forniscono. L’intero processo richiede l’esecuzione di varie serie di attività. In questa sezione verranno introdotte le interfacce che consentono di eseguire queste attività e viene presentato il relativo scenario per illustrarle.

Interfacce per DB2 Spatial Extender e le funzionalità associate

In questa sezione si esaminano le interfacce per la creazione di un DB2 Spatial Extender GIS (l'impostazione delle relative risorse, il richiamo dei dati spaziali, e così via) e l'utilizzo (generazione e analisi delle informazioni relative alle funzioni geografiche).

Per creare un DB2 Spatial Extender GIS è possibile seguire le procedure qui riportate:

- Utilizzo delle finestre e le scelte di menu DB2 Spatial Extender del Centro di controllo DB2. Per istruzioni consultare:
 - “Capitolo 3. Impostazione delle risorse” a pagina 23
 - “Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione” a pagina 33
 - “Capitolo 5. Riempimento delle colonne spaziali” a pagina 41
 - “Capitolo 6. Creazione di indici spaziali” a pagina 51
- Esecuzione di un programma applicativo che richiama le procedure memorizzate DB2 Spatial Extender. Per informazioni sullo sviluppo di questo tipo di programma, consultare il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.
- Utilizzo del Centro di controllo e di un programma applicativo. Ad esempio è possibile utilizzare il Centro di controllo per richiamare il geocoder predefinito. Se si desidera utilizzare un geocoder aggiuntivo, è necessario registrarlo con DB2 Spatial Extender richiamando la procedura memorizzata `db2gse.gse_register_gc` in un programma applicativo. Per informazioni relative ai geocoder non predefiniti, vedere la sezione “Informazioni sul processo di geocoding” a pagina 41. Per informazioni relative alla procedura memorizzata `db2gse.gse_register_gc`, consultare “`db2gse.gse_register_gc`” a pagina 91.
- Utilizzo del Centro di controllo, di un programma applicativo, o entrambi in combinazione con altre interfacce. Per creare ad esempio una tabella per mantenere i dati che devono essere generati da una funzione spaziale, come un geocoder, è possibile utilizzare le interfacce del processore riga comandi o del Centro di controllo.

Per utilizzare un DB2 Spatial Extender GIS è possibile seguire le procedure qui riportate:

- Renderizzare graficamente le informazioni con un geobrowser; ad esempio ArcExplorer, fornito da ESRI (Environmental Systems Research Institute)
- Inoltro esplicito delle interrogazioni SQL dal Centro di controllo o dal processore riga comandi DB2
- Inoltro di interrogazioni SQL da un programma applicativo

Attività da eseguire per la creazione e l'utilizzo di DB2 Spatial Extender GIS

Questa sezione fornisce una panoramica delle attività che consentono la creazione e l'utilizzo di un DB2 Spatial Extender GIS. Le attività per la creazione del GIS prevedono l'impostazione di DB2 Spatial Extender e lo sviluppo di progetti GIS. Le attività per l'utilizzo del GIS prevedono l'implementazione dei progetti. Questa panoramica inizia con l'impostazione di DB2 Spatial Extender, quindi prosegue con lo sviluppo e l'implementazione di un progetto GIS. Infine indica le modalità con cui le attività descritte nella panoramica possono variare nella pratica attuale.

Impostazione di DB2 Spatial Extender

Per impostare DB2 Spatial Extender, effettuare le seguenti operazioni:

1. Pianificare e preparare (stabilire il progetto GIS da sviluppare, stabilire il database da abilitare per DB2 Spatial Extender, selezionare il personale che dovrà gestire DB2 Spatial Extender e sviluppare i progetti, e così via).
2. Installazione di DB2 Spatial Extender.
3. Collocare appropriatamente le risorse per il supporto dei progetti GIS, ad esempio:

Risorse fornite da DB2 Spatial Extender

Includono un catalogo di sistema, i tipi di dati spaziali, le funzioni spaziali (incluso il geocoder predefinito) e così via. L'attività di impostazione di tali risorse viene indicata come *abilitazione del database per le operazioni spaziali*.

Geocoder sviluppati da utenti, altri produttori o entrambi.

Il geocoder predefinito converte gli indirizzi degli Stati Uniti in dati spaziali. L'organizzazione dell'utente o altre possono fornire geocoder che convertono indirizzi di altri paesi e altri tipi di dati di attributo in dati spaziali.

Per istruzioni sull'installazione di DB2 Spatial Extender, consultare il "Capitolo 2. Installazione di DB2 Spatial Extender" a pagina 17. Per istruzioni relative all'utilizzo del Centro di controllo per la collocazione appropriata delle risorse, consultare il "Capitolo 3. Impostazione delle risorse" a pagina 23. Per indicazioni sull'utilizzo di un programma applicativo a tale scopo, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57. Per uno scenario illustrativo sulla procedura completa di installazione di DB2 Spatial Extender, consultare il "Un sistema per l'integrazione di dati spaziali e tradizionali" a pagina 13.

Sviluppo e implementazione di un progetto GIS

Per sviluppare e implementare un progetto GIS:

1. Pianificare e preparare (impostare gli obiettivi del progetto, stabilire quali sono le tabelle e i dati necessari, stabilire il sistema di coordinate o i sistemi da utilizzare, e così via).
2. Stabilire il sistema di riferimento spaziale o i sistemi da utilizzare. I valori di coordinate includono di solito numeri interi positivi, numeri negativi e numeri decimali. E' necessario tuttavia che DB2 Spatial Extender memorizzi tutti i valori delle coordinate nel formato di numeri interi positivi. Un *sistema di riferimento spaziale* è una serie di parametri che definisce le modalità di conversione dei numeri negativi e decimali in un sistema di coordinate specifico in numeri positivi, in modo da consentirne la memorizzazione da parte di DB2 Spatial Extender. Dopo aver stabilito il sistema di coordinate da utilizzare per una colonna spaziale, è necessario specificare il sistema di riferimento spaziale che verrà utilizzato per l'esecuzione della conversione richiesta per quella colonna. Solo i sistemi di riferimento spaziali conformi ai requisiti potranno essere utilizzati, in caso contrario sarà necessario crearli.
3. Definire una o più colonne che dovranno contenere dati spaziali, registrarle su DB2 Spatial Extender e abilitare un geocoder alla gestione automatica di esse.

La registrazione di una colonna spaziale attiva la creazione di un record nel catalogo DB2 Spatial Extender. Dopo la registrazione verrà denominata *struttura*, poiché le informazioni da essa generate aggiungeranno uno strato, o struttura, al panorama geografico virtuale creato da GIS. Dopo la registrazione sarà possibile eseguire operazioni spaziali; ad esempio sarà possibile riempirla e definire un indice spaziale su di essa.

4. Riempimento di colonne spaziali:
 - Per i progetti che richiedono l'utilizzo di un geocoder, impostare i parametri per il geocoder. Eseguirlo quindi in modo che in una singola operazione, verrà eseguito il geocode di tutti i dati di origine disponibili e il caricamento delle coordinate risultanti in una struttura.
 - Importare i dati relativi ai progetti che richiedono l'importazione di dati spaziali.
5. Facilitare l'accesso alle colonne spaziali. In particolare definire gli indici che abilitano DB2 all'accesso rapido e definire le viste che consentono agli utenti il richiamo efficiente di dati interrelati. Dopo aver definito questo tipo di vista, è necessario registrare le relative colonne spaziali come strutture.
6. Generare e analizzare informazioni spaziali e informazioni aziendali correlate. Ciò include l'interrogazione di colonne spaziali e colonne di attributo correlate. In questo tipo di interrogazioni è possibile includere funzioni DB2 Spatial Extender che restituiscono un'ampia varietà di informazioni; ad esempio la distanza minima tra due funzioni geografiche o coordinate che definiscono un'area circostante una funzione geografica. Per informazioni relative alla funzione ST_Buffer che restituisce queste

coordinate, vedere le sezioni “Utilizzo di altri dati spaziali come origini dati” a pagina 7 e “ST_Buffer” a pagina 194. Per gli esempi relativi alle interrogazioni che utilizzano funzioni spaziali, vedere il “Capitolo 7. Richiamo e analisi delle informazioni spaziali” a pagina 53 e il “Capitolo 14. Funzioni spaziali per le interrogazioni SQL” a pagina 165.

Per istruzioni relative all'utilizzo del Centro di controllo per l'esecuzione di attività relative allo sviluppo di un progetto GIS, vedere i seguenti capitoli:

- “Capitolo 3. Impostazione delle risorse” a pagina 23
- “Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione” a pagina 33
- “Capitolo 5. Riempimento delle colonne spaziali” a pagina 41
- “Capitolo 6. Creazione di indici spaziali” a pagina 51

Per un supporto sull'utilizzo del Centro di controllo per l'implementazione di un progetto GIS, vedere il “Capitolo 7. Richiamo e analisi delle informazioni spaziali” a pagina 53.

Per un supporto sull'utilizzo di un programma applicativo per lo sviluppo e l'implementazione di un progetto GIS, vedere il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

Per uno scenario illustrativo sulla procedura completa di installazione, vedere la sezione “Un progetto per la collocazione degli uffici e l'assegnazione dei premi” a pagina 13.

Variabilità delle serie di attività

La serie di attività che è possibile eseguire per creare e utilizzare un GIS DB2 Spatial Extender può variare nel contesto e sequenza, a seconda dei requisiti e delle interfacce utilizzati. Si consideri ad esempio l'attività di definizione di una o più colonne che contengono dati spaziali, la registrazione come strutture e l'abilitazione di un geocoder per la gestione automatica di esse. Il Centro di controllo consente di eseguire queste attività simultaneamente, da un'unica finestra. Se si richiamano le procedure memorizzate da un programma, è tuttavia possibile eseguire le attività separatamente e tempificarle in base alle proprie necessità.

Scenario: Una società di assicurazione aggiorna il GIS

Questa sezione presenta uno scenario per illustrare le serie di attività descritte nella sezione precedente.

Gli ambienti di sistema della società di assicurazione Safe Harbor Real Estate include un sistema DB2 Universal Database e un sistema di gestione database separato GIS. Le interrogazioni potranno pertanto richiamare combinazioni di dati dai due sistemi. Ad esempio, una tabella DB2 memorizza informazioni relative ai ricavi e una tabella GIS memorizza l'ubicazione delle filiali della

società. Pertanto sarà possibile verificare l'ubicazione delle filiali che apportano alla società specifiche quantità di ricavi. I dati dei due sistemi non potranno essere integrati (ad esempio gli utenti non potranno unire le colonne DB2 alle colonne GIS) e i servizi DB2 come l'ottimizzazione delle interrogazioni non risulteranno disponibili per GIS. Per risolvere questi problemi, Safe Harbor integra DB2 Spatial Extender e stabilisce un nuovo reparto di sviluppo GIS. Nelle sezioni che seguono vengono descritte le modalità di impostazione di DB2 Spatial Extender e di esecuzione del primo progetto da parte del reparto.

Un sistema per l'integrazione di dati spaziali e tradizionali

Per impostare DB2 Spatial Extender, il reparto di sviluppo GIS di Safe Harbor utilizza la seguente procedura:

1. Il reparto prepara l'inclusione di DB2 Spatial Extender nell'ambiente DB2. Ad esempio:
 - a. Il team di gestione del reparto assegna al team di gestione spaziale le operazioni di installazione e implementazione di DB2 Spatial Extender e al team di analisi spaziale la generazione e l'analisi delle informazioni spaziali.
 - b. Poiché le decisioni aziendali di Safe Harbor vengono adattate alle esigenze dei clienti, il team di gestione deciderà di installare DB2 Spatial Extender nel database contenente le informazioni relative ai clienti. La maggior parte delle informazioni vengono memorizzate in una tabella denominata CUSTOMERS.

Il database selezionato, per ragioni pratiche, verrà denominato *GIS database* dai membri del reparto di sviluppo GIS. Ovviamente il database non verrà utilizzato solo per i progetti GIS; le applicazioni non spaziali potranno continuare a utilizzarlo.
2. Il team di gestione spaziali installa DB2 Spatial Extender.
3. IL team di gestione spaziale imposta le risorse richieste dai progetti GIS:
 - Il team utilizza il Centro di controllo per fornire le risorse che abilitano il database GIS alle operazioni spaziali. Queste risorse includono il catalogo DB2 Spatial Extender, i tipi di dati spaziali, le funzioni spaziali e così via.
 - Poiché Safe Harbor sta iniziando a estendere la propria attività in Canada, il team di gestione spaziale richiederà ai produttori canadesi i geocoder per la conversione di indirizzi canadesi in dati spaziali.

Un progetto per la collocazione degli uffici e l'assegnazione dei premi

Per eseguire il primo progetto GIS utilizzando DB2 Spatial Extender, il reparto di sviluppo GIS procederà nel modo seguente:

1. Il reparto prepara lo sviluppo del progetto, ad esempio
 - Il team di gestione imposta gli obiettivi del progetto:
 - Per stabilire la collocazione delle nuove filiali.

- Per assegnare i premi in base alla vicinanza dei clienti alle aree di rischio (aree con un numero elevato di incidenti stradali, aree ad alto tasso di criminalità, zone soggette ad alluvioni, zone sismiche e così via).
 - Il progetto GIS è predisposto per i clienti e gli uffici degli Stati Uniti. Pertanto il team di gestione spaziale effettuerà le seguenti decisioni:
 - Utilizzare sistemi di coordinate che definiscono accuratamente le zone degli Stati Uniti in cui opera la società Safe Harbor.
 - Utilizzare il geocoder predefinito, poiché è progettato per l'esecuzione di geocode degli indirizzi degli Stati Uniti.
 - Il team di gestione spaziale stabilisce i dati necessari per il raggiungimento degli obiettivi del progetto e le tabelle che conterranno i dati relativi.
2. Utilizzando il Centro di controllo, il team di gestione spaziale creerà due sistemi di riferimento spaziale. Un sistema per stabilire il tipo di conversione in voci di dati memorizzabili in DB2 Spatial Extender da applicare alle coordinate che definiscono l'ubicazione degli uffici. L'altro sistema per stabilire il tipo di conversione in voci di dati memorizzabili in DB2 Spatial Extender da applicare alle coordinate che definiscono le zone di residenza dei clienti.
 3. Utilizzando il Centro di controllo, il team di gestione spaziale definisce le colonne che contengono dati spaziali, le registra come strutture e abilita il geocoder per la gestione automatica di esse:
 - Il team aggiunge una colonna LOCATION alla tabella CUSTOMERS. La tabella contiene già gli indirizzi dei clienti. Il geocoder predefinito li convertirà in dati spaziali che verranno quindi caricati nella colonna LOCATION.
 - Il team crea una tabella OFFICES contenente i dati che ora risulteranno memorizzati in un GIS separato. Questi dati includono gli indirizzi delle filiali Safe Harbor, i dati spaziali derivati da questi indirizzi dal geocoder e i dati spaziali che definiscono la zona che circonda ciascuna filiale nel raggio di cinque miglia. I dati generati dal geocoder verranno inseriti nella colonna LOCATION. I dati che definiscono le zone verranno inseriti nella colonna SALES_AREA.
 - Il team registra le due colonne LOCATION e le colonne SALES_AREA come strutture.
 - Il team abilita il geocoder predefinito per la gestione automatica delle due colonne LOCATION.
 4. Il team di gestione spaziale compila la colonna LOCATION della tabella CUSTOMER, tutta la tabella OFFICES e una nuova tabella HAZARD_ZONES:
 - Il team utilizza il Centro di controllo per compilare la colonna LOCATION della tabella CUSTOMER:

- a. Il team imposta il geocoder per l'inserimento dei dati spaziali relativi a un indirizzo nella colonna LOCATION, solo al verificarsi della seguente condizione: l'indirizzo e i dati forniti dall'ufficio di censimento degli Stati Uniti dovranno corrispondere al 100% tra. Un file degli indirizzi fornito dall'ufficio di censimento viene fornito con DB2 Spatial Extender. Prima che il geocoder possa convertire gli indirizzi dei dati di origine in dati spaziali, il geocoder dovrà verificare la corrispondenza dell'indirizzo all'interno del file. Gli utenti specificano la percentuale di corrispondenza necessaria per consentire l'inserimento dei dati spaziali nella tabella. La percentuale viene denominata *precisione*.
 - b. Il team esegue il geocoder in modo batch, per consentire l'esecuzione di geocode di tutti gli indirizzi delle tabella in un'unica operazione. Il geocoder respinge circa un indirizzo su dieci!
 - c. Il team assume che gli indirizzi respinti non presentano un'esatta corrispondenza con i record dell'ufficio di censimento. Per risolvere questo problema, il team riduce la precisione a 85.
 - d. Il team esegue quindi nuovamente il geocoder in modo batch. Il tasso di indirizzi respinti è ora a un livello accettabile.
- Utilizzando un programma di utilità fornito da un GIS separato, il team carica i dati dell'ufficio in un file. Utilizza quindi il Centro di controllo per importare questi dati dal file alla nuova tabella OFFICES.
 - Utilizzando il Centro di controllo, il team crea una tabella HAZARD ZONES, in cui registra le colonne spaziali come strutture e importa i dati. I dati provengono da un file forniti da un produttore di cartine.
5. Utilizzando il Centro di controllo, il team di gestione spaziale facilita l'accesso alle nuove strutture:
 - Il team crea i relativi indici.
 - Il team crea una vista che unisce le colonne delle tabelle CUSTOMERS e HAZARD ZONES. Il team registra quindi le colonne spaziali delle viste come strutture.
 6. Il team di analisi spaziale esegue interrogazioni per richiamare informazioni che consentono di raggiungere gli obiettivi originari: stabilire la collocazione delle nuove filiali e regolare i premi sulla base della vicinanza dei clienti alle zone di rischio.

Capitolo 2. Installazione di DB2 Spatial Extender

Questo capitolo fornisce istruzioni relative all'installazione di DB2 Spatial Extender. Verranno trattati i seguenti argomenti:

- "Configurazione di DB2 Spatial Extender"
- "Requisiti di sistema"
- "Installazione di DB2 Spatial Extender" a pagina 19
- "Verifica dell'installazione" a pagina 20
- "Considerazioni successive all'installazione" a pagina 21
- "Novità" a pagina 22

Configurazione di DB2 Spatial Extender

Un sistema DB2 Spatial Extender comprende DB2 Universal Database, DB2 Spatial Extender e un geobrowser (ad esempio, ArcExplorer). Di solito un database abilitato per le operazioni spaziali è ubicato sul server. Utilizzare le applicazioni client per accedere ai dati spaziali tramite le procedure memorizzate DB2 Spatial Extender e le interrogazioni spaziali. Il geobrowser consente inoltre di visualizzare i dati spaziali.

La Figura 5 illustra l'architettura di DB2 Spatial Extender.

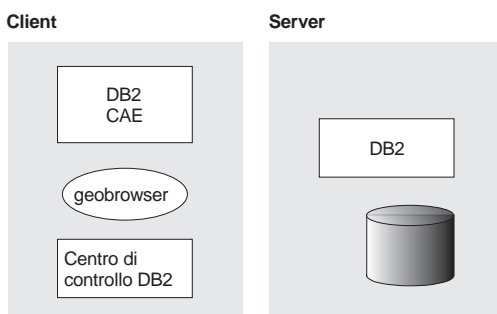


Figura 5. Impostazione client-server

Requisiti di sistema

Questa sezione descrive i requisiti software e hardware relativi a DB2 Spatial Extender.

Sistemi operativi supportati

E' possibile installare DB2 Spatial Extender sui seguenti sistemi operativi:

- AIX 4.2 o successivo
- Windows NT 4.0 o successivo con Service Pack 5

Software database richiesto

Prima di installare DB2 Spatial Extender, è necessario che il software DB2 sia installato e configurato sul sistema. La Tabella 1 elenca i requisiti software database per i componenti client DB2 Spatial Extender e per i componenti server DB2 Spatial Extender.

Tabella 1. Requisiti minimi software

Componente	Software
Client	DB2 Administration Client, Versione 7.1 ¹
Server	Uno dei seguenti: <ul style="list-style-type: none">• DB2 Universal Database Enterprise Edition, Versione 7.1• DB2 Universal Database Enterprise – Extended Edition, Versione 7.1²

Note:

1. Se *non* si intende utilizzare il Centro di controllo DB2, un geobrowser per l'accesso ai dati spaziali o il programma di esempio DB2 Spatial Extender, è possibile utilizzare una versione precedente di DB2 Administration Client.
2. Anche se è possibile utilizzare DB2 Spatial Extender con DB2 Universal Database Enterprise - Extended Edition, l'indice spaziale può non essere suddiviso in partizioni su più nodi come negli ambienti MPP (massive parallel processing).

Requisiti spazio su disco

La Tabella 2 elenca i requisiti di spazio su disco consigliati per DB2 Spatial Extender.

Tabella 2. Requisiti spazio su disco

componente DB2 Spatial Extender	Spazio su disco
Libreria server DB2 Spatial Extender (include libreria server DB2 Spatial Extender, dati di riferimento geocoder e la documentazione)	600 MB
Supporto client DB2 Spatial Extender (include i dati del programma di esempio)	15 MB

Installazione di DB2 Spatial Extender

Questa sezione fornisce informazioni necessarie all'installazione di DB2 Spatial Extender sui sistemi operativi Windows NT e AIX.

Informazioni preliminari

Installare DB2 Administration Client (strumenti di gestione incluso il Centro di controllo e il client di run-time) sulla stazione di lavoro client e installare DB2 Universal Database Enterprise Edition o DB2 Universal Database Enterprise - Extended Edition. Consultare il manuale *Quick Beginnings* per le relative informazioni dettagliate.

Installazione di DB2 Spatial Extender sui sistemi Windows NT

Per installare DB2 Spatial Extender su un sistema Windows NT:

1. Collegarsi al sistema utilizzando un nome utente che dispone delle autorizzazioni di gestione richieste.
2. Chiudere tutti gli altri eventuali programmi in esecuzione.
3. Inserire il CD-ROM nella relativa unità. Verrà aperta la finestra di installazione.
4. Facoltativo: Fare clic su **Note relative al release** per selezionare le Note relative al release DB2 Spatial Extender per tutte le modifiche al processo di installazione, quindi tornare alla finestra di avvio di DB2 Spatial Extender.
5. Fare clic su **Installare**.
6. Rispondere alle richieste del programma di installazione. E' possibile consultare l'aiuto in linea per l'esecuzione dei passi successivi. Per richiamare l'aiuto in linea, fare clic su **Aiuto** o premere il tasto F1.

Al completamento dell'installazione, DB2 Spatial Extender verrà copiato nell'indirizzario %DB2PATH% (ad esempio, c:\sqlib).

Installazione di DB2 Spatial Extender sui sistemi AIX

Per installare DB2 Spatial Extender su un sistema AIX:

1. Collegarsi come root.
2. Inserire il CD-ROM nella relativa unità.
3. Caricare il CD-ROM sul sistema AIX. Per informazioni sul caricamento di un CD-ROM, consultare il manuale *IBM DB2 Universal Database for UNIX Quick Beginnings*.
4. Passare all'indirizzario di caricamento del CD-ROM, immettendo il seguente comando:

```
cd /cdrom
```

in cui *cdrom* corrisponde al punto di caricamento dell'unità CD-ROM su AIX.

5. Immettere il comando **db2setup** per avviare il programma di installazione di DB2. Viene aperta la finestra Installare DB2 Spatial Extender

Nota: Il programma di installazione DB2 verrà avviato lentamente poiché verrà eseguita la scansione delle informazioni di sistema.

6. Dall'elenco dei prodotti presente nella finestra Installare DB2 Spatial Extender, selezionare i prodotti che si desidera installare e fare clic su **OK**. Per ulteriori informazioni o supporto durante l'installazione di DB2 Spatial Extender, fare clic su **Aiuto**.

Al completamento dell'installazione, DB2 Spatial Extender verrà installato nell'indirizzo /usr/lpp/db2_07_01.

Verifica dell'installazione

Dopo aver installato DB2 Spatial Extender, è possibile verificarne l'installazione utilizzando il programma di esempio DB2 Spatial Extender. Prima di eseguire il programma di esempio, è necessario creare un database SAMPLE e creare un file eseguibile del programma di esempio.

Nota: E' necessario utilizzare il compiler specificato nel makefile DB2 Spatial Extender.

Per compilare ed eseguire il programma di installazione per Windows NT:

1. Collegarsi utilizzando l'ID utente con privilegi di responsabile.
2. Da una richiesta riga comandi, immettere **db2sampl** per creare il database DB2 SAMPLE.
3. Da una richiesta riga comandi, immettere il seguente comando:
`cd %DB2PATH%\samples\spatial`

Nota: Per poter eseguire il passo 3 e proseguire nella verifica dell'installazione, è necessario disporre dell'istanza DB2 predefinita (DB2-DB2).

4. Immettere **make rungsedemo**.
5. Immettere **rungsedemo.exe**.
6. Controllare i messaggi di errore e di completamento visualizzati durante l'esecuzione del programma.

Per compilare ed eseguire il programma di installazione per AIX:

1. Collegarsi come root.
2. Creare o aggiornare l'istanza DB2.
3. Da una richiesta riga comandi, immettere **db2sampl** per creare il database DB2 SAMPLE.
4. Da una richiesta riga comandi, immettere il seguente comando:

```
cd $DB2INSTANCE/sql1lib/samples/spatial
```

Nota: Per poter eseguire il passo 4 a pagina 20 e proseguire nella verifica dell'installazione, è necessario disporre dell'istanza DB2 creata o aggiornata.

5. Immettere **make rungsedemo**.
6. Immettere **rungsedemo**.
7. Controllare i messaggi di errore e di completamento visualizzati durante l'esecuzione del programma.

Per informazioni relative ai programmi di esempio, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Considerazioni successive all'installazione

Dopo aver correttamente installato DB2 Spatial Extender, potrebbe essere necessario eseguire le seguenti operazioni:

- Scaricare ArcExplorer
- Eseguire il programma di utilità dell'istanza DB2

Scaricamento ArcExplorer

IBM distribuisce ArcExplorer Java 3.0 come programma di esempio oppure è disponibile nel sito Web ESRI all'indirizzo <http://www.esri.com>.

Per ulteriori informazioni sull'installazione e l'utilizzo di ArcExplorer, consultare il manuale *Using ArcExplorer*, disponibile anche nel sito Web ESRI.

ArcExplorer richiede Java[®] 2 Runtime Environment (Standard Edition o Enterprise Edition), V1.2.2 disponibile gratuitamente nel sito Web della Sun all'indirizzo <http://java.sun.com>.

Importante: DB2 Universal Database V7.1 è fornito con IBM JDK 1.1.8. Quando si installa JRE 1.2.2 per ArcExplorer, è necessario copiarlo in un'indirizzario diverso da DB2. Impostare correttamente la variabile di ambiente CLASSPATH.

Esecuzione del programma di utilità dell'istanza DB2 (db2iupdt)

Il programma di utilità db2iupdt aggiorna un'istanza DB2 specifica su:

- Abilitazione dell'istanza per l'acquisizione di una nuova configurazione del sistema.
- Abilitazione dell'istanza per l'accesso a una funzione associata all'installazione o alla rimozione di alcune opzioni del prodotto.

Su AIX, questo programma di utilità è ubicato nell'indirizzario `/usr/lpp/db2_07_01`. Se si desidera visualizzare l'aiuto, digitare `db2iupdt -h` sulla riga comandi. Sul sistema operativo Windows NT, db2iupdt è ubicato

nell'indirizzario \sqlib\bin. Aprire questo indirizzario per digitare il comando. Per una descrizione completa di questo comando, consultare il manuale *IBM DB2 Universal Database Command Reference*.

Novità

Dopo aver installato DB2 Spatial Extender, è possibile utilizzare il Centro di controllo DB2 per impostare l'ambiente GIS e iniziare a utilizzare le informazioni spaziali.

Per richiamare DB2 Spatial Extender dal Centro di controllo:

1. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Database** nel server su cui si desidera che DB2 Spatial Extender venga eseguito.
2. Fare clic sulla cartella **Database**. I database vengono visualizzati nel pannello indice posto a destra della finestra.
3. Fare clic con il tastino destro del mouse sul database che si desidera utilizzare, quindi fare clic sull'operazione spaziale che si desidera eseguire nel menu a comparsa.

Per ulteriori informazioni sull'utilizzo di DB2 Spatial Extender dal Centro di controllo, consultare i seguenti argomenti:

- "Capitolo 3. Impostazione delle risorse" a pagina 23
- "Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione" a pagina 33
- "Capitolo 5. Riempimento delle colonne spaziali" a pagina 41
- "Capitolo 6. Creazione di indici spaziali" a pagina 51

Capitolo 3. Impostazione delle risorse

Dopo aver installato DB2 Spatial Extender, è possibile fornire al database le risorse necessarie per la creazione di colonne spaziali e la gestione di dati spaziali. In questo capitolo viene fornito un riepilogo delle risorse e una descrizione di due delle attività che è necessario eseguire per renderle disponibili: abilitazione del database per le operazioni spaziali e creazione di sistemi di riferimento spaziali.

Inventario delle risorse

Le risorse che si intende utilizzare per la creazione di colonne spaziali e la gestione di dati spaziali includono:

- *Dati di riferimento*: indirizzi analizzati da DB2 Spatial Extender per rilevare quelli di cui si desidera eseguire un geocode
- Risorse che abilitano un database per le operazioni spaziali: procedure memorizzate, funzioni spaziali e altre
- Geocoder non predefiniti forniti dagli utenti e dagli altri produttori
- Sistemi di riferimento spaziali

In questa sezione vengono illustrati le risorse e i dati di riferimento che abilitano un database alle operazioni spaziali. Per informazioni relative ai geocoder non predefiniti, vedere la sezione "Informazioni sul processo di geocoding" a pagina 41. Per ulteriori informazioni relative ai sistemi di riferimento spaziali, vedere la sezione "Informazioni relative ai sistemi di riferimento spaziali e di coordinate" a pagina 25.

Dati di riferimento

I *dati di riferimento* comprendono gli indirizzi più recenti degli Stati Uniti, raccolti dall'ufficio di censimento. Prima che il geocoder predefinito converta un indirizzo del database in coordinate, questo indirizzo dovrà corrispondere in parte o completamente all'indirizzo contenuto nei dati di riferimento.

I dati di riferimento diventano disponibili dopo l'installazione di DB2 Spatial Extender. Per la quantità di spazio su disco richiesta per questi dati, vedere la sezione "Requisiti spazio su disco" a pagina 18. Verificare il corretto caricamento dei dati di AIX, all'interno dell'indirizzario `$DB2INSTANCE/sqlib/gse/refdata/`. Verificare il corretto caricamento dei dati di Windows all'interno dell'indirizzario `%DB2PATH%\gse\refdata\`.

Risorse che abilitano un database per le operazioni spaziali

La prima attività da eseguire dopo l'installazione DB2 Spatial Extender consiste nell'abilitazione del database per le operazioni spaziali. Ciò comprende l'inizializzazione di un'operazione che causa il caricamento da parte di DB2 Spatial Extender delle seguenti risorse:

- Procedure memorizzate. Quando si richiede un'operazione dal Centro controllo, DB2 Spatial Extender richiama una di queste procedure memorizzate per l'esecuzione dell'operazione.
- Tipi di dati spaziali. E' necessario assegnare un tipo di dati spaziali a ciascuna colonna di vista o di tabella in cui si desidera memorizzare i dati spaziali. Per ulteriori informazioni consultare "Informazioni relative ai tipi di dati spaziali" a pagina 33.
- Viste di tabelle e di catalogo di DB2 Spatial Extender. Alcune operazioni dipendono dal catalogo DB2 Spatial Extender. Ad esempio prima di riempire una colonna con tipi di dati spaziali, è necessario registrare la colonna nel catalogo come struttura. Per informazioni relative ai livelli, consultare il capitolo "Sviluppo e implementazione di un progetto GIS" a pagina 10.
- Un tipo di indice spaziale. Consente di definire gli indici per i livelli.
- Funzioni spaziali. E' possibile utilizzarle per gestire i dati spaziali in diversi modi; ad esempio per determinare i rapporti tra le funzioni geografiche e per generare più dati spaziali. Una di queste funzioni corrisponde al geocoder predefinito. Esso converte gli indirizzi degli Stati Uniti in coordinate, quindi inserisce le coordinate nelle colonne spaziali. Per ulteriori informazioni sulle funzioni spaziali, vedere il "Capitolo 13. Geometrie e funzioni spaziali associate" a pagina 129 e il "Capitolo 14. Funzioni spaziali per le interrogazioni SQL" a pagina 165. Per ulteriori informazioni sul geocoder predefinito, vedere la sezione "Informazioni sul processo di geocoding" a pagina 41.
- Uno schema, denominato DB2GSE, contenente gli oggetti appena elencati.

Per istruzioni su come utilizzare il Centro controllo per inizializzare il caricamento di tali risorse, vedere la sezione "Abilitazione di un database per le operazioni spaziali". Per informazioni sull'utilizzo di una routine in un programma applicativo per l'esecuzione di questa attività, vedere il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Abilitazione di un database per le operazioni spaziali

Per verificare il tipo di autorizzazione richiesto per l'abilitazione di un database per le operazioni spaziali, vedere la sezione "Autorizzazione" a pagina 77.

Per abilitare un database per le operazioni spaziali dal Centro di controllo:

1. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Database** nel server su cui si desidera che DB2 Spatial Extender venga eseguito.
2. Fare clic sulla cartella **Database**. I database vengono visualizzati nel pannello indice posto a destra della finestra.
3. Fare clic con il tastino destro del mouse sul database prescelto, quindi fare clic su **Spatial Extender** → **Abilitare** nel menu a comparsa. DB2 Spatial Extender fornisce il database con le risorse che consentono di creare e gestire i dati e le colonne spaziali.

Nota: Prima di abilitare un database per le operazioni spaziali, è necessario che DB2 Spatial Extender sia installato sul server in cui risiede il database.

Creazione di un sistema di riferimento spaziale

Questa sezione descrive le relazioni tra sistemi di riferimento spaziali e sistemi di coordinate e fornisce informazioni su come creare un sistema di riferimento spaziale dal Centro di controllo.

Informazioni relative ai sistemi di riferimento spaziali e di coordinate

In questa sezione si continua la discussione sui sistemi di coordinate, iniziata nella sezione "Struttura dei dati spaziali" a pagina 6. Si prosegue quindi con la definizione dei sistemi di riferimento spaziali fornita nella sezione "Sviluppo e implementazione di un progetto GIS" a pagina 10. Si forniscono inoltre informazioni per la determinazione dei valori da assegnare ai parametri di un sistema di riferimento spaziale.

Sistemi di coordinate, coordinate e misure

Un sistema di coordinate può essere paragonato a una griglia che ricopre una determinata area geografica. Ad esempio si può utilizzare una griglia che ricopre la terra, una griglia che ricopre una nazione o una griglia che ricopre una regione di uno stato. Ciascuna funzione geografica di un'area è posizionata sul punto di intersezione di una linea della griglia est-ovest e di una linea della griglia nord-sud. Un valore denominato *coordinata X*, indica l'ubicazione del punto sulla linea della griglia est-ovest. Un altro valore, denominato *coordinata Y*, indica l'ubicazione del punto sulla linea della griglia nord-sud. Entrambi i riferiscono l'ubicazione al centro della griglia o all'*origine*.

Le coordinate X e Y all'origine corrispondono a zero. Dall'origine verso est, le coordinate X sono positive; dall'origine verso ovest, sono negative. Allo stesso modo, dall'origine verso nord, le coordinate Y sono positive; dall'origine verso sud, sono negative. Per comprendere questa distribuzione, si consideri il seguente esempio generalizzato: Il sistema di coordinate A include una griglia che ricopre un'elevata area metropolitana. Una coordinata X di valore 7,

denota una posizione di sette unità di misura a est dall'origine di questa griglia. Una coordinata X di valore -9.5, denota una posizione di nove e mezzo unità di misura a ovest dall'origine.

Ciascuna voce di dati di una colonna spaziale, include (1) una coordinata X e una coordinata Y che definiscono l'ubicazione di una funzione geografica o (2) più coordinate X e Y che definiscono le ubicazioni di varie parti di una funzione o che definiscono l'area ricoperta da una funzione. E' inoltre possibile includere due altri tipi di valori, una *coordinata Z* e una *misura*. Diversamente dalle coordinate X e Y, le coordinate Z e le misure non vengono utilizzate in DB2 Spatial Extender per definire le ubicazioni o le aree. Esse vengono infatti utilizzate per trasmettere le informazioni richieste da un'applicazione GIS. Le coordinate A Z indicano di solito l'altezza o la profondità di una funzione geografica. Le coordinate Z poste al di sopra dell'origine sono positive; le coordinate Z poste al di sotto dell'origine sono negative. Le misure sono valori numerici; possono trasmettere qualunque tipo di informazioni. Ad esempio, se si sta rappresentando pozzi petroliferi nel proprio GIS ed è necessario che le applicazioni elaborino valori che denotano ID di punti sequenze per i dati sismici, sarà possibile memorizzare questi valori come misure.

Sistemi di riferimento spaziale, offset e fattori di scala

Come indicato nella sezione "Sistemi di coordinate, coordinate e misure" a pagina 25, le coordinate possono essere valori negativi decimali. La stessa regola è valida per le misure. Tuttavia per ridurre un sovraccarico di memoria, DB2 Spatial Extender memorizza ciascuna coordinata e misura come numero intero non negativo (vale a dire come numero intero positivo o zero). Sarà necessario pertanto convertire le misure e le coordinate decimali negative attuali in numeri interi non negativi, per consentirne la memorizzazione da parte di DB2 Spatial Extender. E' inoltre necessario comunicare a DB2 Spatial Extender le modalità di esecuzione della conversione. A tale scopo è necessario impostare determinati parametri. L'impostazione dei parametri che è necessario utilizzare per convertire le coordinate e le misure in una specifica area geografica vengono denominate collettivamente *sistema di riferimento spaziale*.

Per creare un sistema di riferimento spaziale, è necessario eseguire le operazioni riportate di seguito:

- Determinazione delle misure e delle coordinate negative inferiori per le funzioni rappresentate. I valori negativi decrescono all'aumentare della relativa distanza dallo zero. Una coordinata X di -10 è inferiore a una coordinata X di -5; una misura di -100 è inferiore a una misura di -50.
- Specifica di *fattori offset* (o *offset*): valori che, se sottratti da coordinate o misure negative, restituiscono numeri non negativi.

- Specifica di *fattori di scala*: valori che, se moltiplicati per coordinate e misure decimali, producono numeri interi con precisione almeno corrispondente a quella della coordinate o delle misure. Si consideri ad esempio una coordinata con precisione quattro: 92.77. E' possibile moltiplicare questo valore per un fattore di scala di 100 per ottenere un numero intero con precisione quattro: 9277.

Determinazione delle misure e delle coordinate negative inferiori

Prima di impostare i parametri per un sistema di riferimento spaziale, è necessario determinare la coordinata X, la coordinata Y, la coordinata Z e la misura negative inferiori nell'area geografica contenente le funzioni su cui si desidera richiamare informazioni. E' possibile comprendere questi valori rispondendo acquisendo le seguenti informazioni:

- Se tra le funzioni rappresentate, ve ne sono alcune poste a ovest del sistema di coordinate di origine utilizzato. In caso positivo, è necessario comprendere quale coordinata X indica l'ubicazione o l'estremità occidentale della funzione posta nel punto occidentale massimo. Questo valore corrisponderà al valore più basso tra le coordinate X negative utilizzate. Ad esempio, se si sta rappresentando pozzi petroliferi e alcuni di essi si trovano ad ovest dell'origine, quale coordinata X indica la posizione del pozzo petrolifero all'estremo ovest?
- Se sono presenti funzioni a sud dell'origine. In caso positivo, è necessario comprendere quale coordinata Y indica l'ubicazione o l'estremità meridionale della funzione posta nel punto meridionale massimo. Questo valore corrisponderà al valore più basso tra le coordinate Y negative utilizzate. Ad esempio, se si sta rappresentando pozzi petroliferi e alcuni di essi si trovano a sud dell'origine, quale coordinata Y indica la posizione del pozzo petrolifero all'estremo sud?
- Se si intende utilizzare le coordinate Z per definire la profondità, è necessario stabilire qual'è la funzione più profonda e quale coordinata Z rappresenta il punto più basso di tale funzione. Questo valore corrisponderà al valore più basso tra le coordinate Z negative utilizzate.
- Se si intende includere alcune misure nei dati spaziali, verificare se sono valori negativi. In caso positivo, è necessario stabilire quali sono le misure più basse e più alte.

Verificare quali sono le misure e le coordinate negative inferiori, aggiungere a ciascuna di esse una quantità corrispondente al dieci per cento del relativo valore. Se ad esempio la coordinata X inferiore negative corrisponde a -100, è possibile aggiungere -5 ad essa. Questo manuale richiama la figura risultante dopo un *valore aumentato*.

Specifica di fattori offset

Successivamente, specificare i fattori offset che DB2 Spatial Extender dovrà utilizzare per convertire coordinate e misure da valori negativi a valori non negativi:

- Dopo aver stabilito l'incremento che si desidera assegnare al valore X, specificare un offset che, se sottratto da questo valore, restituisca zero. DB2 Spatial Extender sottrarrà quindi questo numero da tutte le coordinate X negative fino a raggiungere un valore positivo. DB2 Spatial Extender sottrarrà inoltre questo valore da tutte le altre coordinate X.

Se ad esempio il valore X incrementato è -105, è necessario sottrarre -105 per ottenere 0. DB2 Spatial Extender sottrarrà quindi -105 da tutte le coordinate X associate alle funzioni rappresentate. Poiché nessuna di queste coordinate è maggiore di -100, tutti i valori risultanti dalla sottrazione saranno positivi.

- Allo stesso modo gli offset che restituiscono 0 quando vengono sottratti dal valore Y incrementato, dal valore Z incrementato e dalla misura incrementato.

L'offset sottratto dalle coordinate X viene denominato *false X*. Gli offset sottratti dalle coordinate Y, dalle coordinate Z e dalle misure, vengono denominate rispettivamente *false Y*, *false Z* e *false M*. Per informazioni sulla specifica di tali parametri dal Centro di controllo, vedere la sezione "Creazione di un sistema di riferimento spaziale dal Centro di controllo" a pagina 29.

Specifica di fattori di scala

Successivamente, specificare i fattori di scala che DB2 Spatial Extender dovrà utilizzare per convertire coordinate e misure da numeri decimali a numeri interi:

- Specificare un fattore di scala che, se moltiplicato per una coordinata X decimale o per una coordinata Y decimale, restituisce un numero intero a 32 bit. Si consiglia di assegnare a questa scala di fattori un fattore di 10: 10 alla prima potenza (10), 10 alla seconda potenza (100), 10 alla terza potenza (1000) o, se necessario, un fattore più elevato. Per stabilire a quale fattore 10 dovrà corrispondere il fattore di scala:
 1. Stabilire quali coordinate X e Y corrispondono esattamente o approssimativamente a un numero decimale. Si supponga ad esempio che tra le varie coordinate X e Y utilizzate, tre sono quelle corrispondenti a numeri decimali: 1,23, 5,1235 e 6,789.
 2. Utilizzare la coordinata decimale con precisione decimale più lunga. Stabilire quindi per quale fattore di dieci è possibile moltiplicare questa coordinata, per ottenere come risultato un numero intero di uguale precisione. Tra le tre coordinate decimali illustrate nell'esempio, 5,1235 è quella con precisione decimale più lunga. Moltiplicando questo numero per dieci elevato a quattro (10000), il risultato sarà 51235, che corrisponde a un numero intero.
 3. Stabilire se il numero intero prodotto dalla moltiplicazione appena eseguita, è troppo elevato per essere memorizzato come voce di dati a 32 bit. 51235 non è troppo elevato. Si supponga comunque che oltre ai

valori 1,23, 5,11235 e 6,789, l'intervallo di coordinate X e Y includa un quarto valore decimale, 10006,789876. Poiché la precisione decimale di questa coordinata è più lunga di quella della precisione dei primi tre numeri, sarà necessario moltiplicare *questa* coordinata—non 5.11235—per un fattore di 10. Per convertirlo in un numero intero, è possibile moltiplicarlo per 10 elevato a sei (1000000). Il valore risultante, 10006789876, è tuttavia troppo lungo per essere memorizzato come voce di dati a 32 bit. Se DB2 Spatial Extender tenterà di memorizzarlo, potrebbero verificarsi risultati imprevisti.

Per evitare questo problema, selezionare un fattore di 10 che, se moltiplicato per la coordinata originale, restituisce un numero decimale che DB2 Spatial Extender potrà ridurre a un numero intero memorizzabile, con perdita minima di precisione. In tal caso, è possibile selezionare 10 elevato a quattro (10000). Moltiplicando 10000 per 10006,789876 il risultato sarà 100067898,76. DB2 Spatial Extender ridurrà questo numero a 100067898, riducendo la precisione di una quantità irrilevante.

- Se le funzioni rappresentate comprendono coordinate Z decimali, seguire la relativa procedura per utilizzare un fattore di scala per queste coordinate. Se le funzioni sono associate a misure decimali, seguire la stessa procedura per utilizzare un fattore di scala per queste misure.

Il fattore di scala per le coordinate X e Y viene denominato *unità XY*. I fattori di scala per le misure e le coordinate Z vengono denominati rispettivamente *unità Z* e *unità M*. Per informazioni sulla specifica di tali parametri dal Centro di controllo, vedere la sezione "Creazione di un sistema di riferimento spaziale dal Centro di controllo".

Creazione di un sistema di riferimento spaziale dal Centro di controllo

Questa sezione fornisce una panoramica dei passi necessari per la creazione di un sistema di riferimento spaziale dal Centro di controllo. Alla panoramica seguono le informazioni relative al completamento di ciascun passo.

Per l'esecuzione di questi passi non è richiesta alcuna autorizzazione.

Panoramica dei passi necessari per la creazione di un sistema di riferimento spaziale dal Centro di controllo:

1. Aprire la finestra Creare riferimento spaziale.
2. Indicare il sistema di coordinate che si desidera utilizzare.
3. Specificare gli identificativi per il sistema di riferimento spaziale che si desidera creare.
4. Determinare gli intervalli di coordinate e misure da applicare alle funzioni geografiche di cui si desidera richiamare informazioni.

5. Specificare i valori che è possibile utilizzare per convertire coordinate e misure negative o decimali in voci di dati memorizzabili da parte di DB2 Spatial Extender.
6. Impostare DB2 Spatial Extender per la creazione del sistema di riferimento spaziale prescelto.

Passi dettagliati per la creazione di un sistema di riferimento spaziale dal Centro di controllo:

1. Aprire la finestra Creare riferimento spaziale.
 - a. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Database** nel server su cui si desidera che DB2 Spatial Extender venga eseguito.
 - b. Fare clic sulla cartella **Database**. I database vengono visualizzati nel pannello indice posto a destra della finestra.
 - c. Fare clic con il tastino destro del mouse sul database abilitato per i dati spaziali, quindi fare clic su **Spatial Extender** —> **Riferimenti spaziali** nel menu a comparsa. Viene aperta la finestra Riferimenti spaziali.
 - d. Dalla finestra Riferimenti spaziali, fare clic su **Creare**. Viene aperta la finestra Creare riferimento spaziale.
2. Dalla finestra Creare riferimento spaziale, utilizzare il campo **Sistema di coordinate** per indicare il sistema di coordinate che si desidera utilizzare.
3. Specificare gli identificativi per il sistema di riferimento spaziale che si desidera creare.
 - Nel campo **Nome**, digitare un nome sistema di lunghezza compresa tra 1 e 64 caratteri.

Limitazione: Non specificare il nome di un sistema di riferimento spaziale diverso. Non è possibile specificare lo stesso nome per due sistemi di riferimento spaziale del database.
 - Nel campo **ID**, digitare un identificativo numerico. E' necessario utilizzare un numero intero.

Limitazione: Non specificare l'ID di un sistema di riferimento spaziale diverso. Non è possibile specificare lo stesso ID per due sistemi di riferimento spaziale nel database.
4. L'utilizzo di un supporto esterno al Centro di controllo—da esempio, un foglio di carta o una lavagna bianca—determina le coordinate o le misure minime negative applicabili alla funzione geografica rappresentata. Per le relative istruzioni, consultare la sezione "Determinazione delle misure e delle coordinate negative inferiori" a pagina 27.

5. Dalla finestra Creare riferimento spaziale, specificare i valori per la conversione di coordinate e misure negative o decimali in voci di dati supportati da DB2 Spatial Extender, vale a dire numeri interi non negativi a 32 bit.
- a. Specificare i valori per la conversione di coordinate X negative o decimali in numeri interi non negativi:
- Nella colonna **Offset**, nel campo adiacente alla **X**, specificare false X:
 - Se sono presenti valori negativi nell'intervallo di coordinate X identificate nel passo 4 a pagina 30, immettere un false X che, se sottratto dalla coordinata negativa più bassa, restituisca un numero positivo. Per istruzioni, vedere la sezione "Specifica di fattori offset" a pagina 27.
 - Se nessuna coordinata X è negativa, immettere un false X di 0.
 - Nella colonna **Fattore di scala**, specificare un'unità XY nel campo posto nell'estremità destra di **X**. Queste unità XY dovranno essere tali che, se moltiplicate per una coordinata X decimale o per una coordinata Y decimale, restituisce un numero intero che è possibile memorizzare come voce di dati a 32 bit, con una riduzione minima della precisione. Per istruzioni, vedere la sezione "Specifica di fattori di scala" a pagina 28.
- Dopo aver specificato l'unità XY nel campo posto all'estremità destra di **X**, essa verrà visualizzata anche nel campo posto all'estremità destra di **Y**.
- b. Specificare un false Y che consenta a DB2 Spatial Extender di convertire coordinate Y negative in valori positivi. A tale scopo utilizzare la colonna **Offset** nel campo adiacente la **Y**:
- Se sono presenti valori negativi nell'intervallo di coordinate Y identificate nel passo 4 a pagina 30, immettere un false Y che, se sottratto dalla coordinata negativa più bassa, restituisca un numero positivo. Per istruzioni, vedere la sezione "Specifica di fattori offset" a pagina 27.
 - Se tutte le coordinate Y sono positive, immettere un false Y di 0.
- c. Se si intende includere coordinate Z nei dati spaziali, specificare i valori per la conversione di coordinate Z negative o decimali in numeri interi non negativi:
- Nella colonna **Offset**, nel campo adiacente alla **Z**, digitare false Z:
 - Se sono presenti valori negativi nell'intervallo di coordinate Z identificate nel passo 4 a pagina 30, immettere un false Z che, se sottratto dalla coordinata negativa più bassa, restituisca un numero positivo. Per istruzioni, vedere la sezione "Specifica di fattori offset" a pagina 27.
 - Se nessuna coordinata Z è negativa, immettere un false Z di 0.

- Nella colonna **Fattore di scala**, specificare un'unità Z nel campo posto nell'estremità destra di **Z**. Questa unità Z dovrà essere tale che, se moltiplicata per una coordinata Z decimale, restituisce un numero intero che è possibile memorizzare come voce di dati a 32 bit, con una riduzione minima della precisione. Per istruzioni, vedere la sezione "Specifica di fattori di scala" a pagina 28.
- d. Se si intende includere misure nei dati spaziali, specificare i valori per la conversione di misure negative o decimali in numeri interi positivi:
- Nella colonna **Offset**, nel campo adiacente all'etichetta **Lineare**, digitare un false M:
 - Se sono presenti valori negativi nell'intervallo di misure identificate nel passo 4 a pagina 30, immettere un false M che, se sottratto dalla misura negativa più bassa, restituisca un numero positivo. Per istruzioni, vedere la sezione "Specifica di fattori offset" a pagina 27.
 - Se tutte le misure sono positive, immettere un false M di 0.
 - Nella colonna **Fattore di scala**, specificare un'unità M nel campo posto all'estremità destra dell'etichetta **Lineare**. Questa unità M dovrà essere tale che, se moltiplicata per una misura decimale, restituisca un numero intero che è possibile memorizzare come voce di dati a 32 bit, con una riduzione minima della precisione. Per istruzioni, vedere la sezione "Specifica di fattori di scala" a pagina 28.
6. Fare clic su **OK** per creare il sistema di riferimento spaziale prescelto.

Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione

Dopo aver impostato le risorse per il GIS DB2 Spatial Extender, sarà possibile creare oggetti contenenti dati spaziali. Se ad esempio è necessario utilizzare nuove tabelle, è possibile definirle, assegnare tipi di dati spaziali alle colonne in cui si desidera includere i dati. Effettuare questa operazione anche per aggiungere colonne spaziali alle tabelle esistenti.

Quando si immette una tabella nuova o esistente con una colonna spaziale, è necessario registrare la colonna come struttura. Inoltre se si intende riempire la colonna tramite il geocoder, è possibile, al momento della registrazione delle colonne come strutture, abilitare il geocoder alla gestione automatica delle stesse. Questa abilitazione viene effettuata nei seguenti modi: DB2 Spatial Extender definisce trigger codificati per il richiamo del geocoder quando una colonna di attributo corrispondente alla colonna (o colonne) riceve dati nuovi o aggiornati. Quando viene richiamato, geocoder traduce i dati nuovi o aggiornati in dati spaziali e li inserisce nella colonna spaziale.

Dopo aver definito una colonna spaziale per una tabella, è possibile facoltativamente scegliere di creare una colonna di visualizzazione su questa colonna di tabella. E' necessario registrare la colonna di visualizzazione come struttura dopo aver registrato la colonna di tabella come struttura.

Questo capitolo descrive la struttura e l'utilizzo dei tipi di dati che è possibile assegnare alla colonna spaziale. Successivamente, vengono fornite indicazioni su come utilizzare il Centro di controllo per definire una colonna spaziale per una tabella, per registrare la colonna come struttura e per abilitare un geocoder per la relativa gestione. Infine si descrive come utilizzare il Centro di controllo per registrare la colonna di visualizzazione come struttura.

Informazioni relative ai tipi di dati spaziali

Questa sezione introduce i tipi di dati richiesti per le colonne spaziali e fornisce un supporto nella scelta delle funzioni da assegnare a un tipo di dati di una colonna spaziale.

Quando si abilita un database per le operazioni spaziali, DB2 Spatial Extender fornisce il database con una gerarchia dei tipi di dati strutturati. La gerarchia viene illustrata nella Figura 6 a pagina 34. In questa figura, i tipi istanziabili

presentano uno sfondo bianco; i tipi non istanziabili presentano uno sfondo ombreggiato.

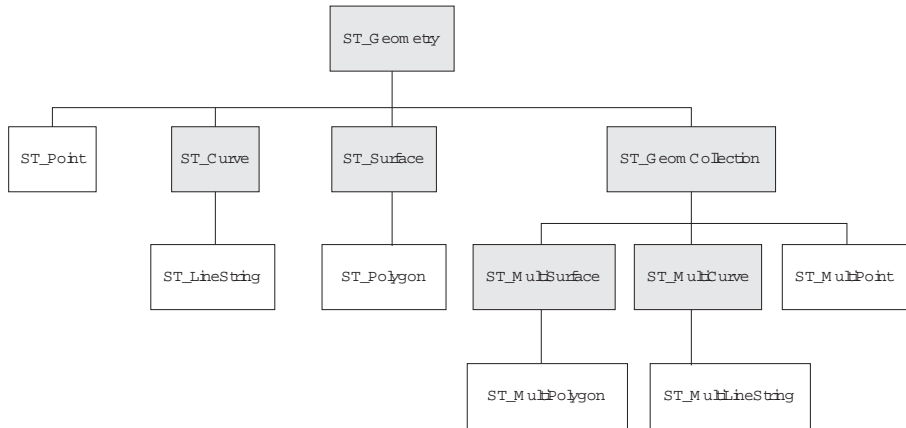


Figura 6. Gerarchia dei tipi di dati spaziali. I tipi di dati compresi nelle caselle bianche sono istanziabili. I tipi di dati compresi nelle caselle ombreggiate non sono istanziabili.

La gerarchia rappresentata nella Figura 6 include:

- Tipi di dati per le funzioni geografiche che è possibile assumere come singola unità; ad esempio residenze individuali e laghi isolati.
- Tipi di dati per le funzioni geografiche composti da più unità o componenti; ad esempio, i sistemi autostradali e le strade di montagna.
- Un tipo di dati per le funzioni geografiche di tutti i tipi.

Tipi di dati per funzioni di unità singola

Utilizzare ST_Point, ST_Linestring e ST_Polygon per memorizzare coordinate che definiscono lo spazio occupato dalle funzioni che è possibile assumere come singole unità:

- Utilizzare ST_Point per indicare il punto dello spazio occupato da un funzione geografica discreta. La funzione può essere di dimensioni molto ridotte, come un pozzo idrico, oppure di dimensioni elevate, come una città; oppure di dimensione intermedia, come un complesso di edifici o un parco. In ogni caso, il punto dello spazio può essere posizionato nel punto di intersezione di una coordinata est-ovest (ad esempio un parallelo) e una coordinata nord-sud (ad esempio un meridiano). Una voce di dati ST_Point include valori come una coordinata X e una coordinata Y che definiscono questo tipo di intersezione. La coordinata X indica il punto di intersezione sulla linea est-ovest; la coordinata Y indica il punto di intersezione sulla linea nord-sud.
- Utilizzare ST_Linestring per le coordinate che definiscono lo spazio occupato dalle funzioni lineari; ad esempio le strade e i canali.

- Utilizzare ST_Polygon se si desidera indicare la quantità di spazio coperta da una funzione su più siti; ad esempio un distretto pubblico, una foresta o una zona popolata da animali selvaggi. Una voce di dati ST_Polygon comprende le coordinate che definiscono il perimetro di questa funzione.

In alcuni casi è possibile utilizzare ST_Polygon e ST_Point per le stesse funzioni. Si supponga ad esempio che si desideri richiamare informazioni spaziali relative a diversi complessi residenziali. Per rappresentare il punto dello spazio in cui è ubicato ciascun complesso, sarà necessario utilizzare ST_Point per memorizzare le coordinate X e Y che definiscono tale punto. D'altra parte se si desidera rappresentare l'area coperta da ciascun complesso, è consigliabile utilizzare ST_Polygon per memorizzare le coordinate che definiscono il perimetro di ciascuna area di questo tipo.

Tipi di dati per funzioni a più unità

Utilizzare ST_MultiPoint, ST_MultiLineString e ST_MultiPolygon per memorizzare le coordinate che definiscono gli spazi occupati dalle funzioni composte da più unità:

- Utilizzare ST_MultiPoint per rappresentare le funzioni composte da unità discrete e per indicare il punto dello spazio occupato da ciascun componente. Una voce di dati ST_MultiPoint include le coppie di coordinate X e Y che definiscono l'ubicazione di ciascun componente come le funzioni. Si consideri, ad esempio una tabella contenente righe che rappresentano arcipelaghi e colonne che includono una colonna ST_MultiPoint. Ciascuna voce di dati di questa colonna include le coppie di coordinate X e Y che definiscono le ubicazioni delle isole in ciascun arcipelago.
- Utilizzare ST_MultiLineString per rappresentare le funzioni composte da unità lineari e per richiamare informazioni relative allo spazio occupato da ciascuna unità. Una voce di dati ST_MultiLineString comprende le coordinate che definiscono questo spazio. Si consideri, ad esempio una tabella contenente righe che rappresentano sistemi di fiumi e colonne che includono una colonna ST_MultiLineString. Ciascuna voce di dati di questa colonna include le serie di coordinate che definiscono i corsi dei fiumi di ciascun sistema.
- Utilizzare ST_MultiPolygon per rappresentare le funzioni composte da unità a più siti e per richiamare informazioni relative allo spazio occupato da ciascuna unità. Si consideri, ad esempio una tabella contenente righe che rappresentano le regioni occidentali e colonne che includono una colonna ST_MultiPolygon. Questa colonna contiene informazioni relative alle zone agricole. In particolare ciascuna voce di dati di questa colonna include le serie di coordinate che definiscono i perimetri delle zone agricole di una determinata regione.

Un tipo di dati per tutte le funzioni

E' possibile utilizzare ST_Geometry se non si è certi degli altri tipi di dati da utilizzare. Poiché ST_Geometry è la radice della gerarchia da cui dipendono gli altri tipi di dati, una colonna è possibile memorizzare in ST_Geometry uno o tutti i valori che è possibile memorizzare nelle colonne a cui vengono assegnati gli altri tipi di dati.

Attenzione: Il geocoder predefinito può convertire gli indirizzi in voci di dati ST_Point o ST_Geometry. Se si intende pertanto utilizzare questo geocoder per compilare una colonna spaziale, è necessario assegnare il tipo di dati ST_Point o ST_Geometry a questa colonna.

Definizione di una colonna spaziale per una tabella, registrazione della colonna come struttura e abilitazione di un geocoder alla gestione della colonna

Questa sezione fornisce una panoramica dei passi richiesti per la definizione di una colonna spaziale per una tabella, per la registrazione della colonna come struttura e per l'abilitazione di un geocoder per la relativa gestione. Alla panoramica seguono le informazioni relative completamento di ciascun passo.

Per verificare le autorizzazioni necessarie per la registrazione di una colonna della tabella come struttura, consultare la sezione "Autorizzazione" a pagina 93. Per verificare le autorizzazioni necessarie per l'abilitazione di un geocoder per la gestione di questa colonna, consultare la sezione "Autorizzazione" a pagina 74.

Panoramica dei passi richiesti per la definizione di una colonna spaziale per una tabella, per la registrazione della colonna come struttura e per l'abilitazione di un geocoder per la relativa gestione:

1. Se la colonna spaziale dovrà essere compresa in una nuova tabella, creare la tabella.
2. Aprire la finestra Creare struttura spaziale.
3. Aggiungere una colonna spaziale a una tabella e indicare che si desidera registrare la colonna come struttura; oppure indicare che si desidera registrare una colonna esistente come struttura.
4. Indicare il sistema di riferimento spaziale che si desidera utilizzare per la struttura.
5. Se la struttura dovrà contenere dati importati o i dati generati da un'altra colonna spaziale, comunicare a DB2 Spatial Extender di creare la struttura.
6. Se la struttura dovrà contenere dati derivati da dati di attributo:
 - a. Specificare la colonna o le colonne che dovranno contenere questi dati di attributo.

- b. Indicare che si desidera abilitare un geocoder alla gestione della struttura.
- c. Comunicare a DB2 Spatial Extender di creare la struttura.

Passi dettagliati per la definizione di una colonna spaziale per una tabella, per la registrazione della colonna come struttura e per l'abilitazione di un geocoder per la relativa gestione:

1. Se la colonna spaziale dovrà essere compresa in una nuova tabella, creare la tabella:
 - Utilizzare l'interfaccia desiderata (ad esempio il Centro di controllo o un processore riga comandi) per creare la tabella.
 - Se si desidera utilizzare un geocoder, includere da una a dieci colonne su cui il geocoder dovrà operare. Il geocoder non può utilizzare più di dieci colonne di dati come input.
 - Includere la colonna spaziale che si intende registrare come struttura oppure definire questa colonna nel passo 3.

Se si desidera utilizzare una tabella esistente, andare al passo successivo.

2. Aprire la finestra Creare struttura spaziale.
 - a. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Tabelle** per le tabelle del database utilizzate per le operazioni spaziali.
 - b. Fare clic sulla cartella **Tabelle**. Le tabelle vengono visualizzate nel pannello indice posto a destra della finestra.
 - c. Fare clic con il tastino destro del mouse sulla tabella prescelta e fare clic su **Spatial Extender** —> **Strutture spaziali** nel menu a comparsa. Viene aperta la finestra Strutture spaziali.
 - d. Dalla finestra Strutture spaziali, fare clic su **Creare**. Viene aperta la finestra Creare struttura spaziale.
3. Dalla finestra Creare struttura spaziale, aggiungere una colonna spaziale a una tabella e indicare che si desidera registrare la colonna come struttura; oppure indicare che si desidera registrare una colonna esistente come struttura.
 - Se si desidera aggiungere una colonna spaziale a una tabella e definire questa colonna come struttura:
 - a. Nel campo **Colonna struttura**, digitare un nome per la colonna.
 - b. Nel campo **Tipo colonna**, selezionare o immettere il tipo di dati che si desidera inserire nella colonna. Per una descrizione dei tipi di dati disponibili, vedere la sezione "Informazioni relative ai tipi di dati spaziali" a pagina 33.
 - Per definire una colonna esistente come struttura, selezionarla nel campo **Colonna struttura**.

Limitazione: Non selezionare le colonne che sono già state definite come strutture.

4. Nel campo **Nome riferimento spaziale**, specificare il nome del sistema di riferimento spaziale da utilizzare per la struttura.
5. Se la struttura dovrà contenere dati importati o dati generati da un'altra colonna spaziale, fare clic su **OK** per registrarla.
6. Se si desidera che la struttura contenga dati derivati da dati di attributo:
 - a. Specificare la colonna o le colonne che dovranno contenere questi dati di attributo:
 - 1) Selezionare la colonna o le colonne nella casella **Colonne disponibili**. È possibile selezionare massimo dieci colonne.
 - 2) Fare clic sul pulsante **>**, sul pulsante **>>** o entrambi, per visualizzare la colonna o le colonne selezionate nella casella **Colonne selezionate**.
 - b. Se si desidera abilitare un geocoder alla gestione della struttura:
 - 1) Selezionare la casella di spunta **Abilitare geocoder automatico**.
 - 2) Nel campo **Nome**, selezionare il nome del geocoder che si desidera utilizzare.
 - 3) Nel campo **Livello di precisione**, specificare in percentuale il grado di corrispondenza tra i record di input e i record dei dati di riferimento per l'elaborazione. Questa percentuale è denominata *precisione*. Si supponga ad esempio che il geocoder legge un record input contenente l'indirizzo 557 Bailey, San Jose 94120. Se la precisione è 100 e la corrispondenza tra questo indirizzo e l'indirizzo contenuto nei dati di riferimento non è del 100 per cento, l'indirizzo verrà rifiutato dal geocoder. Se la precisione è 75 e la corrispondenza tra i due indirizzi raggiunge almeno il 75 per cento, il geocoder inizierà l'elaborazione.
 - 4) Se il geocoder è stato fornito da un'altro produttore, utilizzare la casella **Proprietà** per specificare i parametri di geocoding forniti dal produttore che si desidera utilizzare.
 - c. Fare clic su **OK** per registrare la colonna selezionata come struttura e, se richiesto, abilitare il geocoder alla gestione della colonna.

Registrazione della colonna della vista come struttura

Per verificare le autorizzazioni necessarie per la registrazione di una colonna della vista come struttura, consultare la sezione "Autorizzazione" a pagina 93.

Per registrare la colonna della vista come struttura:

1. Aprire la finestra Creare struttura spaziale.

- a. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Viste** per le viste del database utilizzate per le operazioni spaziali.
 - b. Fare clic sulla cartella **Viste**. Le viste vengono visualizzate nel pannello indice posto a destra della finestra.
 - c. Fare clic con il tastino destro del mouse sulla vista prescelta e fare clic su **Spatial Extender** → **Strutture spaziali** nel menu a comparsa. Viene aperta la finestra Strutture spaziali.
 - d. Dalla finestra Strutture spaziali, fare clic su **Creare**. Viene aperta la finestra Creare struttura spaziale.
2. Utilizzare la colonna **Colonna struttura** per specificare la colonna che si desidera registrare come struttura.
 3. Nel campo **Struttura spaziale sottostante**, specificare il nome della colonna di tabella su cui è basata la colonna di vista selezionata. E' necessario che questa colonna di tabella sia già registrata come struttura.
 4. Fare clic su **OK** per registrare la colonna della vista specificata come struttura.

Capitolo 5. Riempimento delle colonne spaziali

Dopo aver registrato le colonne spaziali come livelli, è possibile inserirvi dati spaziali. Come si è visto per “Provenienza dei dati spaziali” a pagina 6, esistono tre modalità per fornire questi dati: utilizzare una funzione, definita geocoder, per ricavarli dai dati di attributo; utilizzare altre funzioni, per ricavarli da altri dati spaziali o importarli dai file. Il presente capitolo:

- Descrive la funzione geocoding e le modalità di utilizzo del Centro di controllo per eseguire la funzione geocode sui dati di attributo in modalità batch
- Descrive l’importazione e l’esportazione dei dati e le modalità di utilizzo del Centro di controllo per l’importazione e l’esportazione dei dati da GIS

Per ulteriori informazioni sulle funzioni, che possono essere utilizzate per ricavare nuovi dati spaziali da dati spaziali esistenti, vedere “Le funzioni che generano nuove geometrie da quelle esistenti” a pagina 155.

Utilizzo di funzioni geocoder

Questa sezione descrive il processo di geocoding e le modalità di esecuzione di un geocoder in modalità batch dal Centro di controllo.

Informazioni sul processo di geocoding

Questa sezione delinea le differenze di base tra i geocoder e le rispettive origini. Descrive inoltre le due modalità di funzionamento di geocode e introduce i fattori da tenere presente quando si utilizza un geocoder.

Con DB2 Spatial Extender, è possibile:

- Utilizzare il geocoder predefinito, fornito con DB2 Spatial Extender.
- Inserire i geocoder sviluppati da altri fornitori.
- Inserire geocoder propri.

Il geocoder predefinito esegue il processo di geocoding sugli indirizzi degli Stati Uniti e può convertirli in dati ST_Point o ST_Geometry. Se si desidera memorizzare i dati di altri tipi di dati spaziali, è possibile inserire un geocoder, per generare questo tipo di dati. Se sono richiesti dati spaziali che rappresentino località non statunitensi o che non presentano indirizzi —ad esempio, zone agricole il cui suolo varia— è possibile utilizzare un geocoder che funga allo scopo.

Prima di poter utilizzare un geocoder plug-in, è necessario registrarlo. Gli utenti e i fornitori possono registrarlo con la procedura memorizzata db2gse.gse_register_gc. Non è possibile eseguire la registrazione dal Centro di

controllo. Per informazioni su `db2gse.gse_register_gc`, vedere “`db2gse.gse_register_gc`” a pagina 91. Per informazioni di carattere generale sull'utilizzo delle procedure memorizzate DB2 Spatial Extender, vedere “Capitolo 9. Procedure memorizzate” a pagina 67.

Un geocoder opera in due modalità:

- In *modalità batch*, il geocoder tenta, con una sola operazione, di convertire tutti i dati di origine esistenti per una colonna spaziale in dati spaziali e di riempire la colonna con i dati in questione. E' possibile avviare questa operazione dalla finestra Eseguire geocoder. In alternativa, è possibile avviare l'operazione in un programma di applicazione, codificando il programma per richiamare la procedura memorizzata `db2gse.gse_run_gc`.
- In *modalità incrementale*, un geocoder converte i dati, al momento dell'inserimento o dell'aggiornamento in una tabella, immettendo i valori spaziali che risultano dall'operazione in una colonna, per mantenere la colonna aggiornata. Viene attivato da trigger di inserimento o di aggiornamento, che è possibile richiedere dalla finestra Creare struttura spaziale. In alternativa, è possibile richiederli in un programma di applicazione, codificando il programma per richiamare la procedura memorizzata `db2gse.gse_enable_autogc`.

Il geocoding incrementale viene definito anche *geocoding automatico*.

Quando si utilizza un geocoder, è opportuno considerare i seguenti fattori:

1. Quando si utilizza il Centro di controllo, in genere ci si serve della finestra Creare struttura spaziale, prima della finestra Eseguire geocoder. Questo indica che è possibile disporre di trigger di impostazione DB2 Spatial Extender per il geocoding incrementale, prima di avviare il geocoding in modalità batch. Quindi, è possibile che il geocoding incrementale preceda quello in modalità batch. Elaborando tutti i dati di origine in modalità batch, il geocoder sottoporrà a geocoding gli stessi dati su cui opera in modalità incrementale. Questa ridondanza non genererà duplicati (quando i dati spaziali vengono generati due volte, i dati prodotti dopo sovrascriveranno i primi). Tuttavia, questo processo può abbassare il livello delle prestazioni. Un modo per evitare che questo avvenga consiste nel differire l'impostazione dei trigger fino al termine dell'esecuzione del geocoding in modalità batch.
2. Se i trigger sono già impostati quando si è pronti a eseguire il geocoding in modalità batch, è consigliabile disattivarli fino al termine del processo di geocoding in modalità batch. E' possibile disattivare i trigger dalla finestra Eseguire geocoder o in un programma di applicazione, codificando il programma in modo da richiamare la procedura memorizzata `db2gse.gse_disable_autogc`. Se si utilizza la finestra Eseguire geocoder, DB2 Spatial Extender riattiva i trigger automaticamente al termine del processo di geocoding. Se si utilizza la procedura memorizzata

db2gse.gse_disable_autogc, è possibile riattivare i trigger, richiamando la procedura memorizzata db2gse.gse_enable_autogc.

3. Se si desidera eseguire un geocoder in modalità batch, per riempire una colonna spaziale che presenta un indice, prima di tutto disattivare o rilasciare l'indice. Altrimenti, se l'indice resta operativo durante l'esecuzione del geocoder, le prestazioni ne risulteranno compromesse. Se si sta utilizzando il Centro di controllo, è possibile disattivare l'indice dalla finestra Eseguire geocoder. DB2 Spatial Extender riattiva automaticamente l'indice al termine del processo di geocoding. Se si sta utilizzando un programma di applicazione, è possibile rilasciare l'indice con l'istruzione SQL DROP. In questo caso, prendere nota dei parametri dell'indice, in modo da poter ricreare l'indice dopo il processo di geocoding in modalità batch.
4. Quando il geocoder legge un record di dati di origine, tenta di mettere in corrispondenza quel record con un corrispettivo nei dati di riferimento. Perché il geocoder elabori il record, la corrispondenza deve presentare in certo grado di precisione (definito *precisione*). Ad esempio, una precisione di 85 indica che la corrispondenza tra un record di origine e il suo corrispettivo nei dati di riferimento deve presentare un'accuratezza del 85%, perché il record di origine possa essere elaborato.

Occorre specificare il livello di precisione. Potrebbe inoltre essere necessario regolare la precisione. Ad esempio, nel caso in cui la precisione sia 100, se molti record di origine contengono indirizzi più recenti dei dati di riferimento, è impossibile che si verifichino corrispondenze di precisione del 100% tra i record in questione e i dati di riferimento. Di conseguenza, il geocoder rifiuterà questi record. In generale, se un geocoder genera dati spaziali il cui livello di precisione non sufficiente, è possibile risolvere il problema, modificando l'impostazione della precisione ed eseguendo nuovamente il geocoder.

Esecuzione del geocoder in modalità batch

Questa sezione fornisce una panoramica sui passaggi per eseguire un geocoder in modalità batch dal Centro di controllo. Alla panoramica segue una descrizione dettagliata delle modalità per completare ogni passaggio.

Per individuare le autorizzazioni necessarie per eseguire un geocoder in modalità batch, vedere "Autorizzazione" a pagina 100.

Panoramica sui passaggi per eseguire un geocoder in modalità batch:

1. Aprire la finestra Eseguire geocoder.
2. Indicare il geocoder da utilizzare.
3. Disattivare gli oggetti che potrebbero compromettere le prestazioni del geocoder.

4. Specificare quanti record devono essere sottoposti a geocoding prima che DB2 rilasci un commit.
5. Indicare la modalità di funzionamento del geocoder.
6. Indicare a DB2 Spatial Extender di eseguire il geocoder.

Passaggi dettagliati per l'esecuzione di un geocoder in modalità batch:

1. Aprire la finestra Eseguire geocoder.
 - a. Dalla finestra Centro di controllo, espandere la struttura ad albero degli oggetti fino a individuare la cartella **Tabelle** del database abilitato ai dati spaziali.
 - b. Selezionare la cartella **Tabelle**. Le tabelle vengono visualizzate nell'indice, sul lato destro della finestra.
 - c. Fare clic con il tastino destro del mouse sulla tabella desiderata nell'indice e selezionare **Strutture spaziali** nel menu a comparsa. Si apre la finestra Strutture spaziali.
 - d. Dalla finestra Strutture spaziali:
 - 1) Selezionare la struttura definita della colonna che si desidera riempire.
 - 2) Selezionare il pulsante **Eseguire geocoder**. Si apre la finestra Eseguire geocoder.
2. Se si desidera utilizzare il geocoder predefinito, non modificare la casella **Nome**, che riporta il nome del geocoder predefinito. Altrimenti, utilizzare la casella per selezionare il geocoder desiderato.
3. Disattivare gli oggetti che potrebbero compromettere le prestazioni del geocoder:
 - Se la colonna da riempire presenta un indice, selezionare la casella di spunta **Disabilitare temporaneamente gli indici spaziali durante il processo di geocoder**.
 - Se i trigger sono stati impostati per attivare il geocoding incrementale per questa colonna, selezionare la casella di spunta **Disabilitare temporaneamente i trigger spaziali durante il processo di geocoder**.
L'indice e i trigger verranno riattivati automaticamente, quando si seleziona **OK** nella finestra Eseguire geocoder.
4. Utilizzare il pulsante di selezione ciclica **Ambito commit** per specificare i record da sottoporre a geocoding prima che DB2 rilasci un commit. Ad esempio, se si desidera che DB2 esegua il commit di 100 record sottoposti a geocoding alla volta, indicare il numero 100.

Suggerimento: Se si desidera che DB2 rilasci un commit solo al termine dell'elaborazione di tutti i record, specificare zero.
5. Utilizzare i campi nella casella di gruppo **Parametri geocoder** per indicare la modalità di funzionamento del:

- Utilizzare il pulsante di selezione ciclica **Livello di precisione** per specificare, in termini di percentuale, il livello di precisione della corrispondenza tra i record di origine e i loro corrispettivi nei dati di riferimento. Per ulteriori informazioni sulla precisione, vedere “Informazioni sul processo di geocoding” a pagina 41.
- Se ci si server di un geocoder di un fornitore e si desidera utilizzare le proprietà da esso supportate, utilizzare la casella **Proprietà**, per impostare le proprietà.
- Se si desidera sottoporre a geocoding solo un sottoinsieme di righe della tabella selezionata, utilizzare la casella **Clausola WHERE**, per codificare una clausola SELECT WHERE, che specificherà i criteri relativi alle righe desiderate. Questa clausola può fare riferimento a qualsiasi colonna della tabella.

Digitare solo i criteri. Omettere la parola chiave WHERE. Ad esempio, se la tabella prevede una colonna STATE e si desidera sottoporre a geocoding solo le righe che contengono il valore MA in questa colonna, digitare:

```
STATE='MA'
```

6. Selezionare **OK** per eseguire il geocoder.

Importazione ed esportazione di dati

Questa sezione descrive i processi di importazione ed esportazione di dati e le modalità di utilizzo del Centro di controllo per:

- Importare dati da un file di scambio dati a una nuova tabella o a una tabella esistente
- Importare dati da un file di scambio dati a una tabella esistente
- Esportare dati da una tabella a un file di scambio dati

Informazioni su importazione ed esportazione

Questa sezione elenca i motivi per importare ed esportare dati spaziali. Descrive inoltre i file di scambio dati che fungono da interfaccia tra le origini dell'esportazione e le destinazioni di importazione.

E' possibile utilizzare DB2 Spatial Extender per importare ed esportare dati spaziali nei file di scambio dati. Tenere presente le seguenti situazioni:

- Il GIS contiene dati spaziali che rappresentano gli uffici, i clienti e altri elementi aziendali. Si desidera supportare questi dati con dati spaziali che rappresentino l'ambiente culturale dell'azienda—città, strade, località di interesse e così via. I dati desiderati sono reperibili da un fornitore di mappe geografiche. E' possibile utilizzare DB2 Spatial Extender per importare questi dati da un file di scambio dati del fornitore in questione.
- Si desidera migrare i dati spaziali da un sistema Oracle al GIS DB2 Spatial Extender. Si procede utilizzando un'utilità Oracle, per caricare i dati in un

file di scambio dati. Si utilizza quindi DB2 Spatial Extender, per importare i dati da questo file al database abilitato per le operazioni spaziali.

- Si desidera utilizzare un browser GIS per mostrare delle presentazioni di tipo visivo delle informazioni spaziali ai clienti. Il browser necessita solo dei file da cui ricavare i dati; non è necessario stabilire una connessione a un database. E' possibile utilizzare DB2 Spatial Extender, per esportare i dati in un file di scambio dati e servirsi quindi di un'utilità, per caricare i dati nel browser.

Il Centro di controllo supporta due tipi di file di scambio dati per i file shape DB2 Spatial Extender: e i file di trasferimento ESRI_SDE. I file shape vengono utilizzati spesso per importare dati che si originano in file system e per l'esportazione di dati nei file da caricare nei file system. I file di trasferimento ESRI_SDE vengono utilizzati spesso per importare dati che si originano nei database ESRI.

Importazione di dati in una nuova tabella o in una tabella esistente

Questa sezione fornisce una panoramica dei passaggi per importare i dati da un file shape o da un file di trasferimento ESRI_SDE in una tabella nuova o esistente. Alla panoramica segue una descrizione dettagliata delle modalità per completare ogni passaggio.

Per individuare il tipo di autorizzazione necessaria per importare i dati di forme, vedere "Autorizzazione" a pagina 89. Per individuare il tipo di autorizzazione necessaria per importare dati ESRI_SDE, vedere "Autorizzazione" a pagina 86.

Panoramica sui passaggi per importare dati in una tabella nuova o esistente:

1. Aprire la finestra Importare dati spaziali.
2. Specificare percorso, nome e formato del file che contiene i dati da importare.
3. Specificare i record da importare prima di ogni commit.
4. Se si desidera importare dati spaziali in una tabella da creare, assegnare un nome alla tabella e un nome alla colonna in cui devono essere inseriti i dati. Se si importano dati spaziali in una tabella esistente, indicare la colonna in cui devono essere inseriti i dati.
5. Specificare il sistema di riferimento spaziale da associare ai dati.
6. Designare un file per raccogliere i record che non è stato possibile importare.
7. Indicare a DB2 Spatial Extender di importare i dati e, se viene definita una tabella da questa finestra, di creare la tabella e registrare la colonna in cui devono essere inseriti i dati come struttura.

Passaggi dettagliati per importare i dati in una tabella nuova o esistente:

1. Aprire la finestra Importare dati spaziali.

- a. Dalla finestra Centro di controllo, espandere la struttura ad albero degli oggetti fino a individuare la cartella **Database**, nel server in cui si sta eseguendo DB2 Spatial Extender.
 - b. Selezionare la cartella **Database**. I database vengono visualizzati nel pannello indice posto a destra della finestra.
 - c. Fare clic con il tastino destro del mouse per importare i dati e selezionare **Spatial Extender** —> **Importare dati spaziali** nel menu a comparsa. Si apre la finestra Importare dati spaziali.
2. Specificare percorso, nome e formato del file che contiene i dati da importare:
 - a. Utilizzare il campo **Nome file** per specificare il percorso e il nome.
 - b. Utilizzare la casella **Formato file** per specificare il formato. Il formato può essere:

Shape Questo è il valore predefinito.

ESRI_SDE

Se si specifica questo formato, il campo **Nome riferimento spaziale** presenta come impostazione predefinita il nome del sistema di riferimento spaziale associati a questo formato.

3. Utilizzare il campo **Ambito commit**, per specificare il numero di record da importare prima di ogni commit. Ad esempio, per fare in modo che DB2 esegua il commit di 100 record alla volta, specificare il numero 100.

Suggerimento: Se si desidera che DB2 rilasci un commit solo al termine dell'elaborazione di tutti i record, specificare zero.

4. Specificare la tabella e la colonna in cui devono essere inseriti i dati.
 - a. Utilizzare la casella **Schema della struttura**, per specificare lo schema per la tabella in cui devono essere importati i dati.
 - b. Specificare la tabella e la colonna:
 - Se la tabella non esiste ancora:
 - 1) Nel campo **Tabella della struttura**, digitare un nome per la tabella.
 - 2) Nel campo **Colonna della struttura**, digitare un nome per la colonna che deve contenere i dati importati. DB2 Spatial Extender registrerà automaticamente questa colonna come struttura.
 - Se la tabella esiste già:
 - 1) Nel campo **Tabella della struttura**, specificare la tabella. La tabella deve già contenere la colonna in cui devono essere inseriti i dati importati. Inoltre, questa colonna deve essere già registrata come struttura.

- 2) Nel campo **Colonna della struttura**, specificare il nome della colonna in cui devono essere inseriti i dati importati.
5. Nel campo **Nome riferimento spaziale**, digitare o selezionare il sistema di riferimento spaziale da associare ai dati (se i dati devono provenire da un file di trasferimento ESRI_SDE, il nome del sistema di riferimento spaziale associato viene automaticamente visualizzato nel campo).
6. Nel campo **File di eccezione**, specificare il percorso e il nome per un nuovo file, in cui raccogliere i record che non è stato possibile importare. In seguito, è possibile correggere questi record e importarli dal file. DB2 Spatial Extender creerà questo file; non specificare un file già esistente.
7. Selezionare **OK** per importare i dati. Quindi, se è stato assegnato un nome per una tabella che non esiste ancora, verrà creata la tabella e la colonna in cui i dati verranno registrati come struttura. Inoltre, verrà creato il file di eccezione specificato.

Importazione dei dati in una tabella esistente

Questa sezione fornisce una panoramica dei passaggi per importare i dati da un file shape o da un file di trasferimento ESRI_SDE in una tabella nuova o esistente. Alla panoramica segue una descrizione dettagliata delle modalità per completare ogni passaggio.

Per individuare il tipo di autorizzazione necessaria per importare i dati di forme, vedere "Autorizzazione" a pagina 89. Per individuare il tipo di autorizzazione necessaria per importare dati ESRI_SDE, vedere "Autorizzazione" a pagina 86.

Panoramica sui passaggi per importare dati in una tabella nuova o esistente:

1. Aprire la finestra Importare dati spaziali.
2. Specificare percorso e il nome del file che contiene i dati da importare.
3. Specificare i record da importare prima di ogni commit.
4. Specificare la colonna che deve contenere i dati spaziali da importare.
5. Specificare quale sistema di riferimento spaziale deve essere associato a questi dati.
6. Designare un file per raccogliere i record che non è stato possibile importare.
7. Indicare a DB2 Spatial Extender di importare i dati e, se è stata specificata una colonna non ancora creata, di creare la colonna e di registrarvi i dati come una struttura.

Passaggi dettagliati per importare i dati in una tabella esistente:

1. Aprire la finestra Importare dati spaziali.

- a. Dalla finestra Centro di Controllo, espandere l'albero oggetti fino a trovare la cartella **Tabelle** per il database in cui si desidera importare i dati.
 - b. Selezionare la cartella **Tabelle**. Le tabelle vengono visualizzate nell'indice, sul lato destro della finestra.
 - c. Fare clic sulla tabella in cui si stanno importando i dati e selezionare **Spatial Extender** → **Importare dati spaziali** dal menu a comparsa. Si apre la finestra Importare dati spaziali.
2. Nella casella **Nome file**, specificare il percorso e il nome del file che contiene i dati da importare.
 3. Utilizzare la casella **Ambito commit**, per specificare il numero di record da importare prima di ogni. Ad esempio, per fare in modo che DB2 esegua il commit di 100 record alla volta, specificare il numero 100.

Suggerimento: Se si desidera che DB2 rilasci un commit solo al termine dell'elaborazione di tutti i record, specificare zero.

4. Specificare la colonna che deve contenere i dati spaziali da importare.
 - Se la colonna non esiste ancora, utilizzare la casella **Colonna della struttura**, per digitare un nome per la colonna.
 - Se la colonna esiste già, utilizzare la casella **Colonna della struttura**, per selezionare o digitare il nome della colonna.
5. Utilizzare la casella **Nome riferimento spaziale**, per specificare quale sistema di riferimento spaziale deve essere associato ai dati importati.
 - Se si sta aggiungendo una colonna a una tabella, digitare o selezionare il nome del sistema di riferimento spaziale.
 - Se i dati importati devono essere inseriti in una colonna esistente, lasciare la casella **Nome riferimento spaziale** immodificata. La casella visualizza il nome del sistema di riferimento spaziale predefinito.
6. Nel campo **File di eccezione**, specificare il percorso e il nome per un nuovo file, in cui raccogliere i record che non è stato possibile importare. In seguito, è possibile correggere questi record e importarli dal file. DB2 Spatial Extender creerà questo file; non specificare un file già esistente.
7. Selezionare **OK** per importare i dati. Quindi, se è stata specificata una colonna che non esiste ancora, verrà creata la colonna e registrata come una struttura. Inoltre, verrà creato il file di eccezione specificato.

Esportazione dei dati in un file shape

Questa sezione fornisce una panoramica sui passaggi per esportare i dati in un file shape. Alla panoramica segue una descrizione dettagliata delle modalità per completare ogni passaggio.

Per individuare il tipo di autorizzazione necessaria per eseguire i passaggi, vedere “Autorizzazione” a pagina 84.

Panoramica dei passaggi per esportare i dati da un file shape:

1. Aprire la finestra Esportare dati spaziali.
2. Specificare la colonna che contiene i dati spaziali da esportare.
3. Se si desidera esportare un sottoinsieme di righe di dati, identificare questo sottoinsieme in DB2 Spatial Extender.
4. Specificare il percorso e il nome del file in cui si sta esportando i dati.
5. Indicare a DB2 Spatial Extender di esportare i dati.

Panoramica dei passaggi per esportare i dati in un file shape:

1. Aprire la finestra Esportare dati spaziali.
 - a. Dalla finestra Centro di controllo, espandere la struttura ad albero degli oggetti fino a individuare la cartella **Tabelle** o **Viste** del database che contiene i dati spaziali:
 - b. Selezionare la cartella **Tabelle** o **Viste**. Vengono visualizzate le tabelle o le viste nel pannello di indice sul lato destro della finestra.
 - c. Fare clic sulla tabella o vista che contiene i dati da esportare e selezionare **Spatial Extender** —> **Esportare dati spaziali** dal menu a comparsa. Si apre la finestra Esportare dati spaziali.
2. Nel campo **Colonna della struttura**, specificare il nome della colonna che contiene i dati spaziali da esportare.
3. Se si desidera esportare un sottoinsieme di righe di tabella, utilizzare la casella **Clausola WHERE**, per digitare una clausola WHERE che specifichi i criteri per le righe desiderate. In questa clausola, è possibile fare riferimento solo alle colonne della tabella o della vista da cui si stanno esportando i dati.

Digitare solo i criteri. Omettere la parola chiave WHERE. Ad esempio, se la tabella o la vista presenta una colonna STATE e si desidera sottoporre a geocoding solo le righe che contengono il valore MA in questa colonna, digitare:

```
STATE='MA'
```
4. Nel campo **Nome file**, specificare il percorso e il nome del file in cui si stanno esportando i dati.
5. Selezionare **OK** per esportare i dati.

Capitolo 6. Creazione di indici spaziali

Questo capitolo descrive come utilizzare il Centro di controllo per creare un indice per i dati spaziali.

Dopo aver immesso i dati nella colonna spaziale, è possibile iniziare a creare un indice spaziale. Di solito le strutture indicizzate, quali la struttura ad albero B, eseguono ordinamenti lineari, a una dimensionale sui dati di tabella. I dati di tabella abilitati per le operazioni spaziali non vengono memorizzati come voci singole, sono infatti dati bidimensionali. Ad esempio le geometrie spaziali quali il poligono comprendono diversi valori di coordinate in una colonna spaziale o struttura. Poiché l'indice della struttura ad albero B non può gestire tipi di dati spaziali, DB2 Spatial Extender ha creato una tecnologia di indicizzazione proprietaria denominata *indice griglia*. L'indice griglia si basa sull'indice della struttura ad albero B, che è stato migliorato per la gestione di dati bidimensionali e l'esecuzione dell'indice sulle colonne spaziali. L'indice griglia supporta tre livelli ed è progettato per fornire prestazioni elevate per una vasta gamma di oggetti, dimensioni e distribuzioni dei dati. Per ulteriori informazioni sugli indici spaziali, vedere il "Capitolo 12. indici spaziali" a pagina 119.

Per verificare il tipo di autorizzazione richiesto per la creazione di un indice spaziale, consultare la sezione "Autorizzazione" a pagina 78.

Utilizzo del Centro di controllo per creare un indice spaziale

Per creare un indice spaziale utilizzando il Centro di controllo:

1. Nella struttura ad albero, selezionare la cartella **Tabelle**. Tutte le tabella esistenti verranno visualizzate nell'Indice.
2. Nell'Indice, fare clic con il tastino destro del mouse sulla tabella che si desidera creare, quindi fare clic su **Spatial Extender** —> **Indici spaziali** nel menu a comparsa. Verrà aperta la finestra Indici spaziali.
3. Dalla finestra Indici spaziali, fare clic su **Creare**. Viene aperta la finestra Creare indice spaziale.
4. Nel campo **Nome**, digitare il nome del nuovo indice spaziale che si desidera creare.

Nota: Non è necessario specificare uno schema. DB2 Spatial Extender aggiungerà automaticamente lo schema e creerà un nome completo.

5. Nel campo **Colonna struttura**, selezionare la struttura per la quale si sta creando un indice.

La struttura è una colonna spaziale definita o registrata su DB2 Spatial Extender.

6. Nei campi **Dimensione griglia**, digitare il valore di dimensione che si desidera assegnare a ciascun campo.

I livelli di griglia, **Definizione massima**, **Definizione media** e **Definizione minima**, vengono impostati aumentando la dimensione delle celle. Pertanto il secondo livello dovrà presentare una dimensione di cella superiore al primo livello e il terzo superiore al secondo.

Determinazione della dimensione delle celle di griglia

E' possibile determinare la corretta dimensione di griglia eseguendo un processo di prove ed errori. Si consiglia di impostare la dimensione di griglia in base alla dimensione approssimativa dell'oggetto che si desidera indicizzare. Se la dimensione è troppo elevata o troppo bassa, le prestazioni potrebbero risultare ridotte. Le dimensioni troppo ridotte possono influenzare il rapporto chiave/oggetto durante una ricerca di indice. Infatti viene creato un numero eccessivo di chiavi e viene restituito un numero elevato di candidati. Se invece la dimensione di griglia è impostata su un valore troppo elevato, l'indice di ricerca iniziale restituirà un numero inferiore di candidati, ma le prestazioni potrebbero essere rallentate durante la scansione finale della tabella.

Per ulteriori informazioni relative alla selezione delle dimensioni di cella di griglia e il numero di livelli di griglia, vedere la sezione "Selezione della dimensione delle celle di griglia" a pagina 126.

Capitolo 7. Richiamo e analisi delle informazioni spaziali

Dopo aver creato gli indici spaziali, le tabelle spaziali sono pronte per l'utilizzo. Questo capitolo fornisce informazioni sul richiamo e sull'analisi dei dati spaziali. Inoltre, presenta una panoramica delle diverse modalità di richiamo e fornisce degli esempi per l'interrogazione delle tabelle che utilizzano le funzioni spaziali.

Metodi di esecuzione dell'analisi spaziale

E' possibile eseguire l'analisi dei dati utilizzando le funzioni SQL e spaziali con uno dei seguenti ambienti di programmazione:

- Un geobrowser (ad esempio, ArcExplorer di ESRI).
Per ulteriori informazioni sull'utilizzo di ArcExplorer, consultare l'argomento *Using ArcExplorer*, disponibile sul sito Web ESRI all'indirizzo <http://www.esri.com>.
- Istruzioni SQL interattive.
- Applicazioni sviluppate dall'utente (ad esempio, ODBC, JDBC e SQL incorporato).

Applicazioni che possono essere avviate dal Centro comandi DB2, dalla finestra Comandi DB2 o dal processore di riga comandi.

Creazione di un'interrogazione spaziale

Questa sezione fornisce informazioni sulla creazione delle interrogazioni spaziali che utilizzano le funzioni spaziali e i predicati.

Funzioni spaziali e SQL

DB2 Spatial Extender include le funzioni che eseguono diverse operazioni sui dati spaziali. Gli esempi riportati in questa sezione indicano come utilizzare le funzioni spaziali per creare le proprie interrogazioni spaziali.

La Tabella 3 fornisce un elenco di funzioni spaziali e dei tipi di operazione che possono essere eseguiti.

Tabella 3. Operazioni e funzioni spaziali

Tipo di funzione	Esempio di operazione
Calcolo	Calcolare la distanza tra i punti
Confronto	Ricerca tutti i clienti ubicati in un'area a rischio
Scambio di dati	Convertire i dati nei formati supportati

Tabella 3. Operazioni e funzioni spaziali (Continua)

Tipo di funzione	Esempio di operazione
Trasformazione	Aggiungere un raggio di cinque miglia ad un punto

Per ulteriori informazioni sulle funzioni spaziali, consultare le sezioni “Capitolo 13. Geometrie e funzioni spaziali associate” a pagina 129 e “Capitolo 14. Funzioni spaziali per le interrogazioni SQL” a pagina 165.

Esempio 1: Confronto

La seguente interrogazione ricerca la distanza media di un cliente da un centro commerciale. Le funzioni spaziali utilizzate in questo esempio sono ST_Distance e ST_Within.

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

Esempio 2: Scambio di dati

La seguente interrogazione ricerca i clienti che vivono nell’area della Baia di San Francisco. Le funzioni spaziali utilizzate in questo esempio sono ST_AsText (data exchange) e ST_Within. ST_AsText converte i dati spaziali della colonna c.location nel formato OGC TEXT.

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

Esempio 3: Calcolo

La seguente interrogazione ricerca tutte le strade con lunghezza superiore a 10.5 miglia. La funzione spaziale utilizzata in questo esempio è ST_Length.

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

Esempio 4: Trasformazione

Questa interrogazione ricerca i clienti che vivono all’interno dell’area a rischio o entro 2 miglia dai confini di questa area. Le funzioni spaziali utilizzate in questo esempio sono ST_Buffer (transformation) e ST_Within.

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

Predicati spaziali e SQL

Un gruppo specializzato di funzioni spaziali denominate predicati spaziali può ottimizzare le prestazioni dell’interrogazione. L’impiego dei predicati spaziali, come ST_Overlaps che confronta due poligoni per verificare se si sovrappongono, può risultare complesso sia in termini di memoria che di tempo. Pertanto, è importante utilizzare tecniche di ottimizzazione in grado di ridurre i costi di esecuzione. L’ottimizzatore delle interrogazioni DB2 utilizza

l'indice spaziale per migliorare le prestazioni quando vengono impiegati dei predicati spaziali in base ai criteri descritti nell'ultima parte di questa sezione. Per ulteriori informazioni sui predicati spaziali, consultare la sezione "Funzioni predicato" a pagina 143. I predicati spaziali utilizzati per impostare l'indice spaziale sono:

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

Per un elenco completo di tutti i predicati e le funzioni spaziali, consultare la sezione "Capitolo 14. Funzioni spaziali per le interrogazioni SQL" a pagina 165.

Regole per l'impostazione dell'indice

Applicare le seguenti regole se si desidera ottimizzare le interrogazioni spaziali che utilizzano i predicati spaziali:

- Il predicato deve essere utilizzato nella clausola WHERE.
- Il predicato deve trovarsi nella parte sinistra del confronto. Ad esempio:
WHERE db2gse.ST_Within(c.location,:BayArea)=1
- I confronti di uguaglianza devono utilizzare la costante integer 1.
WHERE db2gse.ST_Within(c.location,:BayArea)=1
- Il predicato deve presentare una colonna spaziale utilizzata come destinazione di ricerca e un indice spaziale creato in questa colonna.

Esempi di impostazione dell'indice

La Tabella 4 indica i metodi di creazione delle interrogazioni spaziali corretti e quelli non validi per l'impostazione dell'indice spaziale.

Tabella 4. Regole per l'impostazione dell'indice

Interrogazione spaziale	Regola violata
SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=1	Nessuna condizione è stata violata in questo esempio.

Tabella 4. Regole per l'impostazione dell'indice (Continua)

Interrogazione spaziale	Regola violata
<pre>SELECT * FROM customers c WHERE db2gse.ST_Distance(c.location,:SanJose)<10</pre>	Nessuna condizione è stata violata in questo esempio.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Length(c.location)>10</pre>	Il predicato deve essere utilizzato nella clausola WHERE. (ST_Length è una funzione spaziale, ma <i>non</i> un predicato.)
<pre>SELECT * FROM customers c WHERE 1=db2gse.ST_Within(c.location,:BayArea)</pre>	Il predicato deve trovarsi nella parte sinistra del confronto.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=2</pre>	I confronti di uguaglianza devono utilizzare la costante integer 1.
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(:SanJose,:BayArea)=1</pre>	Il predicato deve presentare una colonna spaziale utilizzata come destinazione di ricerca e un indice spaziale creato in questa colonna. (SanJose e BayArea non sono colonne spaziali e quindi non possono essere associati a un indice spaziale.)

Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender

Questo capitolo descrive le modalità di utilizzo del programma di esempio DB2 Spatial Extender, per scrivere applicazioni e utilizzare e personalizzare informazioni spaziali. Sono inclusi i seguenti argomenti:

- Utilizzo del programma di esempio
- Passaggi del programma di esempio

Utilizzo del programma di esempio

Il programma di esempio DB2 Spatial Extender semplifica la programmazione di applicazioni. Con il programma di esempio, è possibile:

- Automatizzare le procedure spaziali di routine
- Tagliare e incollare il codice di esempio nelle applicazioni
- Comprendere i passaggi richiesti in genere per creare e conservare un database abilitato alle operazioni spaziali

Utilizzare il programma di esempio per codificare le attività più complesse per DB2 Spatial Extender, ad esempio per scrivere un'applicazione che utilizza l'interfaccia del database per richiamare le procedure memorizzate di DB2 Spatial Extender. Dal programma di esempio, è possibile copiare e personalizzare le applicazioni. Se non si ha dimestichezza con i passaggi di programmazione relativi a DB2 Spatial Extender, è possibile eseguire il programma di esempio, che riporta ogni passaggio in dettaglio. Prima di tutto è necessario creare il programma di esempio. A questo scopo basta utilizzare il makefile di esempio. Per istruzioni sulla creazione e sull'esecuzione del programma di esempio, vedere "Verifica dell'installazione" a pagina 20.

Passaggi del programma di esempio

Tabella 5 a pagina 58 riporta i passaggi del programma di esempio, le procedure memorizzate associate e una descrizione di ogni passaggio. Le funzioni C per richiamare le procedure memorizzate vengono visualizzate nella colonna delle azioni di Tabella 5 a pagina 58 e sono racchiuse in parentesi. Per ulteriori informazioni sulle procedure memorizzate, vedere "Capitolo 9. Procedure memorizzate" a pagina 67. Il programma di esempio fa riferimento a situazioni introdotte in "Scenario: Una società di assicurazione aggiorna il GIS" a pagina 12.

Tabella 5. programma di esempio DB2 Spatial Extender

Passaggi del programma di esempio	Azione	Descrizione
Abilitare/disabilitare il database spaziale	<ol style="list-style-type: none"> 1. Abilitare il database spaziale (gseEnableDB) 2. Disabilitare il database spaziale (gseDisableDB) 3. Abilitare il database spaziale (gseEnableDB) 	<ol style="list-style-type: none"> 1. Questo è il primo passaggio richiesto per utilizzare DB2 Spatial Extender. Un database abilitato per le operazioni spaziali include un insieme di tipi spaziali, un insieme di funzioni spaziali, un insieme di predicati spaziali, un nuovo tipo di indice e un insieme di tabelle e viste amministrative. 2. Questo passaggio viene eseguito in genere quando vengono attivate le funzioni spaziali per il database non corretto. Quando si disattiva un database spaziale, si rimuove un insieme di tipi spaziali, un insieme di funzioni spaziali, un insieme di predicati spaziali, un nuovo tipo di indice e un insieme di tabelle e viste amministrative. Nota: Non sarà possibile disattivare il database se sono stati creati oggetti che dipendono dagli oggetti creati dalla procedura di attivazione del database. Ad esempio, la creazione di una tabella con una colonna spaziale del tipo ST_Point renderà impossibile eseguire la procedura di disattivazione del database. Questo accade perché la tabella dipende dal tipo ST_Point, che deve essere rilasciato dalla procedura di disattivazione del database. 3. Come per 1.

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Registrazione i sistemi di riferimento spaziale	1. Registrare il sistema di riferimento spaziale per la colonna LOCATION della tabella CUSTOMERS (gseEnableSref)	1. Questo passaggio definisce un nuovo sistema di riferimento spaziale (SRS), da utilizzare per interpretare i dati spaziali della tabella CUSTOMERS. Un sistema di riferimento spaziale include i dati di geometri in una forma che può essere memorizzata in una colonna di un database abilitato alle operazioni spaziali. Dopo aver registrato SRS in una struttura specifica, le coordinate applicabili alla struttura possono essere memorizzate nella colonna associata della tabella CUSTOMERS.
	2. Registrare il sistema di riferimento spaziale per la colonna LOCATION della tabella OFFICES (gseEnableSref)	2. Questo passaggio definisce un nuovo sistema di riferimento spaziale (SRS), da utilizzare per interpretare i dati spaziali della struttura OFFICES. Ogni struttura di tabella deve prevedere la definizione di un SRS. Le strutture della tabella OFFICES potrebbero richiedere un SRS diverso da quello della struttura della tabella CUSTOMERS.
	3. Annullare la registrazione del sistema di riferimento spaziale per la colonna LOCATION della tabella OFFICES (gseDisableSref)	3. Questo passaggio viene eseguito se si specificano i parametri SRS non corretti per la struttura o la colonna spaziale. Quando si annulla la registrazione di un SRS per la struttura della tabella OFFICES, si rimuove la definizione e i parametri ad essa associati.
	4. Registrare nuovamente il sistema di riferimento spaziale per le colonne ZONE della tabella OFFICES (gseEnableSref)	4. Questo passaggio definisce un nuovo sistema di riferimento spaziale (SRS), da utilizzare per interpretare i dati spaziali della struttura OFFICES.

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Creare tabelle spaziali	<ol style="list-style-type: none"> 1. Modificare la tabella CUSTOMERS, aggiungendo la colonna LOCATION (gseSetupTables) 2. Creare la tabella OFFICES (gseSetupTables) 	<ol style="list-style-type: none"> 1. La tabella CUSTOMERS rappresenta i dati aziendali memorizzati nel database per diversi anni. L'istruzione ALTER TABLE aggiunge una nuova colonna (LOCATION) di tipo ST_Point. Questa colonna verrà riempita eseguendo il geocoding delle colonne indirizzo in un passaggio successivo. 2. La tabella OFFICES rappresenta, tra gli altri dati, le aree di vendita per ogni ufficio di una società assicurativa. L'intera tabella verrà riempita con i dati attribuito da un database non-DB2 in un passaggio successivo. Questo passaggio implica l'importazione dei dati attribuito nella tabella OFFICES da un file SHAPE.
Registrazione le strutture spaziali	<ol style="list-style-type: none"> 1. Registrare la colonna LOCATION nella tabella CUSTOMERS come una struttura (gseRegisterLayer) 2. Registrare la colonna ZONE della tabella OFFICES come una struttura (gseRegisterLayer) 	<p>Questi passaggi registrano le colonne LOCATION e ZONE come strutture per DB2 Spatial Extender. Prima di poter riempire o accedere a una colonna spaziale mediante le utilità DB2 Spatial Extender (ad esempio, geocoder), è necessario registrarla come una struttura.</p>

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Riempire le strutture spaziali	<ol style="list-style-type: none"> 1. Sottoporre a geocoding i dati indirizzo per la colonna LOCATION della tabella CUSTOMERS (gseRunGC) 2. Caricare la tabella OFFICES utilizzando la modalità di inserimento (gseImportShape) 3. Caricare la tabella HAZARD_ZONE, utilizzando la modalità di creazione (gseImportShape) 	<ol style="list-style-type: none"> 1. Questo passaggio esegue geocoding in modalità batch, richiamando l'utilità geocoder. Il geocoding in modalità batch viene eseguito in genere quando è necessario sottoporre a geocoding o risottoporre a geocoding. 2. Questo passaggio carica la tabella OFFICES con i dati spaziali esistenti nella forma di un file SHAPE. Dal momento che la tabella OFFICES esiste e la struttura OFFICES/ZONE è registrata, l'utilità di carico inserirà i nuovi record in una tabella esistente. 3. Questo passaggio carica la struttura HAZARD_ZONE con i dati spaziali esistenti nella forma di un file SHAPE. Dal momento che la tabella e la struttura non esistono, l'utilità di caricamento creerà la tabella e registrerà la struttura prima del caricamento dei dati.
Abilitare gli indici spaziali	<ol style="list-style-type: none"> 1. Abilitare l'indice spaziale per la colonna LOCATION della tabella CUSTOMERS (gseEnableIdx) 2. Abilitare l'indice spaziale per la colonna ZONE della tabella OFFICES (gseEnableIdx) 3. Abilitare l'indice spaziale per la colonna LOCATION della tabella OFFICES (gseEnableIdx) 4. Abilitare l'indice spaziale per la colonna BOUNDARY della tabella HAZARD_ZONE (gseEnableIdx) 	<p>Questi passaggi abilitano l'indice spaziale per le tabelle CUSTOMERS, OFFICES e HAZARD_ZONE.</p>

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Abilitare il geocoding automatico	1. Abilitare il geocoding automatico per le colonne LOCATION e ADDRESS della tabella CUSTOMERS (gseEnableAutoGC)	Questo passaggio attiva il richiamo automatico del geocoder. L'utilizzo del geocoding automatico fa sì che le colonne LOCATION e ADDRESS della tabella CUSTOMERS vengano sincronizzate l'una con l'altra, per le successive operazioni di inserimento e di aggiornamento.
Inserimento/aggiornamento della tabella CUSTOMERS	1. Inserire alcuni record con una strada diversa (gseInsDelUpd) 2. Aggiornare alcuni record con un nuovo indirizzo (gseInsDelUpd)	Questi passaggi mostrano un'operazione di inserimento e aggiornamento sulla colonna LOCATION della tabella CUSTOMERS. Una volta abilitato il geocoding automatico, le informazioni della colonna ADDRESS vengono automaticamente sottoposte a geocoding, quando vengono inserite o aggiornate nella colonna LOCATION. Questo processo è stato attivato nel passaggio precedente.
Disabilitare il geocoding automatico	1. Disabilitare il geocoding automatico per la struttura CUSTOMERS (gseDisableAutoGC) 2. Disabilitare l'indice spaziale per la struttura CUSTOMERS (gseDisableIdxCustomersLayer)	Questi passaggi disabilitano il richiamo automatico del geocoder e l'indice spaziale, per preparare il passaggio successivo (il passaggio successivo implica la riesecuzione del geocoding dell'intera tabella CUSTOMERS). Se si sta caricando una grande quantità di geodati, è consigliabile disattivare l'indice spaziale prima di caricare i dati e di riattivarlo al termine del caricamento.

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Rieseguire il geocoding della tabella CUSTOMERS	<ol style="list-style-type: none"> 1. Eseguire nuovamente il geocoding della struttura CUSTOMERS con un livello di precisione inferiore – 90% invece del 100% (gseRunGC) 2. Riattivare l'indice spaziale per la struttura CUSTOMERS (gseEnableIdx) 3. Riattivare il geocoding automatico con un livello di precisione – 90% invece del 100% (gseEnableAutoGC) 	<p>Questi passaggi eseguono il geocoder in modalità batch, riattivano il geocoding automatico con un nuovo livello di precisione e riabilitano l'indice spaziale e il geocoding automatico. E' consigliabile eseguire quest'operazione quando l'amministratore spaziale rileva un'alta percentuale di errore nel processo di geocoding. Se il livello di precisione è impostato su 100%, è possibile che non venga eseguito il geocoding di un indirizzo, perché non si riesce a trovare un indirizzo corrispondente nei dati di riferimento. Riducendo il livello di precisione, il geocoder ha maggiori probabilità di individuare dei dati corrispondenti. Dopo aver nuovamente sottoposto la tabella a geocoding in modalità batch, vengono riattivati il geocoding automatico e l'indice spaziale, per semplificare la gestione incrementale dell'indice spaziale e della colonna spaziale per le successive operazioni di inserimento e aggiornamento.</p>
Creare una vista e registrarne le colonne come strutture della vista	<ol style="list-style-type: none"> 1. Creare una vista, HIGHRISK_CUSTOMERS, in base al vincolo della tabella CUSTOMERS e HAZARD_ZONE (gseCreateView) 2. Registrare le colonne spaziali della vista come strutture della vista (gseRegisterLayer) 	<p>Questi passaggi creano una vista e ne registrano le colonne spaziali come strutture della vista.</p>

Tabella 5. programma di esempio DB2 Spatial Extender (Continua)

Passaggi del programma di esempio	Azione	Descrizione
Eseguire l'analisi spaziale	<ol style="list-style-type: none"> 1. Individuare la distanza media dei clienti da ogni ufficio (ST_Within, ST_Distance) 2. Individuare le entrate e i premi medi clienti per ogni ufficio (ST_Within) 3. Individuare i clienti che non sono iscritti agli uffici esistenti (ST_Within) 4. Individuare il numero di aree di rischio che ogni area di ufficio copre (ST_Overlaps) 5. Individuare l'ufficio più vicino alla posizione di un cliente specifico, dando per assunto che l'ufficio si trovi nel punto centrale dell'area degli uffici (ST_Distance, ST_Centroid) 6. Individuare i clienti, la cui posizione è vicina al limite di una particolare zona di rischio (ST_Buffer, ST_Overlaps) 7. Individuare quei clienti ad alto rischio iscritti a un particolare ufficio <p>(Tutti i passaggi si servono di gseRunSpatialQueries)</p>	<p>Questi passaggi eseguono l'analisi spaziale, utilizzando i predicati e le funzioni spaziali in linguaggio DB2 SQL. L'optimizer delle query DB2 ripartisce l'indice spaziale sulle colonne spaziali, per migliorare le prestazioni delle query, se possibile.</p>
Esportare le strutture spaziali nei file	Esportare la struttura highRiskCustomers (gseExportShape)	<p>Il passaggio riporta un esempio di esportazione dei risultati della query in un file SHAPE. L'esportazione dei risultati delle query in un altro formato file consente di utilizzare le informazioni mediante uno strumento di terze parti (ad esempio, ESRI ArcInfo).</p>

Parte 2. Materiale di riferimento

Capitolo 9. Procedure memorizzate

Questo capitolo illustra le procedure memorizzate che consentono la creazione di un sistema di informazioni geografiche con DB2 Spatial Extender. Quando si abilita e si utilizza DB2 Spatial Extender dal Centro di controllo, si richiamano in modo implicito queste procedure memorizzate. Ad esempio, quando si fa clic su **OK** da una finestra DB2 Spatial Extender, DB2 richiama le procedure memorizzate associate a tale finestra. In alternativa, è possibile richiamare in modo esplicito le procedure memorizzate in un programma applicativo. Si consiglia di includere il file di intestazione, `db2gse.h`, in tale programma. Questo file contiene le definizioni macro per le costanti assegnate ai parametri delle procedure memorizzate. Su AIX, il file è memorizzato nell'indirizzario `$DB2INSTANCE/sqlib/include/`. Su Windows NT, nell'indirizzario `%DB2PATH%\include\`.

Attenzione:

Tutte le costanti delle stringhe di carattere dei parametri di input delle procedure memorizzate sono sensibili al maiuscolo/minuscolo. Per individuare i parametri che richiedono queste costanti, consultare le tabelle riportate in questo capitolo.

Prima di richiamare una procedura memorizzata, in modo implicito o esplicito, è necessario collegarsi al database in cui è installato DB2 Spatial Extender. La prima procedura memorizzata che si utilizza è `db2gse.gse_enable_db`. Tale procedura abilita il database per le operazioni spaziali. E' possibile utilizzare le altre procedure memorizzate solo dopo l'abilitazione del database.

Le implementazioni delle procedure memorizzate vengono archiviate nella libreria `db2gse` del server DB2 Spatial Extender.

E' possibile utilizzare i seguenti elenchi per ricercare le procedure memorizzate per nome o per attività. Il primo elenco presenta i seguenti nomi:

- “`db2gse.gse_disable_autogc`” a pagina 70
- “`db2gse.gse_disable_db`” a pagina 72
- “`db2gse.gse_disable_sref`” a pagina 73
- “`db2gse.gse_enable_autogc`” a pagina 74
- “`db2gse.gse_enable_db`” a pagina 77

- “db2gse.gse_enable_idx” a pagina 78
- “db2gse.gse_enable_sref” a pagina 81
- “db2gse.gse_export_shape” a pagina 84
- “db2gse.gse_import_sde” a pagina 86
- “db2gse.gse_import_shape” a pagina 89
- “db2gse.gse_register_gc” a pagina 91
- “db2gse.gse_register_layer” a pagina 93
- “db2gse.gse_run_gc” a pagina 100
- “db2gse.gse_unregist_gc” a pagina 103
- “db2gse.gse_unregist_layer” a pagina 104

Il seguente elenco presenta le attività eseguite dalle procedure memorizzate.

- Creazione di un indice per una colonna spaziale (consultare “db2gse.gse_enable_idx” a pagina 78).
- Creazione di un sistema di riferimento spaziale (consultare “db2gse.gse_enable_sref” a pagina 81).
- Disabilitazione di un geocoder in modo che non sia possibile memorizzare automaticamente le colonne spaziali sincronizzate con le corrispondenti colonne attributo (consultare “db2gse.gse_disable_autogc” a pagina 70).
- Disabilitazione del supporto per le operazioni spaziali di un database (consultare “db2gse.gse_disable_db” a pagina 72).
- Eliminazione di un sistema di riferimento spaziale (consultare “db2gse.gse_disable_sref” a pagina 73).
- Abilitazione di un database per supportare operazioni spaziali (consultare “db2gse.gse_enable_db” a pagina 77).
- Abilitazione di un geocoder in modo che sia possibile memorizzare automaticamente le colonne spaziali sincronizzate con le corrispondenti colonne attributo (consultare “db2gse.gse_enable_autogc” a pagina 74).
- Esportazione di una struttura e della relativa tabella associata a un file shape (consultare “db2gse.gse_export_shape” a pagina 84).
- Importazione di una struttura e della relativa tabella associata da un file di trasferimento ESRI_SDE (consultare “db2gse.gse_import_sde” a pagina 86).
- Importazione di una struttura e della relativa tabella associata da un file shape (consultare “db2gse.gse_import_shape” a pagina 89).
- Registrazione di un geocoder diverso dal geocoder predefinito (consultare “db2gse.gse_register_gc” a pagina 91).
- Registrazione di una colonna spaziale come struttura (consultare “db2gse.gse_register_layer” a pagina 93).
- Esecuzione di un geocoder in modalità batch (consultare “db2gse.gse_unregist_gc” a pagina 103).

- Annullamento della registrazione di un geocoder diverso dal geocoder predefinito (consultare “db2gse.gse_unregist_layer” a pagina 104).
- Annullamento della registrazione di una struttura (consultare “db2gse.gse_unregist_layer” a pagina 104)

Per informazioni relative alla sequenza di esecuzione di tali attività, consultare “Capitolo 1. Informazioni relative a DB2 Spatial Extender” a pagina 3 e “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

db2gse.gse_disable_autogc

Utilizzare questa procedura memorizzata per cancellare o disabilitare in modo temporaneo i trigger che conservano una colonna spaziale sincronizzata con le relative colonne attributo associate. Ad esempio, si consiglia di disabilitare i trigger durante l'esecuzione del geocoder sui valori delle colonne attributo in modalità batch. Per ulteriori informazioni relative a tale argomento, consultare la sezione "Informazioni sul processo di geocoding" a pagina 41.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseDisableAutoGc nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di autorizzazioni e privilegi specifici o serie di privilegi, in particolare:

- L'autorizzazione SYSADM o DBADM sul database che contiene la tabella in cui sono definiti i trigger da cancellare o disabilitare temporaneamente.
- Il privilegio CONTROL sulla tabella.
- I privilegi ALTER, SELECT e UPDATE sulla tabella.

Parametri di input

Tabella 6. I parametri di input relativi alla procedura memorizzata *db2gse.gse_disable_autogc*.

Nome	Tipo di dati	Descrizione
operMode	SMALLINT	Indica se i trigger devono essere cancellati o disabilitati temporaneamente. Non è possibile specificare un valore nullo per questo parametro. Commento: per cancellare i trigger, utilizzare le macro GSE_AUTOGC_DROP. Per disabilitarli temporaneamente, utilizzare le macro GSE_AUTOGC_INVALIDATE. Per individuare i valori associati a queste macro, consultare il file db2gse.h. Su AIX, questo file è memorizzato nell'indirizzario \$DB2INSTANCE/sqlib/include/. Su Windows NT, nell'indirizzario %DB2PATH%\include\.

Tabella 6. I parametri di input relativi alla procedura memorizzata *db2gse.gse_disable_autogc*. (Continua)

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	<p>Il nome dello schema a cui appartiene la tabella o la vista specificata nel parametro layerTable.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: Se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata <i>db2gse.gse_disable_autogc</i>.</p>
layerTable	VARCHAR(128)	<p>Il nome della tabella che contiene la definizione dei trigger da cancellare o disabilitare temporaneamente.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
layerColumn	VARCHAR(128)	<p>Il nome della colonna abilitata come spaziale gestita dai trigger che si desidera cancellare o disabilitare temporaneamente.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>

Parametri di output

Tabella 7. I parametri di output relativi alla procedura memorizzata *db2gse.gse_disable_autogc*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_disable_db

Utilizzare questa procedura memorizzata per eliminare le risorse che consentono a DB2 Spatial Extender di memorizzare i dati spaziali e supportare le operazioni eseguite su questi dati.

Lo scopo di questa procedura memorizzata è quello di consentire all'utente la risoluzione dei problemi conseguenti l'abilitazione del database per le operazioni spaziali, *prima* di aggiungervi le colonne di tabelle spaziali o i dati. Ad esempio, nel caso in cui si decida, dopo aver abilitato un database per le operazioni spaziali, di utilizzare DB2 Spatial Extender per un altro database. Finché non si definiscono colonne spaziali o si importano dati spaziali, è possibile richiamare questa procedura memorizzata per eliminare tutte le risorse spaziali dal primo database.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C `gseDisableDB` nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre dell'autorizzazione SYSADM o DBADM sul database da cui le risorse DB2 Spatial Extender devono essere eliminate.

Parametri di output

Tabella 8. I parametri di output relativi alla procedura memorizzata db2gse.gse_disable_db.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_disable_sref

Utilizzare questa procedura memorizzata per cancellare un sistema di riferimento spaziale. Durante l'esecuzione di questa procedura memorizzata, le informazioni relative al sistema di riferimento spaziale vengono eliminate dalla vista catalogo DB2GSE.SPATIAL_REF_SYS. Per informazioni relative a questa vista, consultare la sezione "DB2GSE.SPATIAL_REF_SYS" a pagina 117.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseDisableSref nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

Nessuna richiesta.

Parametro di input

Tabella 9. I parametri di input relativi alla procedura memorizzata *db2gse.gse_disable_sref*.

Nome	Tipo di dati	Descrizione
srId	INTEGER	Identificativo numerico del sistema di riferimento spaziale che si desidera cancellare. Non è possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 10. I parametri di output relativi alla procedura memorizzata *db2gse.gse_disable_sref*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

Limitazione

Prima di cancellare un sistema di riferimento spaziale, è necessario annullare la registrazione delle strutture che lo utilizzano. Nel caso in cui la registrazione di tali strutture non viene annullata, la richiesta di cancellazione del sistema di riferimento spaziale verrà respinta.

db2gse.gse_enable_autogc

Utilizzare questa procedura memorizzata per effettuare le seguenti operazioni:

- Creare i trigger che memorizzano la colonna spaziale sincronizzata con le relative colonne attributo. Ogni volta che i valori vengono inseriti o aggiornati nelle colonne attributo, un trigger richiamerà un geocoder registrato per i valori aggiornati o inseriti e per posizionare i dati derivanti nella colonna spaziale.
- Riattivare i trigger dopo averli disabilitati temporaneamente.
- Determinare la funzione da utilizzare per eseguire il geocoder sui valori aggiornati e inseriti.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C `gseEnableAutoGC` nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di autorizzazioni e privilegi specifici o serie di privilegi, in particolare:

- L'autorizzazione SYSADM o DBADM sul database che contiene la tabella con la definizione dei trigger creati dalla procedura memorizzata.
- Il privilegio CONTROL sulla tabella.
- I privilegi ALTER, SELECT e UPDATE sulla tabella.

Parametri di input

Tabella 11. I parametri di input relativi alla procedura memorizzata `db2gse.gse_enable_autogc`.

Nome	Tipo di dati	Descrizione
<code>operMode</code>	SMALLINT	Il valore che indica se i trigger che inizializzano l'operazione del geocoder devono essere creati per la prima volta o riattivati dopo la disabilitazione temporanea. Non è possibile specificare un valore nullo per questo parametro. Commento: per creare i trigger, utilizzare le macro <code>GSE_AUTOGC_CREATE</code> . Per riattivarli utilizzare le macro <code>GSE_AUTOGC_RECREATE</code> . Per individuare i valori associati a queste macro, consultare il file <code>db2gse.h</code> . Su AIX, questo file è memorizzato nell'indirizzario <code>\$DB2INSTANCE/sqlib/include/</code> . Su Windows NT, nell'indirizzario <code>%DB2PATH%\include\</code> .

Tabella 11. I parametri di input relativi alla procedura memorizzata *db2gse.gse_enable_autogc*. (Continua)

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	<p>Il nome dello schema a cui appartiene la tabella specificata nel parametro layerTable.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata <i>db2gse.gse_enable_autogc</i>.</p>
layerTable	VARCHAR(128)	<p>Il nome della tabella utilizzato dai trigger creati o riattivati da questa procedura memorizzata.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
layerColumn	VARCHAR(128)	<p>Il nome della colonna spaziale che deve essere gestita dai trigger creati o riattivati da questa procedura memorizzata.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
gcId	INTEGER	<p>L'identificativo del geocoder che verrà richiamato dai trigger di inserimento e di aggiornamento creati o aggiornati da questa procedura memorizzata.</p> <p>Non è possibile specificare un valore nullo per questo parametro se il parametro operMode è impostato su <i>GSE_AUTOGC_CREATE</i>. E' possibile invece impostare un valore nullo se il parametro operMode è impostato su <i>GSE_AUTOGC_RECREATE</i>.</p>

Tabella 11. I parametri di input relativi alla procedura memorizzata *db2gse.gse_enable_autogc*. (Continua)

Nome	Tipo di dati	Descrizione
precisionLevel	INTEGER	<p>Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento in modo da consentire al geocoder di elaborare i dati di origine regolarmente.</p> <p>Non è possibile specificare un valore nullo per questo parametro se il parametro <i>operMode</i> è impostato su <i>GSE_AUTOGC_CREATE</i>. E' possibile invece impostare un valore nullo se il parametro <i>operMode</i> è impostato su <i>GSE_AUTOGC_RECREATE</i>.</p> <p>Commento: il livello di precisione è compreso nell'intervallo da 1 a 100%.</p>
vendorSpecific	VARCHAR(256)	<p>Le informazioni tecniche fornite dal fornitore; ad esempio, il percorso e il nome di un file utilizzati dal fornitore per impostare i parametri.</p> <p>Non è possibile specificare un valore nullo per questo parametro se il parametro <i>operMode</i> è impostato su <i>GSE_AUTOGC_CREATE</i>. E' possibile invece impostare un valore nullo se il parametro <i>operMode</i> è impostato su <i>GSE_AUTOGC_RECREATE</i>.</p>

Parametri di output

Tabella 12. I parametri di output relativi alla procedura memorizzata *db2gse.gse_enable_autogc*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

Limitazioni

- Il parametro *layerColumn* deve fare riferimento a una colonna registrata come struttura tabella.
- Se il parametro *operMode* è impostato su *GSE_AUTOGC_CREATE*, è necessario assegnare un identificativo di un geocoder registrato al parametro *gcId*.

db2gse.gse_enable_db

Utilizzare questa procedura memorizzata per fornire un database con le risorse necessarie per memorizzare dati spaziali e per supportare le operazioni. Queste risorse includono tipi di dati spaziali, un tipo di indice spaziale, tabelle e viste di catalogo, funzioni e procedure memorizzate.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseEnableDB nel programma di esempio. Per informazioni relative a questo programma, consultare il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

Autorizzazione

L’ID utente con il quale la procedura memorizzata viene richiamata deve disporre dell’autorizzazione SYSADM o DBADM sul database che si desidera abilitare.

Parametri di output

Tabella 13. I parametri di output relativi alla procedura memorizzata db2gse.gse_enable_db.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_enable_idx

Utilizzare questa procedura memorizzata per creare un indice per una colonna spaziale.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseEnableIdx nel programma di esempio. Per informazioni relative a questo programma, consultare il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

Autorizzazione

L’ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- L’autorizzazione SYSADM o DBADM sul database che contiene la tabella per la quale l’indice abilitato viene utilizzato.
- Il privilegio CONTROL o INDEX sulla tabella.

Parametri di input

Tabella 14. I parametri di input relativi alla procedura memorizzata db2gse.gse_enable_idx.

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	Il nome dello schema a cui appartiene la tabella specificata nel parametro layerTable. E’ possibile specificare un valore nullo per questo parametro. Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l’ID utente con cui è stata richiamata la procedura memorizzata db2gse.gse_enable_idx.
layerTable	VARCHAR(128)	Il nome della tabella in cui definire la definizione dell’indice che si desidera creare. Non è possibile specificare un valore nullo per questo parametro.
layerColumn	VARCHAR(128)	Il nome della colonna abilitata come spaziale da ricercare tramite l’indice che viene creato. Non è possibile specificare un valore nullo per questo parametro.

Tabella 14. I parametri di input relativi alla procedura memorizzata *db2gse.gse_enable_idx*. (Continua)

Nome	Tipo di dati	Descrizione
indexName	VARCHAR(128)	<p>Il nome dell'indice che si desidera creare.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p> <p>Commento: non specificare un nome di schema. DB2 Spatial Extender assegna automaticamente l'indice allo schema di riferimento del parametro <i>layerSchema</i>.</p>
gridSize1	DOUBLE	<p>Il numero che indica la granularità della griglia di indice migliore.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
gridSize2	DOUBLE	<p>Il numero che indica che (1) non è necessaria una seconda griglia per l'indice oppure (2) il tipo di granularità della seconda griglia.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: se non è necessaria una seconda griglia, specificare, 0. In caso contrario, è necessario che sia con granularità inferiore alla griglia specificata con <i>gridSize1</i>.</p>
gridSize3	DOUBLE	<p>Il numero che indica che (1) non è necessaria una terza griglia per l'indice oppure (2) il tipo di granularità della terza griglia.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: se non è necessaria una terza griglia, specificare, 0. In caso contrario, è necessario che sia con granularità inferiore alla griglia specificata con <i>gridSize2</i>.</p>

Parametri di output

Tabella 15. I parametri di output per la procedura memorizzata *db2gse.gse_enable_idx*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.

Tabella 15. I parametri di output per la procedura memorizzata db2gse.gse_enable_idx. (Continua)

Nome	Tipo di dati	Descrizione
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_enable_sref

Utilizzare questa procedura memorizzata per specificare quanti numeri negativi e decimali in un sistema di coordinate specifico devono essere convertite in valori interi positivi, in modo da consentirne la memorizzazione da parte di DB2 Spatial Extender. Le specifiche vengono complessivamente denominate *sistema di riferimento spaziale*. Durante l'esecuzione di questa procedura memorizzata le informazioni relative al sistema di riferimento spaziale vengono aggiunte alla vista catalogo DB2GSE.SPATIAL_REF_SYS. Per informazioni relative a questa vista, consultare la sezione "DB2GSE.SPATIAL_REF_SYS" a pagina 117.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseEnableSref nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

Nessuna richiesta.

Parametri di input

Tabella 16. I parametri di input relativi alla procedura memorizzata db2gse.gse_enable_sref.

Nome	Tipo di dati	Descrizione
srId	INTEGER	L'identificativo numerico relativo al sistema di riferimento spaziale. Non è possibile specificare un valore nullo per questo parametro. Commento: questo identificativo deve essere univoco nel database abilitato come spaziale.
srName	VARCHAR(64)	La descrizione breve del riferimento spaziale. Non è possibile specificare un valore nullo per questo parametro. Commento: la descrizione deve essere univoca nel database abilitato come spaziale.
falsex	DOUBLE	Un numero che, una volta sottratto da un valore di coordinata X negativo, diventa un numero non negativo (ossia un numero positivo o zero). Non è possibile specificare un valore nullo per questo parametro.

Tabella 16. I parametri di input relativi alla procedura memorizzata *db2gse.gse_enable_sref*. (Continua)

Nome	Tipo di dati	Descrizione
falsey	DOUBLE	<p>Un numero che, una volta sottratto da un valore di coordinata Y negativo, diventa un numero non negativo (ossia un numero positivo o zero).</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
xyunits	DOUBLE	<p>Un numero che, una volta moltiplicato per una coordinata X o Y decimale, produce un valore intero che è possibile memorizzare come dato a 32 bit.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
falsez	DOUBLE	<p>Un numero che, una volta sottratto da un valore di coordinata Z negativo, diventa un numero non negativo (ossia un numero positivo o zero).</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
zunits	DOUBLE	<p>Un numero che, una volta moltiplicato per una coordinata Z decimale, produce un valore intero che è possibile memorizzare come dato a 32 bit.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
falsem	DOUBLE	<p>Un numero che, una volta sottratto da una misura non negativa, diventa un numero non negativo (ossia un numero positivo o zero).</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
munits	DOUBLE	<p>Un numero che, una volta moltiplicato per una misura decimale, produce un valore intero che è possibile memorizzare come dato a 32 bit.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>

Tabella 16. I parametri di input relativi alla procedura memorizzata *db2gse.gse_enable_sref*. (Continua)

Nome	Tipo di dati	Descrizione
scId	INTEGER	L'identificativo numerico del sistema di coordinate da cui deriva il sistema di riferimento spaziale. Per rilevare il tipo dell'identificativo numerico del sistema di coordinate, consultare la vista catalogo DB2GSE.COORD_REF_SYS "DB2GSE.COORD_REF_SYS" a pagina 115. Non è possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 17. I parametri di output relativi alla procedura memorizzata *db2gse.gse_enable_sref*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_export_shape

Utilizzare questa procedura memorizzata per esportare una struttura e la relativa tabella associata in un file shape oppure per creare un file shape nuovo ed esportare la struttura e la relativa tabella associata in un file nuovo.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseExportShape nel programma di esempio. Per informazioni relative a questo programma, consultare il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

Autorizzazione

L’ID utente con il quale questa procedura memorizzata viene richiamata deve disporre del privilegio SELECT sulla tabella che si desidera esportare.

Parametri di input

Tabella 18. I parametri di input relativi alla procedura memorizzata db2gse.gse_export_shape.

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	Il nome dello schema a cui appartiene la tabella specificata nel parametro layerTable. E’ possibile specificare un valore nullo per questo parametro. Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l’ID utente con cui è stata richiamata la procedura memorizzata db2gse.gse_export_shape.
layerTable	VARCHAR(128)	Il nome della tabella che si desidera esportare. Non è possibile specificare un valore nullo per questo parametro.
layerColumn	VARCHAR(30)	Il nome della colonna registrata come struttura che si desidera esportare. Non è possibile specificare un valore nullo per questo parametro.
fileName	VARCHAR(128)	Il nome del file shape in cui la struttura specificata viene esportata. Non è possibile specificare un valore nullo per questo parametro.

Tabella 18. I parametri di input relativi alla procedura memorizzata *db2gse.gse_export_shape*. (Continua)

Nome	Tipo di dati	Descrizione
whereClause	VARCHAR(1024)	Il corpo della clausola SQL WHERE. Definisce una restrizione sulla serie di record per i quali eseguire il geocode. La clausola può fare riferimento a una colonna attributo della tabella che si sta esportando. E' possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 19. I parametri di output relativi alla procedura memorizzata *db2gse.gse_export_shape*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

Limitazione

E' possibile esportare solo una struttura alla volta.

db2gse.gse_import_sde

Utilizzare questa procedura memorizzata per importare un file di trasferimento SDE in un database che è stata abilitata per operazioni spaziali. La procedura memorizzata può funzionare nei seguenti due modi:

- Se il file di trasferimento SDE è di destinazione per una tabella esistente con una colonna struttura registrata, DB2 Spatial Extender caricherà la tabella con i dati del file.
- In caso contrario, DB2 Spatial Extender creerà una tabella con una colonna spaziale, registrerà questa colonna come struttura e caricherà la struttura e le altre colonne della tabella con i dati del file.

Il sistema di riferimento spaziale specificato nel file di trasferimento SDE verrà confrontato con i sistemi di riferimento spaziale registrati su DB2 Spatial Extender. Se il sistema specificato corrisponde a un sistema registrato, i valori decimali e negativi dei dati di trasferimento, verranno caricati, modificati come prescritto nel sistema registrato. Se il sistema specificato non corrisponde ad alcun sistema registrato, DB2 Spatial Extender creerà un nuovo sistema di riferimento spaziale per prescrivere le modifiche.

Autorizzazione

Quando si importano i dati in una tabella esistente, l'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- L'autorizzazione SYSADM o DBADM sul database che contiene la tabella in cui vengono importati i dati.
- Il privilegio CONTROL sulla tabella.

Quando è necessario creare la tabella in cui si desidera importare i dati, l'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- L'autorizzazione SYSADM o DBADM sul database che contiene la tabella che si desidera creare.

Parametri di input

Tabella 20. I parametri di input relativi alla procedura memorizzata *db2gse.gse_import_sde*.

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	<p>Il nome dello schema a cui appartiene la tabella o la vista specificata nel parametro layerTable.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata <i>db2gse.gse_import_sde</i>.</p>
layerTable	VARCHAR(128)	<p>Il nome della tabella in cui i dati di trasferimento SDE vengono caricati.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
layerColumn	VARCHAR(30)	<p>Il nome della colonna registrata come struttura in cui i dati spaziali del file di trasferimento SDE vengono caricati.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
fileName	VARCHAR(128)	<p>Il nome del file di trasferimento SDE che si desidera importare.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
commitScope	INTEGER	<p>Numero di record per punto di controllo.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>

Parametri di output

Tabella 21. I parametri di output relativi alla procedura memorizzata *db2gse.gse_import_sde*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	<p>Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.</p>

Tabella 21. I parametri di output relativi alla procedura memorizzata *db2gse.gse_import_sde*. (Continua)

Nome	Tipo di dati	Descrizione
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_import_shape

Utilizzare questa procedura memorizzata per importare un file shape in un database abilitato per le operazioni spaziali. La procedura memorizzata può funzionare nei seguenti due modi:

- Se il file shape è di destinazione per una tabella esistente con una colonna struttura registrata, DB2 Spatial Extender caricherà la tabella con i dati del file.
- In caso contrario, DB2 Spatial Extender creerà una tabella con una colonna spaziale, registrerà questa colonna come struttura e caricherà la struttura e le altre colonne della tabella con i dati del file.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C `gseImportShape` nel programma di esempio. Per informazioni relative a questo programma, consultare il “Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender” a pagina 57.

Autorizzazione

L’ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- L’autorizzazione SYSADM o DBADM sul database che contiene la tabella in cui i dati shape importati vengono caricati.
- Il privilegio CONTROL sulla tabella.

Parametri di input

Tabella 22. I parametri di input relativi alla procedura memorizzata `db2gse.gse_import_shape`.

Nome	Tipo di dati	Descrizione
<code>layerSchema</code>	<code>VARCHAR(30)</code>	<p>Il nome dello schema a cui appartiene la tabella o la vista specificata nel parametro <code>layerTable</code>.</p> <p>E’ possibile specificare un valore nullo per questo parametro.</p> <p>Commento: se non si specifica un valore per il parametro <code>layerSchema</code>, per impostazione predefinita viene utilizzato l’ID utente con cui è stata richiamata la procedura memorizzata <code>db2gse.gse_import_shape</code>.</p>
<code>layerTable</code>	<code>VARCHAR(128)</code>	<p>Il nome della tabella in cui il file shape importato viene caricato.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>

Tabella 22. I parametri di input relativi alla procedura memorizzata *db2gse.gse_import_shape*. (Continua)

Nome	Tipo di dati	Descrizione
layerColumn	VARCHAR(30)	Il nome della colonna registrata come struttura in cui i dati shape vengono caricati. Non è possibile specificare un valore nullo per questo parametro.
fileName	VARCHAR(128)	Il nome del file shape che si desidera importare. Non è possibile specificare un valore nullo per questo parametro.
exceptionFile	VARCHAR(128)	Il nome e il percorso del file in cui vengono memorizzati i dati shape che non possono essere importati. Questo è un file nuovo che verrà creato quando si esegue la procedura memorizzata <i>db2gse.gse_import_shape</i> . Non è possibile specificare un valore nullo per questo parametro.
srId	INTEGER	L'identificativo del sistema di riferimento spaziale utilizzato per la struttura in cui i dati shape vengono caricati. E' possibile specificare un valore nullo per questo parametro. Commento: se questo identificativo non è specificato, la trasformazione interna verrà impostata sulla risoluzione massima per il file shape.
commitScope	INTEGER	Numero di record per punto di controllo. E' possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 23. I parametri di output relativi alla procedura memorizzata *db2gse.gse_import_shape*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_register_gc

Utilizzare questa procedura memorizzata per registrare un geocoder diverso da quello predefinito. Per rilevare se un geocoder è stato già registrato, consultare la vista catalogo DB2GSE.SPATIAL_GEOCODER (descritta in “DB2GSE.SPATIAL_GEOCODER” a pagina 116).

Autorizzazione

L’ID utente con il quale la procedura memorizzata viene richiamata deve disporre dell’autorizzazione SYSADM o DBADM sul database che contiene il geocoder la cui registrazione viene effettuata da questa procedura memorizzata.

Parametri di input

Tabella 24. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_gc*.

Nome	Tipo di dati	Descrizione
gcId	INTEGER	L’identificativo numerico del geocoder che si desidera registrare. Non è possibile specificare un valore nullo per questo parametro. Commento: questo identificativo deve essere univoco all’interno del database.
gcName	VARCHAR(64)	La descrizione breve del geocoder che si desidera registrare. Non è possibile specificare un valore nullo per questo parametro. Commento: la descrizione deve essere una stringa di caratteri univoca nel database.
vendorName	VARCHAR(64)	Il nome del fornitore del geocoder che si desidera registrare. Non è possibile specificare un valore nullo per questo parametro.
primaryUDF	VARCHAR(256)	Il nome completo del geocoder che si desidera registrare. Non è possibile specificare un valore nullo per questo parametro.

Tabella 24. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_gc*. (Continua)

Nome	Tipo di dati	Descrizione
precisionLevel	INTEGER	<p>Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento in modo da consentire al geocoder di elaborare i dati di origine regolarmente.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p> <p>Commento: il livello di precisione è compreso nell'intervallo da 1 a 100%.</p>
vendorSpecific	VARCHAR(256)	<p>Le informazioni tecniche fornite dal fornitore; ad esempio, il percorso e il nome di un file utilizzati dal fornitore per impostare i parametri.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>
geoArea	VARCHAR(256)	<p>Area geografica per la quale eseguire il geocoder.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>
descrizione	VARCHAR(256)	<p>Commenti forniti dal fornitore.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>

Parametri di output

Tabella 25. I parametri di output relativi alla procedura memorizzata *db2gse.gse_register_gc*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_register_layer

Utilizzare questa procedura memorizzata per registrare una colonna spaziale come struttura. Durante l'esecuzione di questa procedura memorizzata, le informazioni relative alla struttura di cui si esegue la registrazione vengono aggiunte alla vista catalogo DB2GSE.GEOMETRY_COLUMNS. Per informazioni relative a questa vista, consultare la sezione "DB2GSE.GEOMETRY_COLUMNS" a pagina 116.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseRegisterLayer nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Per una struttura di tabella:
 - L'autorizzazione SYSADM o DBADM sul database che contiene la tabella cui appartiene questa struttura.
 - Il privilegio CONTROL o ALTER sulla tabella.
- Per una struttura della vista:
 - Il privilegio SELECT sulla tabella o tabelle di base che contengono (1) i dati indirizzo da sottoporre al processo geocode per la struttura e (2) i dati spaziali che derivano dal processo geocode.

Parametri di input

Tabella 26. I parametri di input relativi alla procedura memorizzata db2gse.gse_register_layer.

Nome	Tipo di dati	Descrizione
layerSchema	INTEGER(30)	Il nome dello schema a cui appartiene la tabella o la vista specificata nel parametro layerTable. E' possibile specificare un valore nullo per questo parametro. Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata db2gse.gse_register_layer.

Tabella 26. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
layerTable	VARCHAR(128)	<p>Il nome della tabella che contiene la colonna che si desidera registrare come struttura.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
layerColumn	VARCHAR(128)	<p>Il nome della tabella che si desidera registrare come struttura. Se la colonna non esiste, DB2 Spatial Extender ne creerà una nuova.</p> <p>Non è possibile specificare un valore nullo per questo parametro.</p>
layerTypeName	VARCHAR(64)	<p>Il tipo di dati della colonna che si desidera registrare come struttura. E' necessario specificare il tipo di dati in maiuscolo, ad esempio: ST_POINT</p> <p>Non è possibile specificare un valore nullo per questo parametro se la colonna è una colonna tabella che viene creata durante l'elaborazione della procedura memorizzata. In caso contrario, se la colonna è una colonna esistente all'interno di una tabella o vista, è possibile specificare un valore nullo per questo parametro.</p>
srId	INTEGER	<p>L'identificativo del sistema di riferimento spaziale utilizzato per questa struttura.</p> <p>Non è possibile specificare un valore nullo per una struttura di tabella. DB2 Spatial Extender ignora questo parametri quando si registra una struttura di vista.</p>
geoSchema	VARCHAR(30)	<p>Viene utilizzato quando si registra una colonna vista come struttura. Il parametro geoSchema è lo schema della tabella alla base della vista cui appartiene la colonna.</p> <p>Quando si registra una colonna vista come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna tabella come struttura.</p> <p>Commento: se non si specifica un valore per il parametro geoSchema, per impostazione predefinita viene utilizzato il valore del parametro layerSchema.</p>

Tabella 26. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
geoTable	VARCHAR(128)	<p>Viene utilizzato quando si registra una colonna vista come struttura. Il parametro geoTable è il nome della tabella alla base della vista cui appartiene la colonna.</p> <p>Quando si registra una colonna vista come struttura non è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna tabella come struttura.</p>
geoColumn	VARCHAR(128)	<p>Viene utilizzato quando si registra una colonna vista come struttura. Il parametro geoColumn è il nome della colonna tabella alla base di questa colonna vista.</p> <p>Quando si registra una colonna vista come struttura non è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna tabella come struttura.</p>
nAttributes	SMALLINT	<p>Il numero delle colonne che contengono i dati di origine di questa struttura per i quali eseguire il geocode.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p>
attr1Name	VARCHAR(128)	<p>Il nome della prima colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Se si intende utilizzare il geocoder predefinito, è necessario memorizzare gli indirizzi nella colonna attr1Name.</p>

Tabella 26. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
attr2Name	VARCHAR(128)	<p>Il nome della seconda colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Se si intende utilizzare il geocoder predefinito, è necessario memorizzare i nomi delle città nella colonna attr2Name.</p>
attr3Name	VARCHAR(128)	<p>Il nome della terza colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Se si intende utilizzare il geocoder predefinito, è necessario memorizzare i nomi o le abbreviazioni degli stati nella colonna attr3Name.</p>
attr4Name	VARCHAR(128)	<p>Il nome della quarta colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Se si intende utilizzare il geocoder predefinito, è necessario memorizzare i CAP nella colonna attr4Name.</p>

Tabella 26. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
attr5Name	VARCHAR(128)	<p>Il nome della quinta colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora l'attributo Attr5Name.</p>
attr6Name	VARCHAR(128)	<p>Il nome della sesta colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora la colonna Attr6Name.</p>
attr7Name	VARCHAR(128)	<p>Il nome della settima colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora la colonna Attr7Name.</p>

Tabella 26. I parametri di input relativi alla procedura memorizzata *db2gse.gse_register_layer*. (Continua)

Nome	Tipo di dati	Descrizione
attr8Name	VARCHAR(128)	<p>Il nome dell'ottava colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora la colonna Attr8Name.</p>
attr9Name	VARCHAR(128)	<p>Il nome della nona colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora la colonna Attr9Name.</p>
attr10Name	VARCHAR(128)	<p>Il nome della decima colonna che contiene i dati di origine da sottoporre al processo geocode per questa struttura.</p> <p>Quando si registra una colonna tabella come struttura è possibile specificare un valore nullo per questo parametro. DB2 Spatial Extender ignora questo parametro nel momento in cui registra una colonna vista come struttura.</p> <p>Il geocoder predefinito ignora la colonna Attr10Name.</p>

Parametri di output

Tabella 27. I parametri di output per la procedura memorizzata *db2gse.gse_register_layer*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

Limitazioni

- Se si esegue la registrazione di una colonna vista come struttura, è necessario che si basi su una colonna tabella già registrata come struttura.
- Non più di dieci colonne attributo possono contenere i dati da sottoporre al processo geocode per la struttura che si sta registrando.

db2gse.gse_run_gc

Utilizzare questa procedura memorizzata per l'esecuzione di un geocoder in modalità batch. Per informazioni relative a questa attività, consultare la sezione "Esecuzione del geocoder in modalità batch" a pagina 43.

Per un esempio del codice di richiamo di questa procedura memorizzata, consultare la funzione C gseRunGC nel programma di esempio. Per informazioni relative a questo programma, consultare il "Capitolo 8. Scrittura di applicazioni per DB2 Spatial Extender" a pagina 57.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- L'autorizzazione SYSADM o DBADM sul database che contiene la tabella in cui eseguire il geocoder specificato.
- Il privilegio CONTROL o UPDATE su questa tabella.

Parametri di input

Tabella 28. I parametri di input relativi alla procedura memorizzata db2gse.gse_run_gc.

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	Il nome dello schema a cui appartiene la tabella o la vista specificata nel parametro layerTable. E' possibile specificare un valore nullo per questo parametro. Commento: se non si specifica una valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata db2gse.gse_run_gc.
layerTable	VARCHAR(128)	Il nome della tabella che contiene la colonna che si desidera inserire i dati sottoposti al processo geocode. Non è possibile specificare un valore nullo per questo parametro.
layerColumn	VARCHAR(128)	Il nome della colonna in cui si desidera inserire i dati sottoposti al processo geocode. Non è possibile specificare un valore nullo per questo parametro.

Tabella 28. I parametri di input relativi alla procedura memorizzata db2gse.gse_run_gc. (Continua)

Nome	Tipo di dati	Descrizione
gcId	INTEGER	<p>L'identificativo del geocoder che si desidera eseguire.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Per rilevare gli identificativi dei geocoder registrati, consultare la vista catalogo DB2GSE.SPATIAL_GEOCODER.</p>
precisionLevel	INTEGER	<p>Il livello in base al quale i dati di origine devono corrispondere ai relativi dati di riferimento in modo da consentire al geocoder di elaborare i dati di origine regolarmente.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p> <p>Commento: il livello di precisione è compreso nell'intervallo da 1 a 100%.</p>
vendorSpecific	VARCHAR(256)	<p>Le informazioni tecniche fornite dal fornitore; ad esempio, il percorso e il nome di un file utilizzati dal fornitore per impostare i parametri.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>
whereClause	VARCHAR(256)	<p>Il corpo della clausola WHERE. Definisce una restrizione sulla serie di record per i quali eseguire il geocode. La clausola può fare riferimento a una colonna attributo della tabella in cui viene eseguito il geocoder.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>
commitScope	INTEGER	<p>Numero di record per punto di controllo.</p> <p>E' possibile specificare un valore nullo per questo parametro.</p>

Parametri di output

Tabella 29. I parametri di output relativi alla procedura memorizzata db2gse.gse_run_gc.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_unregist_gc

Utilizzare questa procedura memorizzata per annullare la registrazione di un geocoder diverso da quello predefinito.

Per rilevare le informazioni relative al geocoder di cui si desidera annullare la registrazione, consultare la vista catalogo DB2GSE.SPATIAL_GEOCODER; consultare la sezione "DB2GSE.SPATIAL_GEOCODER" a pagina 116.

Autorizzazione

L'ID utente con il quale la procedura memorizzata viene richiamata deve disporre dell'autorizzazione SYSADM o DBADM sul database che contiene il geocoder che si desidera annullare registrare.

Parametro di input

Tabella 30. I parametri di input relativi alla procedura memorizzata *db2gse.gse_unregist_gc*.

Nome	Tipo di dati	Descrizione
gcId	INTEGER	L'identificativo del geocoder di cui si desidera annullare la registrazione. Non è possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 31. I parametri di output relativi alla procedura memorizzata *db2gse.gse_unregist_gc*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

db2gse.gse_unregist_layer

Utilizzare questa procedura memorizzata per annullare la registrazione di una struttura. L'annullamento della registrazione viene eseguito dalla procedura memorizzata mediante le seguenti operazioni:

- Rimozione della definizione della struttura dalle tabelle catalogo di DB2 Spatial Extender.
- Cancellazione della restrizione di controllo stabilita da DB2 Spatial Extender su questa tabella di base della struttura per garantire che i dati spaziali della struttura siano conformi ai requisiti del sistema di riferimento spaziale della struttura.
- Cancellazione dei trigger utilizzati per aggiornare la colonna spaziale ogni volta che i dati indirizzo vengono aggiunti, modificati o rimossi.

Durante l'esecuzione di questa procedura memorizzata, le informazioni relative alla struttura vengono eliminate dalla vista meta DB2GSE.GEOMETRY_COLUMNS. Per informazioni relative a questa vista, consultare la sezione "DB2GSE.GEOMETRY_COLUMNS" a pagina 116.

Autorizzazione

L'ID utente con il quale questa procedura memorizzata viene richiamata deve disporre di una delle seguenti autorizzazioni o privilegi:

- Per una struttura di tabella:
 - L'autorizzazione SYSADM o DBADM sul database che contiene questa tabella di base della struttura.
 - Il privilegio CONTROL o ALTER sulla tabella.
- Per una struttura della vista:
 - Il privilegio SELECT sulla tabella o tabelle di base che contengono (1) i dati indirizzo da sottoporre al processo geocode per questa struttura e (2) i dati spaziali che derivano dal processo geocode.

Parametri di input

Tabella 32. I parametri di input relativi alla procedura memorizzata *db2gse.gse_unregister_layer*.

Nome	Tipo di dati	Descrizione
layerSchema	VARCHAR(30)	Il nome dello schema a cui appartiene la tabella specificata nel parametro layerTable. E' possibile specificare un valore nullo per questo parametro. Commento: se non si specifica un valore per il parametro layerSchema, per impostazione predefinita viene utilizzato l'ID utente con cui è stata richiamata la procedura memorizzata <i>db2gse.gse_unregister_layer</i> .
layerTable	VARCHAR(128)	Il nome della tabella che contiene la colonna specificata nel parametro layerColumn. Non è possibile specificare un valore nullo per questo parametro.
layerColumn	VARCHAR(128)	Il nome della colonna spaziale definita come struttura di cui si desidera annullare la registrazione. Non è possibile specificare un valore nullo per questo parametro.

Parametri di output

Tabella 33. I parametri di output relativi alla procedura memorizzata *db2gse.gse_unregister_layer*.

Nome	Tipo di dati	Descrizione
msgCode	INTEGER	Il codice associato ai messaggi restituiti dal programma che richiama questa procedura memorizzata.
Reserved	VARCHAR(1024)	Il messaggio di errore completo, creato sul server DB2 Spatial Extender.

Limitazione

Se una colonna vista definita come struttura vista si basa su una colonna tabella definita come struttura tabella, non è possibile annullare la registrazione di questa struttura tabella a meno che non si annulli la registrazione della struttura vista.

Capitolo 10. Messaggi

Questo capitolo descrive i messaggi che DB2 Spatial Extender restituisce agli utenti. A ciascun messaggio è associato un identificativo. Gli identificativi che terminano con la lettera E, contrassegnano i messaggi di errore, quelli che terminano con W, contrassegnano i messaggi di avvertenza e quelli che terminano con I, contrassegnano i messaggi informativi.

DBA7200E Sono state selezionate più di 10 colonne come input per un geocoder

Spiegazione: E' possibile selezionare un massimo di 10 colonne come input per geocoder.

Risposta utente: Spostare i nomi delle colonne dalla casella Colonne selezionate alla casella Colonne disponibili, finché nella casella Colonne selezionate non verranno elencati massimo dieci nomi.

DBA7201E Il database non è abilitato per le operazioni Spatial Extender.

Spiegazione: E' necessario abilitare il database per Spatial Extender prima di poterlo utilizzare.

Risposta utente: Fare clic con il tastino destro del mouse sul database e selezionare dai menu Spatial Extender —> Abilitare.

GSE0000I L'operazione è stata eseguita correttamente.

GSE0001E Spatial Extender non ha potuto eseguire l'operazione richiesta ("**<nome-operazione>**") con l'ID utente "**<id-utente>**".

Spiegazione: L'operazione è stata richiesta utilizzando un ID utente che non dispone di privilegi o autorizzazione per l'esecuzione.

Risposta utente: Consultare la documentazione per verificare quali sono le autorizzazioni

corrette o per richiederle al responsabile Spatial Extender.

GSE0002E "**<valore>**" non è un valore valido per l'argomento "**<nome-argomento>**".

Spiegazione: Il valore immesso non è corretto o la sintassi è errata.

Risposta utente: Consultare la documentazione o rivolgersi a un responsabile Spatial Extender per verificare il valore o l'intervallo di valori corretto.

GSE0003E Spatial Extender non ha potuto eseguire l'operazione richiesta poiché l'argomento "**<nome-argomento>**" non è stato specificato.

Spiegazione: Non è stato specificato un argomento richiesto per questa operazione.

Risposta utente: Specificare l'argomento "**<nome-argomento>**" con il valore prescelto; quindi richiedere nuovamente l'operazione.

GSE0004W L'argomento "**<nome-argomento>**" non è stato valutato.

Spiegazione: L'operazione richiesta non utilizza l'argomento "**<nome-argomento>**".

Risposta utente: Nessuna richiesta.

GSE0005E Spatial Extender non ha potuto eseguire la richiesta di creazione dell'oggetto "<nome-oggetto>".

Spiegazione: L'oggetto "<nome-oggetto>" esiste già oppure non si dispone delle autorizzazioni appropriate per la creazione. Può essere una tabella, una colonna, un trigger, un indice, un file o un altro tipo di oggetto.

Risposta utente: Se "<nome-oggetto>" è l'oggetto prescelto, non eseguire alcuna operazione. In caso contrario specificare il nome corretto e verificare che si disponga delle autorizzazioni appropriate per la creazione di oggetti.

GSE0006E Spatial Extender non ha potuto eseguire l'operazione richiesta sul nome oggetto abilitato o registrato "<nome oggetto>".

Spiegazione: L'oggetto "<nome-oggetto>" è già abilitato o registrato oppure esiste già. Può essere una struttura, un indice, un sistema di riferimento spaziale, un sistema di coordinate, un geocoder o un altro tipo di oggetto.

Risposta utente: Verificare che l'oggetto "<nome-oggetto>" esista e immettere nuovamente la richiesta.

GSE0007E Spatial Extender non ha potuto eseguire l'operazione richiesta su "<nome-oggetto>", poiché non è stato ancora abilitato o registrato.

Spiegazione: L'oggetto "<nome-oggetto>" non è stato abilitato o registrato. Può essere una struttura, un indice, un sistema di riferimento spaziale, un sistema di coordinate spaziale, un geocoder o un altro tipo di oggetto.

Risposta utente: Abilitare o registrare l'oggetto "<nome oggetto>". Inoltrare quindi nuovamente la richiesta.

GSE0008E Si è verificato un errore SQL imprevisto ("<messaggio-errore-sql>").

Risposta utente: Ricercare il messaggio descrittivo associato a SQLCODE nel messaggio di errore SQL "<messaggio-errore-sql>". Se necessario, rivolgersi al rappresentante IBM di zona.

GSE0009E Non è stato possibile eseguire l'operazione richiesta sull'oggetto denominato "<nome-oggetto>" già esistente.

Spiegazione: "<nome-oggetto>" esiste già nel database o nel sistema operativo. Può essere un file, una tabella, una colonna, un trigger, un indice o un altro tipo di oggetto.

Risposta utente: E' necessario specificare correttamente l'oggetto quando si tenta di accedervi. Se necessario, eliminare l'oggetto.

GSE0010E Non è stato possibile eseguire l'operazione richiesta sull'oggetto denominato "<nome-oggetto>"; è probabile che questo oggetto non esista già.

Spiegazione: "<nome-oggetto>" non esiste nel database o nel sistema operativo. Può essere un file, una tabella, una colonna, un trigger, un indice o un altro tipo di oggetto.

Risposta utente: Verificare che le autorizzazioni di cui si dispone sono sufficienti per l'accesso all'oggetto. Se si dispone delle corrette autorizzazioni e l'oggetto non esiste, è necessario crearlo.

GSE0011E Spatial Extender non ha potuto disattivare o annullare la registrazione dell'oggetto "<nome-oggetto>".

Spiegazione: "<nome-oggetto>" è dipendente da un altro oggetto. "<nome-oggetto>" può essere un sistema di riferimento spaziale, una struttura, un geocoder o un altro tipo di oggetto.

Risposta utente: Consultare la documentazione per verificare i tipi di oggetti di cui “<nome-oggetto>” può essere dipendente. Quindi rimuovere l’oggetto specifico da cui “<nome-oggetto>” è dipendente.

GSE0012E Spatial Extender non ha potuto eseguire la richiesta poiché la colonna spaziale completa “<schema-struttura.nome-struttura.colonna-struttura>” non è registrata come struttura di tabella.

Spiegazione: E’ necessario registrare il nome completo della colonna “<schema-struttura.nome-struttura.colonna-struttura>” come struttura tabella prima di eseguire alcune operazioni ad essa associate (ad esempio l’abilitazione dell’indice, l’abilitazione di un geocoder per la compilazione in modo batch o l’aggiornamento automatico).

Risposta utente: Verificare che la colonna spaziale completa “<schema-struttura.nome-struttura.colonna-struttura>” sia registrata come struttura tabella all’interno della vista DB2GSE.GEOMETRY_COLUMNS nel catalogo di Spatial Extender. Verificare inoltre che la tabella contenente questa colonna includa colonne di attributo corrispondenti valide.

GSE0013E Il database non è abilitato per le operazioni spaziali.

Spiegazione: Il database non è abilitato per le operazioni spaziali. Pertanto il catalogo Spatial Extender non esiste.

Risposta utente: Abilitare il database per le operazioni spaziali.

GSE0014E Il database è già stato abilitato per le operazioni spaziali.

Spiegazione: Il database è già stato abilitato per le operazioni spaziali.

Risposta utente: Verificare che il database sia stato abilitato nel modo desiderato. Se necessario, disabilitare il database.

GSE0498E Si è verificato il seguente errore: “<messaggio-errore>”.

GSE0499W Spatial Extender ha emesso il seguente messaggio di avvertenza: “<messaggio-avvertenza>”.

GSE0500E Il modo operazione specificato (“<modo-operazione>”) non è valido.

Spiegazione: Il modo specificato non è supportato dall’operazione richiesta.

Risposta utente: Consultare la documentazione per verificare i modi supportati dall’operazione.

GSE1001E Spatial Extender non è stato in grado di registrare un struttura vista denominato “<nome-schema.nome-vista.nome-colonna>” basato sulla colonna spaziale “<nome-schema.nome-tabella.nome-colonna>”.

Spiegazione: La colonna spaziale specificata (“<nome-schema.nome-tabella.nome-colonna>”) non è stata registrata come struttura tabella.

Risposta utente: Registrare la colonna “<nome-schema.nome-tabella.nome-colonna>” as come struttura tabella.

GSE1002E Spatial Extender non è stato in grado di registrare un struttura vista denominato “<nome-schema.nome-vista.nome-colonna>” basato sulla tabella “<nome-schema.nome-tabella>”.

Spiegazione: La tabella specificata (“<nome-schema.nome-tabella>”) non è compresa direttamente o indirettamente nella vista “<nome-schema.nome-vista.nome-colonna>”.

Risposta utente: Verificare e specificare il tipo di tabella di base per la vista “<nome-schema.nome-vista.nome-colonna>”.

GSE1003E Spatial Extender non ha potuto accedere a una colonna denominata "<nome-colonna>" in una tabella o vista denominata "<nome-schema.nome-oggetto>".

Spiegazione: La tabella o vista "<nome-schema.nome-oggetto>" non contiene una colonna "<nome-colonna>".

Risposta utente: Verificare la definizione della tabella o vista "<nome-schema.nome-oggetto>" per controllare il nome corretto della colonna prescelta.

GSE1004E Spatial Extender non è stato in grado di registrare la colonna spaziale completa "<nome-schema.nome-tabella.nome-colonna>" come struttura tabella.

Spiegazione: La colonna "<nome-schema.nome-tabella.nome-colonna>" non contiene tipi di dati spaziali oppure non è associata a una tabella di base.

Risposta utente: Definire un tipo di dati spaziali per la colonna "<nome-schema.nome-tabella.nome-colonna>" oppure verificare che questa colonna sia compresa nella tabella di base locale.

GSE1005E Il sistema di riferimento spaziale ("<id-riferimento-spaziale-struttura-vista>") specificato per una struttura vista è differente dal sistema di riferimento spaziale ("<id-riferimento-spaziale-struttura-tabella>") utilizzato per la struttura di tabella sottostante questa struttura.

Spiegazione: Un sistema di riferimento spaziale di una struttura vista sottostante il sistema di riferimento spaziale della struttura tabella.

Risposta utente: Specificare il sistema di riferimento spaziale della struttura tabella sottostante per la struttura vista.

GSE1006E Poiché "<id-riferimento-spaziale>" è un ID di sistema di riferimento spaziale non valido, Spatial Extender non ha eseguito la registrazione della struttura richiesto.

Spiegazione: Il sistema di riferimento spaziale specificato ("<id-riferimento-spaziale>") non è stato abilitato o registrato.

Risposta utente: Abilitare o registrare il sistema di riferimento spaziale. Quindi inoltrare nuovamente la richiesta di registrazione della struttura.

GSE1007E E' possibile che si sia verificato un errore SQL (SQLSTATE "<sqlstate>") durante il tentativo non riuscito di aggiungere una colonna spaziale ("<nome-colonna>") alla tabella "<nome-schema.nome-tabella>".

Risposta utente: Verificare il messaggio associato a SQLSTATE "<sqlstate>".

GSE1008E DB2 Spatial Extender non è stato in grado di registrare una struttura vista "<schema-struttura.nome-struttura.colonna-struttura>" poiché il tipo di dati spaziali "<struttura-colonna-tipo>" della struttura vista non corrisponde al tipo di dati spaziali "<geo-colonna-tipo>" della tabella sottostante la struttura tabella "<schema-geo.nome-geo.colonna-geo>".

Spiegazione: Il tipo di dati spaziali della struttura vista "<schema-struttura.nome-struttura.colonna-struttura>" dovrà corrispondere al tipo di dati spaziali della struttura di tabella sottostante la struttura "<schema-geo.nome-geo.colonna-geo>". La mancata corrispondenza tra questi due tipi di dati causa problemi durante l'elaborazione dei dati spaziali.

Risposta utente: Verificare la corrispondenza tra i tipi di dati spaziali della struttura vista e la

relativa struttura tabella sottostante.

GSE1020E **"<id-riferimento-spaziale>" non è un ID di sistema di riferimento spaziale valido.**

Spiegazione: Non è stato abilitato alcun sistema di riferimento spaziale con identificativo "<id-riferimento-spaziale>".

Risposta utente: Verificare che il riferimento spaziale specificato sia stato abilitato.

GSE1021E **Spatial Extender non ha potuto abilitare il sistema di riferimento spaziale "<id-riferimento-spaziale>" poiché l'ID di sistema di coordinate spaziale corrispondente "<id-coordinata-spaziale>" non è valido.**

Spiegazione: Nel sistema di coordinate con identificativo "<id-coordinata-spaziale>" non è definito nel catalogo di Spatial Extender.

Risposta utente: Verificare l'identificativo di sistema di coordinate "<id-coordinate-spaziale>" nella vista DB2GSE.COORD_REF_SYS del catalogo di Spatial Extender.

GSE1030E **Poiché "<nome-schema.nome-tabella>" non è una tabella di base, Spatial Extender potrebbe non abilitare il relativo geocoder.**

Spiegazione: E' necessario che l'oggetto che contiene i dati di origine di cui si desidera eseguire un geocode corrisponda a una tabella di base.

Risposta utente: Verificare che le colonne che contengono i dati di origine di cui si desidera eseguire un geocode siano parte di una tabella di base.

GSE1031E **Spatial Extender non ha potuto abilitare il geocoder "<id-geocoder>" per operare automaticamente in modo creazione per la struttura "<schema-struttura.nome-struttura.colonna-struttura>".**

Spiegazione: Le possibili cause sono:

- Il geocoder è già abilitato per l'aggiornamento automatico della struttura "<schema-struttura.nome-struttura.colonna-struttura>".
- Il geocoder è stato disabilitato temporaneamente per questa struttura.
- Non è stato definito alcun dato di origine per questa struttura.

Risposta utente: Se il geocoder è stato disabilitato temporaneamente, abilitarlo per il funzionamento automatico in modalità ricreazione.

GSE1032E **Spatial Extender non ha potuto abilitare il geocoder "<id-geocoder>" per operare automaticamente in modo ricreazione per la struttura "<schema-struttura.nome-struttura.colonna-struttura>".**

Spiegazione: Le possibili cause sono:

- Il geocoder è già abilitato per l'aggiornamento automatico della struttura "<schema-struttura.nome-struttura.colonna-struttura>".
- Il geocoder non è stato disabilitato precedentemente per questa struttura.
- Non è stato definito alcun dato di origine per questa struttura.

Risposta utente: Se il geocoder è stato precedentemente disabilitato in modalità cancellazione o se non è mai stato definito per questa struttura, abilitarlo per il funzionamento automatico in modalità creazione.

GSE1033E Si è verificato un errore SQL durante il tentativo di Spatial Extender di aggiungere trigger alla tabella contenente la colonna per la struttura "`<schema-struttura.nome-struttura.colonna-struttura>`" (SQLSTATE "`<sqlstate>`").

Spiegazione: I trigger gestiscono l'integrità dei dati tra colonne di attributo da cui proviene l'input del geocoder e colonne spaziali cui è destinato il relativo output. L'errore SQL si è verificato durante il tentativo eseguito dal DB2 di creare tali trigger.

Risposta utente: Verificare il messaggio associato a SQLSTATE "`<sqlstate>`".

GSE1034E Spatial Extender non ha potuto disabilitare il geocoder "`<id-geocoder>`" in modo cancellazione per la struttura "`<schema-struttura.nome-struttura.colonna-struttura>`".

Spiegazione: Le possibili cause sono:

- Il geocoder non è stato abilitato per l'aggiornamento automatico della struttura "`<schema-struttura.nome-struttura.colonna-struttura>`".
- Il geocoder è stato disabilitato in modo cancellazione.

Risposta utente: Verificare lo stato del geocoder prima di tentare di disabilitarlo. Verificare ad esempio, se è stato registrato, oppure se è stato abilitato. Quindi stabilire se è necessario disabilitarlo in modo cancellazione. Se ad esempio non è mai stato abilitato, non sarà necessario disabilitarlo.

GSE1035E Spatial Extender non ha potuto disabilitare il geocoder "`<id-geocoder>`" in modo disabilitazione per la struttura "`<schema-struttura.nome-struttura.colonna-struttura>`".

Spiegazione: Le possibili cause sono:

- Il geocoder non è stato abilitato per l'aggiornamento automatico della struttura "`<schema-struttura.nome-struttura.colonna-struttura>`".
- Il geocoder è stato disabilitato in modo disabilitazione o in modo cancellazione.

Risposta utente: Verificare lo stato del geocoder prima di tentare di disabilitarlo. Verificare ad esempio, se è stato registrato, oppure se è stato abilitato. Quindi stabilire se è necessario disabilitarlo in modo disabilitazione. Ad esempio, se è stato già disattivato in modalità di invalidazione, non sarà necessario disabilitarlo una seconda volta in questa modalità.

GSE1036E Si è verificato un errore SQL quando Spatial Extender ha tentato di rilasciare i trigger da una tabella che contiene la colonna per la struttura "`<layer-schema.layer-name.layer-column>`" (SQLSTATE "`<sqlstate>`").

Spiegazione: I trigger sono stati creati per conservare l'integrità dei dati tra le colonne di attributo da cui proviene l'input del geocoder e la colonna spaziale in cui viene inserito l'output del geocoder. L'errore SQL si è verificato durante il tentativo eseguito dal DB2 di rilasciare tali trigger.

Risposta utente: Verificare il messaggio associato a SQLSTATE "`<sqlstate>`".

GSE1037E Spatial Extender non riesce a sottoporre a geocoding i dati relativi alla struttura della tabella "`<layer-schema.layer-name.layer-column>`", probabilmente perché è stato assegnato un valore non corretto "`<numero di attributi>`" all'argomento che specifica quante colonne di attributo devono dati di origine per questa struttura.

Spiegazione: Il numero di colonne di attributo associate a questa struttura o il nome di una o più colonne non è stato specificato correttamente.

Risposta utente: Verificare che questa struttura sia registrata con il numero corretto e i nomi delle colonne di attributo associate o verificare la correttezza dei dati di input e output per il geocoder.

GSE1038E Si è verificato un errore SQL, quando Spatial Extender ha tentato di sottoporre a geocoding i dati di origine relativi alla struttura della tabella "<layer-schema.layer-column>" in modalità batch (SQLSTATE "<sqlstate>").

Risposta utente:

- Verificare il messaggio associato a SQLSTATE "<sqlstate>".
- Verificare che il contenuto e l'argomento primaryUDF della struttura siano stati definiti correttamente.

GSE1050E La dimensione della griglia specificata ("<dimensione griglia>") non è valida per il primo livello di griglia.

Spiegazione: E' stato specificato zero o un numero negativo come dimensione della griglia per il primo livello della griglia.

Risposta utente: Specificare un numero positivo come dimensione della griglia.

GSE1051E La dimensione della griglia specificata ("<dimensione griglia>") non è valida per il secondo e in terzo livello della griglia.

Spiegazione: E' stato specificato un numero negativo come dimensione della griglia per il secondo e il terzo livello della griglia.

Risposta utente: Specificare zero o un numero positivo come dimensione della griglia.

GSE1052E Si è verificato un errore SQL, quando Spatial Extender ha tentato di creare un indice spaziale "<index-schema.index-column>" per una struttura di tabella "<layer-schema.layer-name.layer-column>" (SQLSTATE "<sqlstate>").

Risposta utente:

- Verificare che l'indice spaziale sia stato specificato correttamente e che la colonna spaziale non presenti un indice associato.
- Verificare il messaggio associato a SQLSTATE "<sqlstate>".

GSE1500I Il record di origine "<numero record>" è stato sottoposto a geocoding.

Spiegazione: Un record contenente i dati di attributo è stato sottoposto a geocoding.

GSE1501W Non è stato possibile sottoporre a geocoding il record di origine "<numero record>".

Spiegazione: Il livello di precisione era troppo alto.

Risposta utente: Eseguire il geocoding con un livello di precisione inferiore.

GSE1502W Il record di origine "<numero record>" non è stato individuato.

Risposta utente: Verificare che il record esista nel database.

GSE2001E Il file di trasferimento specificato ("<nome file>") non è valido.

Risposta utente: Verificare che il file specificato sia un file di trasferimento SDE e che il nome percorso sia stato definito correttamente.

GSE2002E **La clausola SQL WHERE fornita (“<clausola SQL-where>”) non è valida.**

Risposta utente: Verificare che la sintassi SQL della clausola WHERE sia corretta, controllare inoltre gli errori di digitazione e i nomi delle colonne.

GSE2003E **Il valore shape fornito non è legale.**

Risposta utente: Verificare che il valore shape fornito corrisponda al tipo di colonna spaziale specificato. Se i tipi corrispondono o sono compatibili, la forma della geometria non è legale. Verificare i poligoni sovrapposti, gli archi a punti singoli e così via.

GSE2004E **Lo schema del file di trasferimento non è compatibile con lo schema della struttura specificata.**

Risposta utente: Verificare che lo schema e i nomi della struttura sia specificati correttamente. Se gli schemi non corrispondono, caricare i dati come una nuova tabella e risolvere le differenze di schema.

GSE2005E **Il tipo di geometri del file di trasferimento non compatibile con il tipo di geometria della struttura specificata.**

Risposta utente: Verificare che lo schema e i nomi della struttura sia specificati correttamente.

GSE2006E **Si è verificato un errore I/O per un file definito “<nome file>”.**

Risposta utente: Verificare che il file esista, che si disponga del tipo di accesso appropriato al file e che il file non sia utilizzato da un altro utente.

GSE2007E **Si è verificato un errore di conversione degli attributi.**

Risposta utente: Verificare che tutti i tipi di attributi della tabella siano supportati - ad

esempio, i dati BLOB non sono supportati nei file shape. Verificare inoltre i valori dei dati esterni all'intervallo o i valori dei dati illegali come i dati corrotti.

GSE2008E **La funzione di importazione/esportazione ha esaurito la memoria.**

Risposta utente: Verificare che sia disponibile sufficiente memoria.

Capitolo 11. Viste del catalogo

Le viste del catalogo di DB2 Spatial Extender contengono metadati su:

- I sistemi di coordinate che è possibile utilizzare. Per informazioni come quelle relative agli identificatori di sistema e ai testi di annotazione, vedere "DB2GSE.COORD_REF_SYS".
- Le colonne spaziali sono state registrate come strutture. Per informazioni, come i nomi delle colonne, i tipi di dati e i sistemi di riferimento spaziale associati, vedere "DB2GSE.GEOMETRY_COLUMNS" a pagina 116.
- I geocoder che è possibile utilizzare. Per informazioni relative agli identificatori dei geocoder e alle regioni che contengono le posizioni che i geocoder elaborano, vedere "DB2GSE.SPATIAL_GEOCODER" a pagina 116.
- I sistemi di riferimento spaziale che è possibile utilizzare. Per informazioni sugli identificatori dei sistemi e sulle descrizioni di essi, vedere "DB2GSE.SPATIAL_REF_SYS" a pagina 117.

DB2GSE.COORD_REF_SYS

Quando si abilita un database alle operazioni spaziali, DB2 Spatial Extender registra i sistemi di coordinate che è possibile utilizzare in una tabella di catalogo. Le colonne selezionate da questa tabella comprendono la vista di catalogo DB2GSE.COORD_REF_SYS, descritta in Tabella 34.

Tabella 34. Colonne della vista della vista di catalogo DB2GSE.COORD_REF_SYS

Nome	Tipo di dati	Valori Null consentiti?	Contenuto
SCID	INTEGER	No	Identificatore numerico univoco per questo sistema di coordinate.
SC_NAME	VARCHAR(64)	No	Nome del sistema di coordinate.
AUTH_NAME	VARCHAR(256)	Sì	Il nome dell'azienda che ha compilato questo sistema di coordinate coincide con, ad esempio, European Petroleum Survey Group (EPSG).
AUTH_SRID	INTEGER	Sì	Un identificatore numerico assegnato a questo sistema di coordinate dall'azienda specificata nella colonna AUTH_NAME.
DESC	VARCHAR(256)	Sì	Descrizione del sistema di coordinate.
SRTEXT	VARCHAR(2048)	No	Testo di annotazione per il sistema di coordinate.

DB2GSE.GEOMETRY_COLUMNS

Quando si crea una struttura, DB2 Spatial Extender la registra, registrandone l'identificatore e le informazioni correlate in una tabella del catalogo. Le colonne selezionate da questa tabella comprendono la vista del catalogo DB2GSE.GEOMETRY_COLUMNS, descritta in Tabella 35.

Tabella 35. Colonne nella vista del catalogo DB2GSE.GEOMETRY_COLUMNS

Nome	Tipo di dati	Valori Null consentiti?	Contenuto
LAYER_CATALOG	VARCHAR(30)	Si	Nome completo della struttura.
LAYER_SCHEMA	VARCHAR(30)	No	Schema della tabella o della vista che contiene la colonna registrata come struttura.
LAYER_NAME	VARCHAR(128)	No	Nome della tabella o della vista che contiene la colonna registrata come struttura.
LAYER_COLUMN	VARCHAR(30)	No	Nome della colonna registrata come struttura.
GEOMETRY_TYPE	INTEGER	No	Tipo di dati della colonna registrata come struttura.
SRID	INTEGER	No	Identificatore del sistema di riferimento spaziale utilizzato per i valori della colonna registrata come struttura.
STORAGE_TYPE	INTEGER	Si	Informazioni relative alle modalità con cui DB2 memorizza i valori nella colonna registrata come struttura. Ad esempio, i dati in STORAGE_TYPE possono indicare che i valori sono stati memorizzati come oggetti di grandi dimensioni (LOB) o come istanze di tipi di dati astratti (ADT).

DB2GSE.SPATIAL_GEOCODER

I geocoder disponibili sono registrati in una tabella del catalogo. Le colonne selezionate da questa tabella comprendono la vista del catalogo DB2GSE.SPATIAL_GEOCODER, descritta in Tabella 36.

Tabella 36. Colonne della vista del catalogo DB2GSE.SPATIAL_GEOCODER

Nome	Tipo di dati	Valori Null consentiti?	Contenuto
GCID	INTEGER	No	Identificatore numerico di questo geocoder.
GC_NAME	VARCHAR(64)	No	Breve descrizione di questo geocoder.
VENDOR_NAME	VARCHAR(128)	No	Nome del fornitore del geocoder.

Tabella 36. Colonne della vista del catalogo DB2GSE.SPATIAL_GEOCODER (Continua)

Nome	Tipo di dati	Valori Null consentiti?	Contenuto
PRIMARY_UDF	VARCHAR(256)	No	Nome completo del geocoder.
PRECISION_LEVEL	INTEGER	No	Il livello di corrispondenza richiesto tra i dati di origine e i dati di riferimento per garantire una elaborazione corretta del geocoder.
VENDOR_SPECIFIC	VARCHAR(256)	Sì	Percorso e nome di un file che un fornitore può utilizzare per impostare i parametri speciali che questo geocoder supporta.
GEO_AREA	VARCHAR(256)	Sì	Area geografica che contiene le posizione da sottoporre a geocoding.
DESCRIPTION	VARCHAR(256)	Sì	Note del fornitore.

DB2GSE.SPATIAL_REF_SYS

Quando si crea un sistema di riferimento spaziale, DB2 Spatial Extender lo registra, registrandone l'identificatore e le informazioni correlate in una tabella del catalogo. Le colonne selezionate da questa tabella comprendono la vista del catalogo DB2GSE.SPATIAL_REF_SYS, descritta in Tabella 37.

Tabella 37. Colonne della vista del catalogo DB2GSE.SPATIAL_REF_SYS

Nome	Tipo di dati	Valori Null consentiti?	Contenuto
SRID	INTEGER	No	Identificatore definito dall'utente per questo sistema di riferimento spaziale.
SR_NAME	VARCHAR(64)	No	Nome del sistema di riferimento spaziale.
SCID	INTEGER	No	Identificatore numerico per il sistema di coordinate che sottostà al sistema di riferimento spaziale.
SC_NAME	VARCHAR(64)	No	Nome del sistema di coordinate che sottostà al sistema di riferimento spaziale.
AUTH_NAME	VARCHAR(256)	Sì	Nome dell'azienda che imposta gli standard per questo sistema di riferimento spaziale.
AUTH_SRID	INTEGER	Sì	L'identificatore che l'azienda ha specificato nella colonna AUTH_NAME viene assegnato a questo sistema di riferimento spaziale.
SRTEXT	VARCHAR(2048)	No	Testo di annotazione per questo sistema di riferimento spaziale.

Capitolo 12. indici spaziali

Poiché le colonne spaziali contiene dati geografici bidimensionali, le applicazioni che interrogano tali colonne richiedono una strategia di indice che identifica rapidamente tutte le geometrie comprese in una determinata lunghezza. Per questa ragione DB2 Spatial Extender fornisce un indice spaziale a tre livelli basato su una griglia.

In questo capitolo si descrive questo tipo di indice e si forniscono informazioni sul relativo utilizzo. Argomenti trattati:

- “Frammento di programma di esempio”
- “indici con struttura ad albero B” a pagina 120
- “Modalità di creazione di un indice spaziale” a pagina 120
- “Modalità di generazione di un indice spaziale” a pagina 121
- “Istruzioni sull’utilizzo di un indice spaziale” a pagina 125

Frammento di programma di esempio

Di seguito è riportato un esempio di creazione e utilizzo di un indice SQL. Per ulteriori informazioni sui comandi CREATE INDEX e CREATE INDEX EXTENSION, consultare *SQL Reference*. Si noti che dopo aver creato un indice, è possibile immettere istruzioni standard DDL e DML che utilizzano funzioni e predicati spaziali.

```
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 spatial_index(10e0,
100e0, 1000e0)
create index storesx1 on stores(loc) extend using spatial_index(10e0, 100e0,
1000e0)
create index storesx2 on stores(zone) extend using 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)

```

indici con struttura ad albero B

La tecnologia di indicizzazione spaziale è basata sull'indice con struttura ad albero B gerarchico tradizionale, ma presenta differenze significative. L'indice spaziale utilizza l'*indicizzazione di griglia* progettata per l'indicizzazione di colonne spaziali bidimensionali. L'indice con struttura ad albero B può gestire solo dati a una dimensione e non può essere utilizzato con informazioni GIS. Questa sezione descrive la struttura e l'utilizzo di un indice con struttura ad albero B.

Il livello superiore di un indice con struttura ad albero B, denominato nodo radice, contiene una chiave per ciascun nodo al livello successivo. Il valore di ciascuna chiave è il valore chiave più elevato per il nodo corrispondente al livello successivo. Il numero di nodi intermedi varia e seconda del numero di valori presente nella tabella di base. Tali nodi formano un ponte tra il nodo radice e i nodi dipendenti contenenti gli ID riga della tabella di base corrente.

Database manager ricerca gli indici di struttura B partendo dal nodo radice. Prosegue quindi attraverso i nodi intermedi fino a raggiungere il nodo dipendente che include l'ID riga della tabella di base.

L'indice con struttura ad albero B non può essere applicato alla colonna spaziale poiché la caratteristica bidimensionale della colonna spaziale richiede la struttura di un indice spaziale. Non è possibile pertanto applicare un indice spaziale a una colonna non spaziale. Inoltre l'indice spaziale non può essere applicato a una colonna composta di alcun tipo.

Modalità di creazione di un indice spaziale

Esistono varie procedure di creazione di un indice spaziale:

- Definirlo dalla finestra Creare indice spaziale. Per le istruzioni relative, vedere il "Capitolo 6. Creazione di indici spaziali" a pagina 51.

- Richiamare la procedura memorizzata `db2gse.gse_enable_idx` in un programma applicativo. Per informazioni relative a una procedura memorizzata, vedere il “Capitolo 9. Procedure memorizzate” a pagina 67.
- Immettere il comando **db2 create index** con la funzione **spatial_index** nella clausola **USING**. Ad esempio:


```
create index storesx1 on customers (loc) using spatial_index(10e0,
100e0, 1000e0)
```

La natura dei dati spaziali richiede che i progettisti del database considerino la relativa distribuzione di dimensione. I progettisti dovranno determinare la dimensione ottimale e il numero di livelli di griglia necessari per creare l'indice spaziale.

I livelli di griglia, <livello griglia 1>, <livello griglia 2> e <livello griglia 3>, vengono impostati aumentando la dimensione delle celle. Pertanto il secondo livello dovrà presentare una dimensione di cella superiore al primo livello e il terzo superiore al secondo. Il primo livello di griglia è obbligatorio, ma è possibile disattivare il secondo e il terzo con un valore zero a doppia precisione (0.0e0).

Modalità di generazione di un indice spaziale

Un indice spaziale viene generato utilizzando le *buste*. La busta è una geometria che rappresenta la lunghezza minima e massima X e Y di una geometria. Per la maggior parte delle geometrie, la busta corrisponde a una casella, ma per le stringhe di righe orizzontali e verticali la busta corrisponde a una stringa di riga a due punti. Per i punti, la busta corrisponde al punto. Per ulteriori informazioni relative alle buste, vedere la sezione “Inviluppo” a pagina 134.

L'indice spaziale viene costruito su una colonna spaziale, creando una o più voci per le intersezioni di ciascuna busta di geometria con la griglia. Le intersezioni vengono registrate come ID interno della geometria e coordinate minime X e Y di intersezione della cella di griglia. Ad esempio il poligono riportato nella Figura 7 a pagina 122 presenta un'intersezione con la griglia sulle coordinate (20,30), (30,30), (40,30), (20,40), (30,40), (40,40), (20,50), (30,50) e (40,50). Vedere la Tabella 38 a pagina 122 per le coordinate minime X e Y per tutte le geometrie riportate nella Figura 7 a pagina 122.

Se esistono più livelli di griglia, DB2 Spatial Extender tenterà di utilizzare il livello di griglia inferiore possibile. Le geometrie che intersecano quattro o più celle di griglia a un determinato livello, verranno promosse al successivo livello superiore. Pertanto se un indice spaziale presenta i tre livelli di griglia 10.0e0, 100.0e0 e 1000.0e0, DB2 Spatial Extender intersecherà ciascuna geometria con il livello di griglia 10.0e0. Le geometrie che intersecano quattro o più celle di griglia 10.0e0, verranno promosse e intersecate con la griglia di

livello 100.0e0. Se quattro o più intersezioni risultano al livello 100.0e0, la geometria viene promossa al livello 1000.0e0. Al livello 1000.0e0, le intersezioni dovranno essere immesse nell'indice spaziale che è il massimo livello possibile.

La Figura 7 illustra le modalità di intersezione tra i quattro diversi tipi di geometrie e la griglia 10.0e. Tutte le 23 intersezioni per le quattro geometrie vengono registrate nell'indice spaziale.

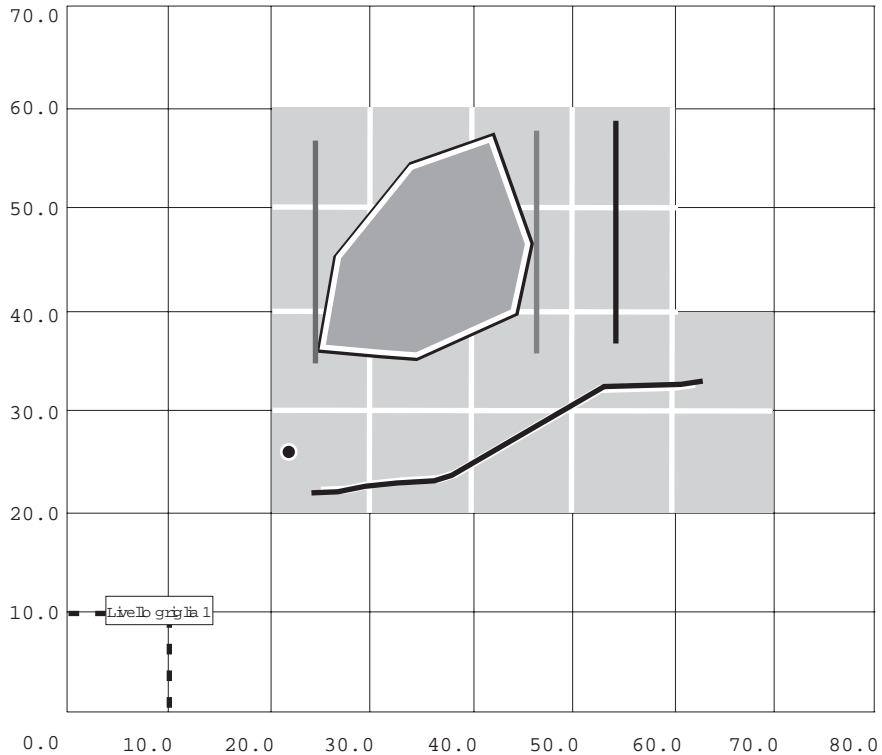


Figura 7. Applicazione di un livello di griglia 10.0e0

La Tabella 38 elenca le geometrie e le intersezioni di griglia corrispondenti. Le buste di quattro diversi tipi di geometrie intersecano la griglia 10.0e. Le coordinate minime X e Y di ciascuna cella di griglia da esse intersecata, vengono immesse nell'indice spaziale.

Tabella 38. Le voci della cella di griglia 10.0e0 per le geometrie di esempio

Geometria	Griglia X	Griglia Y
Poligono	20.0	30.0
Poligono	30.0	30.0

Tabella 38. Le voci della cella di griglia 10.0e0 per le geometrie di esempio (Continua)

Geometria	Griglia X	Griglia Y
Poligono	40.0	30.0
Poligono	20.0	40.0
Poligono	30.0	40.0
Poligono	40.0	40.0
Poligono	20.0	50.0
Poligono	30.0	50.0
Poligono	40.0	50.0
Stringa di riga verticale	50.0	30.0
Stringa di riga verticale	50.0	40.0
Stringa di riga verticale	50.0	50.0
Punto	20.0	20.0
Stringa di riga orizzontale	20.0	20.0
Stringa di riga orizzontale	30.0	20.0
Stringa di riga orizzontale	40.0	20.0
Stringa di riga orizzontale	50.0	20.0
Stringa di riga orizzontale	60.0	20.0
Stringa di riga orizzontale	20.0	30.0
Stringa di riga orizzontale	30.0	30.0
Stringa di riga orizzontale	40.0	30.0
Stringa di riga orizzontale	50.0	30.0
Stringa di riga orizzontale	60.0	30.0

La Figura 8 a pagina 124 mostra come il numero di intersezioni viene ridotto a otto aggiungendo i livelli di griglia 30.0e0 e 60.0e0. In questo caso, il poligono identificato come geometria 1 viene promosso al livello di griglia 30.0e0 e la stringa di riga identificata come geometria 4 viene promossa al livello di griglia 60.0e0. Dopo la promozione le intersezioni delle geometrie al livello 10.0e0 verranno ridotte da nove e dieci a due.

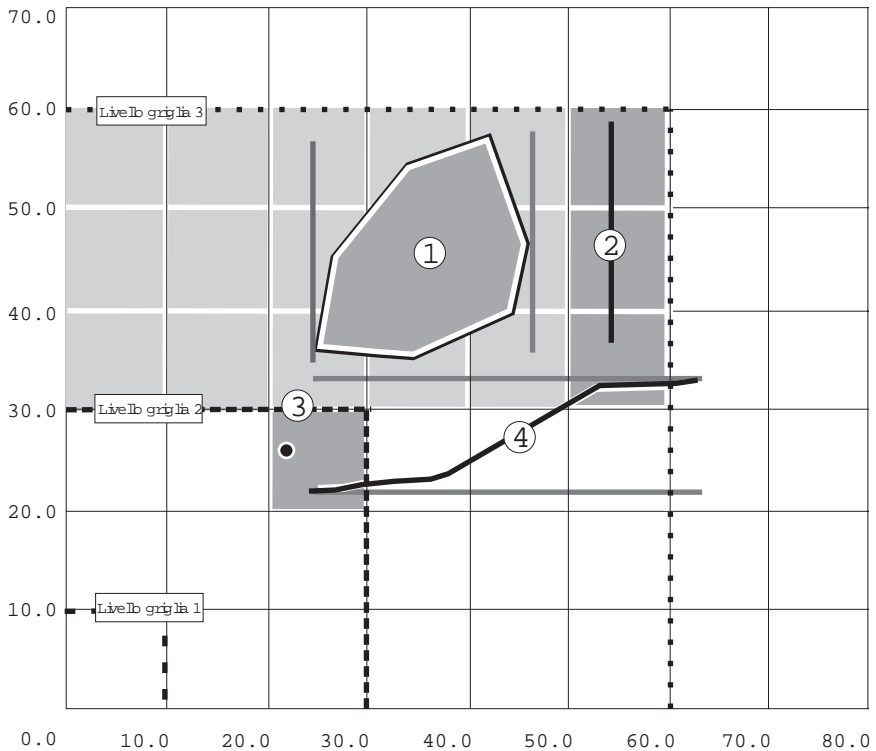


Figura 8. Effetti dell'aggiunta di livelli di griglia 30.0e0 e 60.0e0. La busta del poligono identificata come geometria 1 interseca nove celle di griglia. La busta della riga di stringa verticale identificata come geometria 2 interseca tre celle di griglia. La busta del punto identificata come geometria 3 interseca solo una cella di griglia. La busta della riga di stringa identificata come geometria 4 interseca dieci celle di griglia.

DB2 Spatial Extender utilizza i parametri di livello di griglia specificati nell'istruzione CREATE INDEX e verifica ciascun oggetto spaziale per determinare le coordinate e il numero di blocchi di griglia contenenti l'oggetto. Nella Figura 8, i livelli di griglia 10.0e0, 30.0e0 e 60.0e0 vengono illustrate righe crescenti di diverso spessore e differenti gradazioni di grigio. Le intersezioni tra la stringa di riga verticale e la cella di busta del punto vengono inserite nell'indice a livello di griglia 10.0e0, poiché entrambe generano meno di quattro intersezioni. Il poligono interseca nove celle di griglia 10.0e0 e viene pertanto promosso al livello di griglia 30.0e0. A questo livello, il poligono interseca due celle di griglia, che vengono immesse nell'indice. La stringa di riga identificata come geometria 4 interseca dieci celle di griglia 10.0e0 e viene pertanto promossa al livello di griglia 30.0e0. A questo livello interseca quindi sei celle di griglia e verrà quindi promossa al livello di griglia 60.0e0, in cui genera due intersezioni. Le intersezioni di griglia 60.0e0 della stringa di riga vengono inserite in due indici. Se la stringa di riga ha generato quattro o più intersezioni a questo livello, queste verranno sempre inserite nell'indice che rappresenta il livello superiore a cui una

geometria può essere promossa.

Tabella 39. Intersezioni delle geometrie nell'indice a tre livelli

Geometria	Griglia X	Griglia Y
<i>Le intersezioni tra la stringa di riga verticale e il punto del livello 1 (dimensione griglia 10.0e0)</i>		
2	50.0	30.0
2	50.0	40.0
2	50.0	50.0
3	20.0	20.0
<i>Le intersezioni del poligono nel livello 2 (dimensione di griglia 30.0e0)</i>		
1	0.0	30.0
1	30.0	30.0
<i>Le intersezioni della stringa di riga nel livello 3 (dimensione di griglia 60.0e0)</i>		
4	0.0	0.0
4	60.0	0.0

DB2 Spatial Extender attualmente non crea alcuna struttura di griglia del poligono. DB2 Spatial Extender indica ciascun livello di griglia in parametri, definendo l'origine all'offset X,Y del sistema di riferimento spaziale della colonna. Estende quindi la griglia in spazio di coordinata positivo. Utilizzando una griglia parametrica, DB2 Spatial Extender generate matematicamente le intersezioni.

Istruzioni sull'utilizzo di un indice spaziale

DB2 Spatial Extender utilizza l'indice spaziale per migliorare le prestazioni delle interrogazioni spaziali. L'interrogazione base più diffusa è l'interrogazione casella. Essa richiede a DB2 Spatial Extender tutte le geometrie comprese parzialmente o totalmente in una casella definita dall'utente. Se l'indice non esiste, DB2 Spatial Extender dovrà confrontare tutte le geometrie con la casella. Tuttavia DB2 Spatial Extender potrà utilizzare l'indice per individuare tutte le voci di indice la cui coordinata inferiore sinistra è maggiore o uguale alla casella e la cui coordinata superiore destra è minore o uguale alla casella. Poiché l'indice viene ordinato da questo sistema di coordinate, DB2 Spatial Extender potrà richiamare rapidamente un elenco di geometrie candidate. Il processo appena illustrato viene denominato *primo passo*.

Un *secondo passo* determina se ciascuna busta di candidato interseca la casella. Una geometria adatta al primo passo poiché la relativa busta delle celle di griglia intersecano la casella, può presentare una busta che non viene intersecata.

Un *terzo passo* confronta le coordinate attuali del candidato con la casella per stabilire se una parte della geometria è attualmente contenuta nella casella. Quest'ultimo processo più complesso di confronto utilizza un elenco di candidati composto da una serie secondaria di elementi totali, che viene ridotto sensibilmente dai primi due passi.

Tutte le interrogazioni spaziali eseguono i tre passi ad eccezione della funzione `EnvelopesIntersect`. Questa funzione esegue solo i primi due passi. La funzione `EnvelopesIntersect` è stata progettata per la visualizzazione di operazioni che spesso prevedono l'impiego di routine clipping incorporate e non richiede la granularità del terzo passo.

Selezione della dimensione delle celle di griglia

La forma irregolare delle buste di geometria complica la selezione della dimensione delle celle di griglia. A causa di questa irregolarità, alcune buste di geometria intersecano più griglie, invece altre utilizzano una sola cella di griglia. Altre celle di griglia, al contrario, a seconda della distribuzione spaziale dei dati, intersecano molte buste di geometria.

Per un corretto funzionamento dell'indice spaziale, è necessario selezionare un numero e una dimensione appropriata di griglie. Si consideri una colonna spaziale contenente geometrie con dimensioni uniformi. In tal caso sarà sufficiente un singolo livello di griglia. Iniziare utilizzando una dimensione di cella di griglia che può comprendere una busta di geometria media. Durante la verifica dell'applicazione, si noterà che aumentando la cella di griglia le prestazioni delle interrogazioni risulteranno notevolmente migliorate. Ciò avviene poiché ciascuna cella di griglia contiene più geometrie e pertanto il primo passo è in grado di scaricare geometrie non specificate in modo più rapido. Si noterà tuttavia che continuando ad aumentare la dimensione di cella le prestazioni diminuiranno. Ciò avviene poiché il secondo passo verrà ostacolato da più candidati.

Selezione del numero di livelli

Se gli oggetti che si desidera indicizzare sono della stessa dimensione relativa, è possibile utilizzare un singolo livello di griglia. Anche in questo caso non tutte le colonne conterranno geometrie della stessa dimensione relativa. Di solito le geometrie delle colonne spaziali possono essere raggruppate in diversi intervalli di dimensioni. Si consideri ad esempio una rete di strade in cui le geometrie sono divise in strade, strade principali e autostrade. Le strade sono circa della stessa lunghezza e possono essere raggruppate in un intervallo di dimensione. Lo stesso dicasi per le strade principali e le

autostrade. E' possibile pertanto raggruppare le strade che rappresentano un intervallo di dimensione in un primo livello di griglia, le reti di strade in un secondo livello e le autostrade in un terzo livello. Un altro esempio include una colonna relativa al paese che contiene raggruppamenti di piccoli centri urbani circondati da aree rurali più vaste. In questo esempio si utilizzano due intervalli di dimensioni e due livelli di griglia uno per i piccoli centri urbani e l'altro per le aree rurali più vaste. Queste situazioni sono molto frequenti e richiedono l'utilizzo di una griglia multilivello.

Per selezionare la dimensione di cella per ciascun livello di griglia, selezionare dimensioni di cella di griglia più elevate rispetto a ciascun intervallo di dimensione. Verificare l'indice eseguendo interrogazioni delle colonne spaziali.

Ciascun livello aggiuntivo richiede la scansione di indici aggiuntivi. Provare ad aumentare o diminuire le dimensioni di griglia per determinare se le prestazioni vengono migliorate.

Capitolo 13. Geometrie e funzioni spaziali associate

Questo capitolo descrive le unità di informazioni, denominate *geometrie* che sono composte da coordinate e che rappresentano gli elementi geografici. Il capitolo inoltre descrive funzioni spaziali che rilevano le geometrie come input e restituiscono risultati che consentono di analizzare gli elementi geografici e di spostare i dati spaziali tra i sistemi di informazioni geografiche. Argomenti trattati:

- La natura delle geometrie
- Proprietà delle geometrie; le funzioni che restituiscono informazioni correlate a queste proprietà.
- Geometrie per le quali è possibile eseguire istanze; funzioni che vengono utilizzate con tali geometrie
- Funzioni che:
 - Mostrano le relazioni e le similitudini tra elementi geografici
 - Generano geometrie
 - Convertono i valori delle geometrie in formati che è possibile esportare e importare

Informazioni sulle geometrie

Nel dizionario il termine *geometria* "è definito come ramo della matematica che utilizza le proprietà e le relazioni tra righe, angoli, superfici e solidi." Nell'Agosto 11, 1997, la Open GIS Consortium Inc. (OGC) nella relativa pubblicazione, *Open GIS Features for ODBC (SQL) Implementation Specification*, ha coniato un'altra definizione per il termine. La parola *geometria* è stata selezionata per indicare gli elementi geometrici utilizzati, nel millennio passato, dai cartografi per creare carte geografiche. Una definizione generica di questo nuovo significato del termine geometria potrebbe essere "un punto o un aggregato di punti che rappresenta un elemento della terra."

In DB2 Spatial Extender, una definizione *operativa* di geometria potrebbe essere "modello di un elemento geografico." Il modello potrebbe essere espresso in termini di coordinate dell'elemento e, in alcuni casi, in termini di simboli visivi. Il modello converte le informazioni; ad esempio, le coordinate identificano la posizione dell'elemento rispettando i punti fissi di riferimento e il simbolo traccia la relativa forma. Inoltre, è possibile utilizzare il modello per produrre informazioni; ad esempio, la funzione `ST_Overlaps` può rilevare le coordinate di due regioni simili come input e restituisce informazioni per rilevare se è possibile sovrapporre le regioni.

Le coordinate di un'elemento rappresentate da una geometria sono considerate *proprietà* della geometria. Vari tipi di geometrie presentano altre proprietà; ad esempio:

- Un *interno* rappresenta il contenuto dell'elemento della geometria.
- Un *esterno* rappresenta lo spazio intorno all'elemento.
- Un *limite* rappresenta il confine tra il contenuto e l'area circostante.

Queste e altre proprietà sono discusse nella sezione "Proprietà delle geometrie e delle funzioni associate" a pagina 131.

Le geometrie supportate da DB2 Spatial Extender formano una gerarchia, illustrata nella Figura 9. E' possibile eseguire l'istanza di sei membri della gerarchia; tali membri possono essere espressi come simboli visivi, come illustrato nell'elemento.

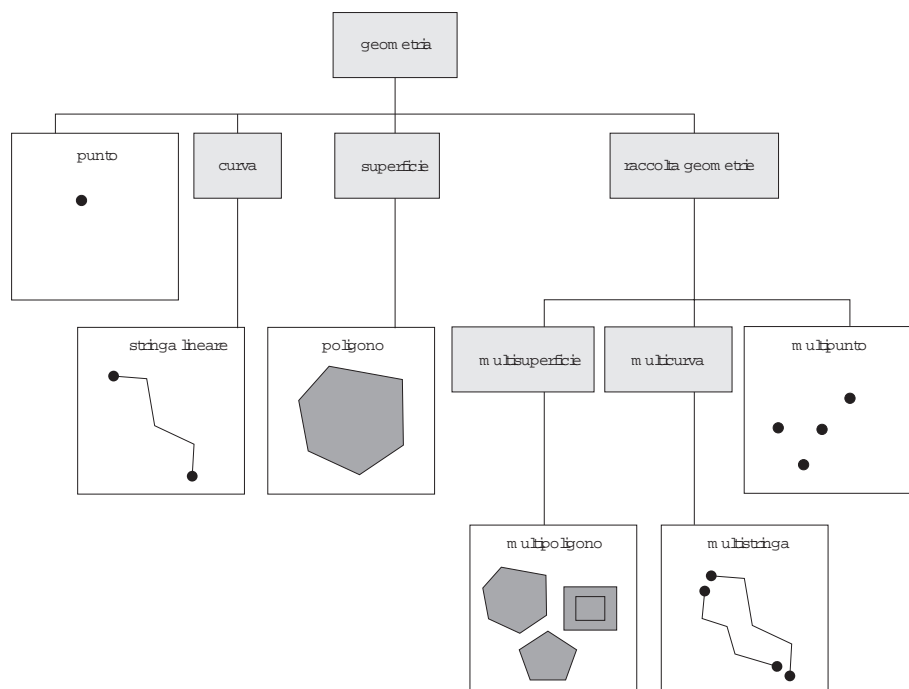


Figura 9. Gerarchia delle geometrie supportate da DB2 Spatial Extender. Le geometrie istanziabili possono essere espresse come simboli visivi. Questi simboli sono riportati con i nomi di tali geometrie.

Come riportato nella Figura 9, una classe principale denominata *geometria* è la radice della gerarchia. Le classi secondarie si suddividono in due categorie: le classi secondarie della geometria di base e le classi secondarie della raccolta omogenea. Le geometrie di base includono:

- *Punti*, che indicano gli elementi che occupano il luogo in cui una linea di coordinate est-ovest (ad esempio un parallelo) interseca una linea di coordinate nord-sud (ad esempio un meridiano). Ad esempio, supporre che in una scala cartografica ciascuna città della carta si trova nell'intersezione di un parallelo e un meridiano. In questa scala, ciascuna città può essere indicata da un punto.
- *Stringhe lineari*, indicano gli elementi geografici lineari (ad esempio, strade e canali).
- *Poligoni*, indicano gli elementi geografici con più lati (ad esempio, un distretto pubblico, foreste e zone abitate da animali selvaggi).

Le raccolte omogenee includono:

- *Multipunti*, indicano gli elementi con più parti i cui componenti sono situati nell'intersezione di una linea di coordinate est-ovest e una linea di coordinate nord-sud (ad esempio, una catena di isole i cui membri sono situati nell'intersezione di un parallelo e un meridiano).
- *Multistringhe lineari*, indicano gli elementi con più parti composti da unità o componenti lineari (ad esempio, sistemi fluviali e sistemi stradali).
- *Multipoligoni*, indicano gli elementi con più parti composti da unità o componenti con più lati (ad esempio, le zone agricole di una regione specifica o un sistema di laghi).

Come specificato dal nome stesso, le raccolte omogenee sono raccolte di geometrie di base. Le raccolte omogenee, oltre a condividere le proprietà della geometria di base, presentano anche proprie proprietà.

I tipi di dati spaziali supportati da DB2 Spatial Extender sono implementazioni delle geometrie illustrate nella Figura 9 a pagina 130. Per una descrizione di tali tipi di dati, consultare la sezione "Informazioni relative ai tipi di dati spaziali" a pagina 33.

Proprietà delle geometrie e delle funzioni associate

Questa sezione descrive le proprietà delle geometrie e le funzioni spaziali associate a queste proprietà. Nella seguente sezione vengono riportate le proprietà principali:

- Classe di appartenenza di una geometria
- Coordinate X e Y

Inoltre fornisce informazioni su:

- coordinate Z
- Misure
- Interno, esterno e limite di una geometria
- La qualità semplice o non semplice

- La qualità vuoto o non vuoto
- L'involuppo della geometria
- Dimensione
- Identificativo di un sistema di riferimento spaziale associato alla geometria

Classe

Ciascuna geometria appartiene a una classe della gerarchia illustrata nella Figura 9 a pagina 130. Come indicato nella sezione "Informazioni sulle geometrie" a pagina 129, le sei classi secondarie della gerarchia—punti, stringhe lineari, poligoni, multipunti, multistringhe lineari e multipoligoni— sono istanziabili. La classe principale e le altre classi secondarie non sono istanziabili.

La funzione `ST_GeometryType` rileva una geometria e restituisce la classe secondaria istanziabile nel formato di una stringa di caratteri. Per ulteriori informazioni consultare "ST_GeometryType" a pagina 222.

La funzione `ST_IsValid` rileva una geometria che è stata assegnata a un tipo di dati `ST_Geometry`. La funzione restituisce 1 (TRUE) se la geometria è valida e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare "ST_IsValid" a pagina 238.

Coordinate X e Y

Un *valore di coordinata X* indica un'ubicazione relativa a un punto con riferimento est-ovest. Un *valore di coordinata Y* indica un'ubicazione relativa a un punto con riferimento nord-sud. Per ulteriori informazioni, consultare le sezioni "Struttura dei dati spaziali" a pagina 6 e "Informazioni relative ai sistemi di riferimento spaziali e di coordinate" a pagina 25.

Coordinate Z

Alcune geometrie non presentano un'altitudine o una profondità associata. Ogni punto che forma la geometria di un elemento può includere una coordinata Z che rappresenta un'altitudine o una profondità normale della superficie terrestre.

La funzione predicato `Is3d` rileva una geometria e restituisce 1 (TRUE) se la funzione presenta coordinate Z e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare "Is3d" a pagina 170.

Misure

Una misura è un valore che indica le informazioni relative a un elemento geografico e che vengono memorizzate con le coordinate che definiscono l'ubicazione dell'elemento. Si supponga ad esempio che si desideri rappresentare i sistemi di trasporto nel proprio GIS. Se si desidera che l'applicazione elabori i valori che denotano le distanze lineari o in miglia, è

possibile memorizzare questi valori con le coordinate che definiscono le postazioni dei sistemi. Le misure vengono memorizzate come numeri di precisione doppia.

Il predicato `IsMeasured` rileva una geometria e restituisce 1 (TRUE) se contiene misure, 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare “`IsMeasured`” a pagina 171.

Interno, limite ed esterno

Tutte le geometrie occupano una posizione nell’area definita dall’interno, il limite e l’esterno. L’esterno di una geometria è l’area non occupata dalla geometria. Il limite di una geometria rappresenta l’interfaccia tra l’esterno e l’interno. L’interno è lo spazio occupato dalla geometria. Le classi secondarie utilizzano le stesse proprietà dell’interno ed esterno, mentre la proprietà del limite differisce per ciascuna di esse.

La funzione `ST_Boundary` rileva una geometria e ne restituisce una che rappresenta il limite della geometria di origine. Per ulteriori informazioni consultare “`ST_Boundary`” a pagina 192.

Semplice o non semplice

Alcune classi secondarie della geometria (stringhe lineari, multipunti e multistringhe lineari) presentano entrambe le qualità, semplice o non semplice. Una classe secondaria è semplice se rispetta tutte le regole topologiche imposte alla classe secondaria, non è semplice in caso contrario. Una stringa lineare è semplice se non interseca il proprio interno. Un multipunto è semplice se nessuno dei relativi elementi occupa la stessa area di coordinate. Una multistringa lineare è semplice se nessuno interno dei relativi elementi è intersecato dal proprio interno.

La funzione predicato `ST_IsSimple` rileva una geometria e restituisce 1 (TRUE) se la geometria è semplice e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare “`ST_IsSimple`” a pagina 237.

Vuoto o non vuoto

Una geometria è vuota se non presenta alcun punto. L’involuppo, il limite, l’interno e l’esterno di una geometria vuota è NULL. Una geometria vuota è sempre semplice e può presentare misure e coordinate Z. Le stringhe e le multistringhe lineari vuote hanno una lunghezza uguale a 0. I poligoni e i multipoligoni hanno un’area 0.

La funzione predicato `ST_IsEmpty` rileva una geometria e restituisce 1 (TRUE) se la geometria è vuota e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare “`ST_IsEmpty`” a pagina 234.

Inviluppo

L'inviluppo di una geometria è la geometria limite formata dalle coordinate minimo e massimo (X,Y). Con le seguenti eccezioni, gli inviluppi della maggior parte delle geometrie formano un rettangolo con limite:

- L'inviluppo di un punto è il punto stesso, in quanto le coordinate di minimo e massimo sono uguali.
- L'inviluppo di una stringa lineare verticale od orizzontale è una stringa lineare rappresentata dal limite (i punti finali) della stringa lineare di origine.

La funzione ST_Envelope rileva una geometria e restituisce una geometria con limite che rappresenta il relativo inviluppo. Per ulteriori informazioni consultare "ST_Envelope" a pagina 212.

Dimensione

Una geometria può presentare una dimensione uguale a 0, 1 o 2. Le dimensioni sono elencate di seguito:

- 0 Non presenta alcuna lunghezza o area
- 1 Presenta una lunghezza
- 2 Contiene area

Le classi secondarie punto e multipunto presentano una dimensione zero. I punti rappresentano le funzioni dimensionali che è possibile modellare con una coordinata singola, mentre le classi secondarie del multipunto rappresentano i dati che è necessario modellare con un cluster delle coordinate scollegate.

Le classi secondarie stringa e multistringa lineari hanno una dimensione 1. Tali classi memorizzano segmenti di strade, sistemi di diramazioni fluviali e altri elementi lineari della natura.

Le classi secondarie poligono e multipoligono presentano una dimensione due. Gli elementi il cui perimetro racchiude un'area definibile, quali foreste, porzioni di terra e bacini idrici che possono essere rappresentati dal tipo di dati poligono o multipoligono.

La dimensione è importante non solo come proprietà della classe secondarie, ma anche per la determinazione della relazione spaziale di due elementi. La dimensione dell'elemento o l'elemento derivanti determina se l'operazione è stata eseguita regolarmente. DB2 Spatial Extender esamina la dimensione degli elementi per determinare se è possibile confrontarli.

La funzione ST_Dimension rileva una geometria e restituisce la relativa dimensione come valore intero. Per ulteriori informazioni consultare "ST_Dimension" a pagina 206.

SRID (Spatial reference system identifier)

Il sistema di riferimento spaziale identifica la trasformazione delle coordinate per ciascuna geometria.

E' possibile accedere a tutti i sistemi di riferimento spaziale noti al database mediante la vista catalogo DB2GSE.SPATIAL_REF_SYS. Per informazioni relative a questa vista, consultare la sezione "Capitolo 11. Viste del catalogo" a pagina 115.

La funzione ST_SRID rileva una geometria e restituisce il relativo identificativo di riferimento spaziale come valore intero. Per ulteriori informazioni consultare "ST_SRID" a pagina 270.

La funzione ST_Transform assegna una geometria a un sistema di riferimento spaziale diverso da quello correntemente assegnato alla geometria. Per ulteriori informazioni consultare "ST_Transform" a pagina 275.

Geometrie istanziabili e funzioni associate

Questa sezione descrive le sei classi secondarie delle geometrie istanziabili e descrive le funzione associate. Le classi secondarie sono:

- Punti
- Stringhe lineari
- Poligoni
- Multipunti
- Multistringhe lineari
- Multipoligoni

Per illustrazioni della gerarchia di tali classi secondarie e dei simboli visivi associati, consultare la Figura 9 a pagina 130.

Punti

Un punto è una geometria di dimensione zero che occupa una singola postazione in un'area di coordinate. Un punto include una coordinata X e una coordinata Y che definisce questa postazione. Inoltre può includere una coordinata Z e una misura.

Un punto è semplice e presenta un limite NULL. I punti sono spesso utilizzati per definire elementi quali pozzi petroliferi, punti di riferimento terrestri etc.

Le funzioni che vengono utilizzate unicamente con le classi secondarie del punto:

ST_Point

Rileva una coordinata X, la relativa coordinata Y e l'identificativo del sistema di riferimento spaziale cui appartengono queste coordinate e restituisce il punto definito dalle coordinate. Per ulteriori informazioni consultare "ST_Point" a pagina 261.

ST_CoordDim

Restituisce un valore che indica le coordinate contenute in un punto e se il punto contiene una misura. Questo valore è denominato *dimensione delle coordinate*. Le dimensioni delle coordinate possibili sono:

- 2 Il punto consiste di una coordinata X e una coordinata Y.
- 3 Il punto consiste di una coordinata X, una coordinata Y e una coordinata Z.
- 4 Il punto consiste di una coordinata X, una coordinata Y, una coordinata Z e una misura.

Per ulteriori informazioni consultare "ST_CoordDim" a pagina 201.

ST_PointFromText

Rileva una rappresentazione WKT (well-known text) OGC di un punto e restituisce il punto. Per ulteriori informazioni consultare "ST_PointFromText" a pagina 259.

ST_X Restituisce una valore della coordinata X del tipo di dati ST_Point come numero di precisione doppia. Per ulteriori informazioni consultare "ST_X" a pagina 281.

ST_Y Restituisce una valore della coordinata Y del tipo di dati ST_Point come numero di precisione doppia. Per ulteriori informazioni consultare "ST_Y" a pagina 282.

Z Restituisce una valore della coordinata Z del tipo di dati ST_Point come numero di precisione doppia. Per ulteriori informazioni consultare "Z" a pagina 283.

M Restituisce una misura del tipo di dati ST_Point come numero di precisione doppia. Per ulteriori informazioni consultare "M" a pagina 178.

Stringhe lineari

Una stringa lineare è un oggetto monodimensionale memorizzato come sequenza di punto che definiscono un percorso di interpolazione lineare. La stringa lineare è semplice se non interseca il proprio interno. I punti finali (il limite) di una stringa lineare chiusa occupano lo stesso punto nell'area. Una

stringa lineare è un anello se è chiusa e se il relativo interno non presenta intersezioni. Oltre alle altre proprietà ereditate dalla geometria della superclasse, le stringhe lineari presentano una lunghezza. Le stringhe lineari sono spesso utilizzate per definire elementi lineari quali strade, fiumi etc.

Una stringa lineare semplice il cui punto iniziale e quello finale corrispondono è denominata *anello*.

I punti finali generalmente formano il limite di una stringa lineare a meno la stringa lineare non sia chiusa, in tal caso il limite è NULL. L'interno di una stringa lineare è il percorso di collegamento tra i punti finali, a meno che non sia chiusa, in tal caso l'interno è continuo.

Le funzioni che vengono utilizzate con le stringhe lineari:

ST_StartPoint

Rileva una stringa lineare e restituisce il relativo primo punto. Per ulteriori informazioni consultare "ST_StartPoint" a pagina 271.

ST_EndPoint

Rileva una stringa lineare e restituisce il relativo ultimo punto. Per ulteriori informazioni consultare "ST_EndPoint" a pagina 211.

ST_PointN

Rileva una stringa lineare e un indice nel *nesimo* punto e restituisce quel punto. Per ulteriori informazioni consultare "ST_PointN" a pagina 262.

ST_Length

Rileva una stringa lineare e restituisce la relativa lunghezza come numero di precisione doppia. Per ulteriori informazioni consultare "ST_Length" a pagina 240.

ST_NumPoints

Rileva una stringa lineare e restituisce il numero di punti nella relativa sequenza come valore intero. Per ulteriori informazioni consultare "ST_NumPoints" a pagina 254.

ST_IsRing

Rileva una stringa lineare e restituisce 1 (TRUE) se la stringa lineare è un anello e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare "ST_IsRing" a pagina 236.

ST_IsClosed

Rileva una stringa lineare e restituisce 1 (TRUE) se la stringa lineare è chiusa e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare "ST_IsClosed" a pagina 232.

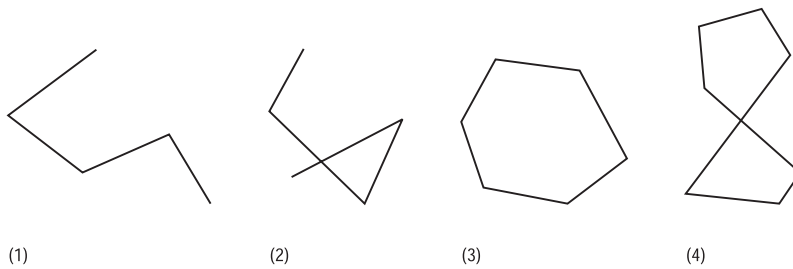


Figura 10. Oggetti della stringa lineare.

1. Una stringa lineare semplice non chiusa.
2. Una stringa lineare non semplice non chiusa.
3. Una stringa lineare semplice chiusa e quindi un anello.
4. Una stringa lineare non semplice chiusa. Non è un anello.

Poligoni

Un poligono è una superficie bidimensionale memorizzata come sequenza di punti che definiscono il relativo anello di limite esterno e 0 o più anelli interni. Non è possibile eseguire la sovrapposizione di un anello di poligono. Quindi, per definizione, i poligoni sono sempre semplici. La maggior parte delle volte i poligoni definiscono porzioni di terra, bacini idrici e altri elementi con estensione spaziale.

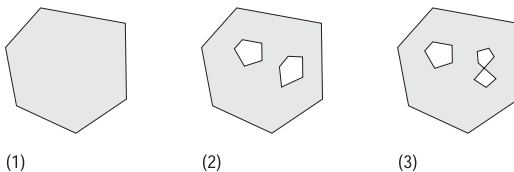


Figura 11. Poligoni.

1. Un poligono il cui limite è definito da un anello esterno.
2. Un poligono il cui limite è definito da un anello esterno e due anelli interni. L'area interna degli anelli interni fa parte dell'esterno dei poligoni.
3. Un poligono legale in quanto gli anelli si intersecano in un singolo punto tangente.

L'esterno e gli anelli interni definiscono il limite di un poligono e l'area racchiusa tra gli anelli definisce l'interno del poligono. Gli anelli di un poligono possono intersecarsi in un punto tangente ma mai incrociarsi. Oltre alle altre proprietà ereditate dalla geometria della superclasse, i poligoni presentano un'area.

Le funzioni che vengono utilizzate con i poligoni:

ST_Area

Rileva un poligono e restituisce la relativa area come numero di precisione doppia. Per ulteriori informazioni consultare "ST_Area" a pagina 188.

ST_ExteriorRing

Rileva un poligono e restituisce il relativo anello esterno come stringa lineare. Per ulteriori informazioni consultare "ST_ExteriorRing" a pagina 215.

ST_NumInteriorRing

Rileva un poligono e restituisce il numero di anelli interni in esso contenuti. Per ulteriori informazioni consultare "ST_NumInteriorRing" a pagina 253.

ST_InteriorRingN

Rileva un poligono e un indice e restituisce il *n*esimo anello interno come stringa lineare. Per ulteriori informazioni consultare "ST_InteriorRingN" a pagina 224.

ST_Centroid

Rileva un poligono e restituisce un punto che rappresenta il centro dell'estensione del poligono. Per ulteriori informazioni consultare "ST_Centroid" a pagina 196.

ST_PointOnSurface

Rileva un poligono e restituisce un punto che si trova nella superficie del poligono. Per ulteriori informazioni consultare "ST_PointOnSurface" a pagina 263.

ST_Perimeter

Rileva un poligono e restituisce il perimetro della relativa superficie. Per ulteriori informazioni consultare "ST_Perimeter" a pagina 258.

Multipunti

Un multipunto è una raccolta di punti e, come i relativi elementi, presenta una dimensione 0. Un multipunto è semplice se nessuno dei relativi elementi occupa la stessa area di coordinate. Il limite di un multipunto è NULL. I multipunti possono essere utilizzati per definire fenomeni quali la distribuzione di trasmissione aerea e l'incidenza di un'esplosione epidemica.

Le funzioni che vengono utilizzate con i multipunti:

ST_NumGeometries

Rileva una raccolta omogenea e restituisce il numero degli elementi della geometria di base in essa contenuti. Per ulteriori informazioni consultare "ST_NumGeometries" a pagina 252.

ST_GeometryN

Rileva una raccolta omogenea e un indice e restituisce la Nesima geometria di base. Per ulteriori informazioni consultare "ST_GeometryN" a pagina 221.

Multistringhe lineari

Una multistringa lineare è una raccolta di stringhe lineari. Le multistringhe lineari sono semplici se si intersecano solo nei punti finali degli elementi della stringa lineare. Le multistringhe lineari non sono semplici se gli interni degli elementi della stringa lineare si intersecano.

Il limite di una multistringa lineare è il punto finale non di intersezione degli elementi della stringa lineare. La multistringa lineare è chiusa se tutti gli elementi delle relative stringhe lineari sono chiusi. Il limite di una multistringa lineare è NULL se tutti i punti finali di tutti gli elementi si intersecano. Oltre alle altre proprietà ereditate dalla geometria della superclasse, le multistringhe lineari presentano una lunghezza. Le multistringhe lineari vengono utilizzate per definire flussi o reti stradali.

Le funzioni che vengono utilizzate con le multistringhe lineari:

ST_Length

Rileva una multistringa lineare e restituisce la lunghezza complessiva di tutti gli elementi delle stringhe lineari come numero di precisione doppia. Per ulteriori informazioni consultare "ST_Length" a pagina 240.

ST_IsClosed

Rileva una multistringa lineare e restituisce 1 (TRUE) se la multistringa lineare è chiusa e 0 (FALSE) in caso contrario. Per ulteriori informazioni consultare "ST_IsClosed" a pagina 232.

ST_NumGeometries

Rileva una raccolta omogenea e restituisce il numeri degli elementi della geometria di base in essa contenuti. Per ulteriori informazioni consultare "ST_NumGeometries" a pagina 252.

ST_GeometryN

Rileva una raccolta omogenea e un indice e restituisce la Nesima geometria di base. Per ulteriori informazioni consultare "ST_GeometryN" a pagina 221.

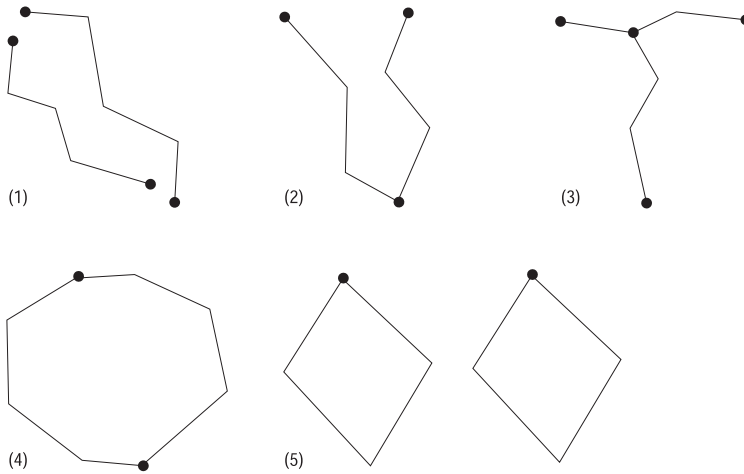


Figura 12. Multistringhe lineari.

1. Una multistringa lineare semplice il cui limite è definito dai quattro punti finali dei relativi due elementi della stringa lineare.
2. Una multistringa lineare semplice in quanto solo i punti finali degli elementi delle stringhe lineari si intersecano. Il limite è definito dai due punti finali che non presentano intersezioni.
3. Una stringa lineare non semplice in quanto l'interno di uno dei relativi elementi delle stringhe lineari presentano intersezioni. Il limite di questa multistringa lineare è definita dai quattro punti finali, compreso il punto di intersezione.
4. Una multistringa lineare non chiusa semplice. Non è chiusa in quanto le relative stringhe lineari non sono chiuse. E' semplice in quanto nessun interno delle stringhe lineari presentano intersezioni.
5. Una multistringa lineare chiusa semplice. E' chiusa in quanto tutti i relativi elementi sono chiusi. E' semplice in quanto nessun elemento presenta intersezioni nei relativi interni.

Multipoligoni

Il limite di un multipoligono è la lunghezza complessiva dei relativi esterni dell'elemento e degli anelli interni. L'interno di un multipoligono è definito come interno complessivo dei relativi poligoni dell'elemento. Il limite degli elementi del multipoligono possono intersecarsi solo in un punto tangente. Oltre alle altre proprietà ereditate dalla geometria della superclasse, i multipoligoni presentano un'area. I multipoligoni definiscono elementi quali una foresta o una porzione di terra non continua come una catena di isole.

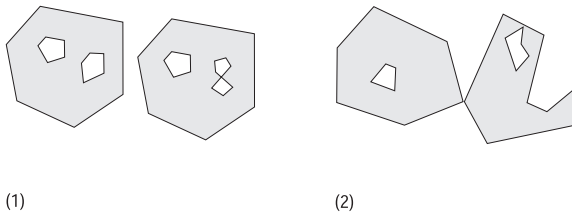


Figura 13. Multipoligoni.

1. Un multipoligono con due elementi poligono. Il limite è definito da due anelli esterni e tre anelli interni.
2. Un multipoligono con due elementi poligono. Il limite è definito da due anelli esterni e due anelli interni. I due elementi poligono si intersecano in un punto tangente.

Le funzioni che vengono utilizzate con i multipoligono:

ST_Area

Rileva un multipoligono e restituisce l'area complessiva di tutti gli elementi poligono come numero di precisione doppia. Per ulteriori informazioni consultare "ST_Area" a pagina 188.

ST_Centroid

Rileva un multipoligono e restituisce un punto che rappresenta il centro ponderato geometrico. Per ulteriori informazioni consultare "ST_Centroid" a pagina 196.

ST_NumGeometries

Rileva una raccolta omogenea e restituisce il numero degli elementi della geometria di base in essa contenuti. Per ulteriori informazioni consultare "ST_NumGeometries" a pagina 252.

ST_GeometryN

Rileva una raccolta omogenea e un indice e restituisce la *Nesima* geometria di base. Per ulteriori informazioni consultare "ST_GeometryN" a pagina 221.

Le funzioni che mostrano le relazioni e i confronti, generano geometrie e convertono i formati dei valori

Le sezioni precedenti descrivono tre categorie di funzioni spaziali:

- Le funzioni associate alle proprietà delle geometrie
- Le funzioni associate alle geometrie specifiche

Questa sezione descrive altre tre categorie:

- Le funzioni che determinano in modi in cui gli elementi geografici eseguono correlazioni e confronti
- Le funzioni che generano nuove geometrie

- Le funzioni che convertono i valori di una geometria in un formato che è possibile importare o esportare

Le funzioni che mostrano le relazioni e le similitudini tra elementi geografici

Varie funzioni spaziali restituiscono informazioni sui modi in cui gli elementi geografici vengono correlati e confrontati tra loro. La maggior parte di queste funzioni, denominate *predicati*, sono funzioni booleane. Questa sezione fornisce informazioni generali sui predicati e informazioni specifiche relative a ciascuna funzione.

Funzioni predicato

Le funzioni predicato restituisce 1 (TRUE) se un confronto soddisfa i criteri della funzione oppure 0 (FALSE) in caso contrario. I predicati che controllano una relazione spaziale confrontano coppie di geometrie di tipo o dimensione differente.

I predicati confrontano le coordinate X e Y delle geometrie interessate. Le coordinate Z e le misure (se esistenti) vengono ignorate. Ciò consente il confronto di geometrie con coordinate Z o misure con geometrie prive di esse.

Il modello *DE-9IM* (*Dimensionally Extended 9 Intersection Model*)¹ è un approccio matematico che definisce la relazione spaziale tra geometrie di tipi e dimensioni differenti. Questo modello descrive le relazioni spaziali tra tutti i tipi di geometrie come intersezioni dei relativi interni, limiti ed esterni considerando la dimensione delle intersezioni derivanti.

Nelle geometrie a e b : $I(a)$, $B(a)$ ed $E(a)$ rappresentano l'interno, il limite e l'esterno di a . $I(b)$, $B(b)$ ed $E(b)$ rappresentano l'interno, il limite e l'esterno di b . Le intersezioni di $I(a)$, $B(a)$ ed $E(a)$ con $I(b)$, $B(b)$ ed $E(b)$ producono una matrice 3 per 3. Ogni intersezione può creare geometrie di dimensioni differenti. Ad esempio, l'intersezione dei limiti di due poligoni consiste di un punto e una stringa lineare, in tal caso la funzione *dim* dovrebbe restituire come dimensione massima 1.

La funzione *dim* restituisce un valore 1, 0, 1 o 2. Il valore 1 corrisponde alla serie *null* o *dim(null)*, che viene restituito quando non viene rilevata alcuna intersezione.

1. Il modello DE-9IM è stato sviluppato da Clementini e Felice, per estendere il modello 9 Intersection Model di Egenhofer e Herring. DE-9IM scaturisce dalla collaborazione di quattro autori, Clementini, Eliseo, Di Felice e van Osstroom. Tale modello è stato pubblicato nel manuale "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. Il modello 9 Intersection Model di Springer-Verlag Singapore (1993) Egenhofer M.J. e Herring, J., è stato pubblicato nel manuale "Categorizing binary topological relationships between regions, lines, and points in geographic databases," *Tech. Report, Department of Surveying Engineering*, University of Maine, Orono, ME 1991.

	Interno	Limite	Esterno
Interno	$\dim(I(a) \cap I(b))$	$\dim(I(a) \cap B(b))$	$\dim(I(a) \cap E(b))$
Limite	$\dim(B(a) \cap I(b))$	$\dim(B(a) \cap B(b))$	$\dim(B(a) \cap E(b))$
Esterno	$\dim(E(a) \cap I(b))$	$\dim(E(a) \cap B(b))$	$\dim(E(a) \cap E(b))$

Il risultato dei predicati di relazione spaziale possono essere interpretati o verificati confrontando i risultati del predicato con una matrice modello che rappresenta i valori accettabili per DE-9IM.

La matrice modello contiene i valori accettabili per ciascuna cella della matrice di intersezione. I valori modello possibili sono:

- T** E' necessaria la presenza di un'intersezione, $\dim = 0, 1$ o 2 .
- F** E' necessario che l'intersezione non esista, $\dim = -1$.
- *** La presenza di un'intersezione non è rilevante, $\dim = -1, 0, 1$ o 2 .
- 0** E' necessaria la presenza di un'intersezione e la relativa dimensione massima deve essere 0 , $\dim = 0$.
- 1** E' necessaria la presenza di un'intersezione e la relativa dimensione massima deve essere 1 , $\dim = 1$.
- 2** E' necessaria la presenza di un'intersezione e la relativa dimensione massima deve essere 2 , $\dim = 2$.

Ad esempio, la seguente matrice modello per il predicato ST_Within include i valori T, F e *.

Tabella 40. Matrice per ST_Within. La matrice modello del predicato ST_Within per le combinazioni di geometrie.

		b		
		Interno	Limite	Esterno
a	Interno	P	*	F
	Limite	*	*	F
	Esterno	*	*	*

Il predicato ST_Within restituisce TRUE quando gli interni di entrambe le geometrie si intersecano e quando l'interno e il limite di *a* non intersecano l'esterno di *b*. Tutte le altre situazioni non sono rilevanti.

Ciascun predicato presenta almeno una matrice modello, ma alcuni richiedono più di una matrice per descrivere le relazioni di varie combinazioni di tipi di geometria.

ST_Equals

ST_Equals restituisce 1 (TRUE) se le due geometrie dello stesso tipo presentano valori di coordinate X,Y identici.






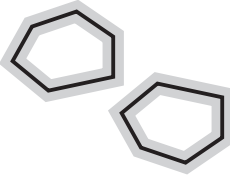
	
punto / punto	multipunto / multipunto
	
stringa lineare /stringa lineare	multistringa /multistringa
	
poligono / poligono	multipoligono / multipoligono

Figura 14. ST_Equals. Le geometrie sono uguali se presentano coordinate X,Y corrispondenti.

Tabella 41. Matrice per l'uguaglianza. La matrice modello DE-9IM per l'uguaglianza garantisce l'intersezione degli interni e che nessun interno o limite di entrambe le geometrie si intersecano con l'esterno dell'altra.

		b		
		Interno	Limite	Esterno
a	Interno	P	*	F
	Limite	*	*	F
	Esterno	F	F	*

Per ulteriori informazioni consultare "ST_Equals" a pagina 214.

ST_OrderingEquals

ST_OrderingEquals confronta le due geometrie e restituisce 1 (TRUE) se le geometrie sono uguali e le coordinate sono nello stesso ordine; in caso contrario restituisce 0 (FALSE). Per ulteriori informazioni consultare "ST_OrderingEquals" a pagina 255.

ST_Disjoint

ST_Disjoint restituisce 1 (TRUE) se l'intersezione delle due geometrie è una serie vuota.

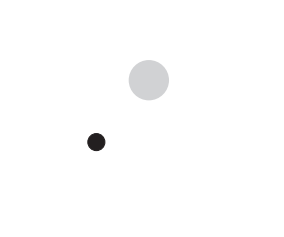
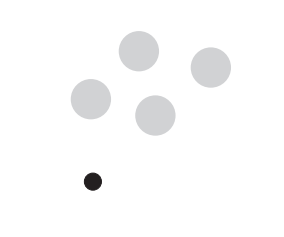
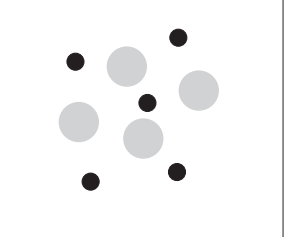



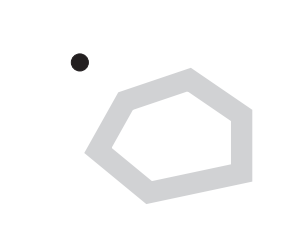

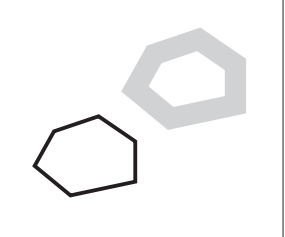
		
punto / punto	punto / multipunto	multipunto / multipunto
		
punto / stringa lineare	multistringa / stringa lineare	poligono / stringa lineare
		
punto / poligono	multipunto / multipoligono	poligono / poligono

Figura 15. ST_Disjoint. Le geometrie sono separate se non si intersecano in alcun modo.

Tabella 42. Matrice per *ST_Disjoint*. La matrice modello del predicato *ST_Disjoint* specifica che gli interni e i limiti di entrambe le geometrie non si intersecano.

		b		
		Interno	Limite	Esterno
a	Interno	F	F	*
	Limite	F	F	*
	Esterno	*	*	*

Per ulteriori informazioni consultare “*ST_Disjoint*” a pagina 208.

ST_Intersects

ST_Intersects restituisce 1 (TRUE) se l’intersezione non risulta in una serie vuota. L’intersezione restituisce l’opposto esatto di *ST_Disjoint*.

Il predicato *ST_Intersects* restituisce TRUE se le condizioni delle seguenti matrici modello restituisce TRUE.

Tabella 43. Matrice per *ST_Intersects* (1). Il predicato *ST_Intersects* restituisce TRUE se gli interni di entrambe le geometrie si intersecano.

		b		
		Interno	Limite	Esterno
a	Interno	P	*	*
	Limite	*	*	*
	Esterno	*	*	*

Tabella 44. Matrice per *ST_Intersects* (2). Il predicato *ST_Intersects* restituisce TRUE se il limite della prima geometria si interseca con il limite della seconda geometria.

		b		
		Interno	Limite	Esterno
a	Interno	*	P	*
	Limite	*	*	*
	Esterno	*	*	*

Tabella 45. Matrice per *ST_Intersects* (3). Il predicato *ST_Intersects* restituisce TRUE se il limite della prima geometria si interseca con l’interno della seconda geometria.

		b		
		Interno	Limite	Esterno
a	Interno	*	*	*
	Limite	P	*	*
	Esterno	*	*	*

Tabella 46. Matrice per *ST_Intersects* (4). Il predicato *ST_Intersects* restituisce TRUE se i limiti di entrambe le geometrie si intersecano.

		b		
		Interno	Limite	Esterno
a	Interno	*	*	*
	Limite	*	P	*
	Esterno	*	*	*

Per ulteriori informazioni consultare “*ST_Intersects*” a pagina 231.

EnvelopesIntersect

Questa funzione restituisce 1 (TRUE) se gli involuipi delle due geometrie si intersecano. E' una funzione di convenienza che implementa in modo efficiente *ST_Intersects* (*ST_Envelope*(g1), *ST_Envelope*(g2)). Per ulteriori informazioni consultare “*EnvelopesIntersect*” a pagina 168.

ST_Touches

ST_Touches restituisce 1 (TRUE) se nessuno dei punti comuni delle geometrie intersecano gli interni di entrambe le geometrie. Almeno una geometria deve essere una stringa lineare, un poligono, una multistringa lineare o un multipoligono.


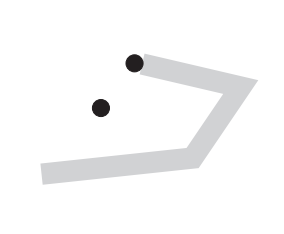

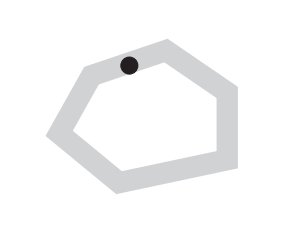
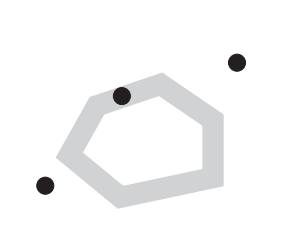
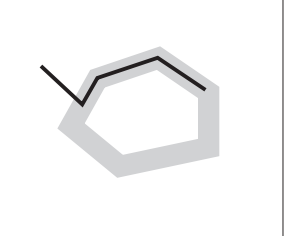
		
punto / stringa lineare	multipunto / stringa lineare	stringa lineare / stringa lineare
		
punto / poligono	multipunto / poligono	stringa lineare / poligono

Figura 16. *ST_Touches*

Le matrici modello illustrano che il predicato *ST_Touches* restituisce TRUE quando gli interni della geometria non si intersecano e il limite delle

geometrie si intersecano con gli interni e i relativi limiti delle altre geometrie.

Tabella 47. Matrice per *ST_Touches* (1)

		b		
		Interno	Limite	Esterno
a	Interno	F	P	*
	Limite	*	*	*
	Esterno	*	*	*

Tabella 48. Matrice per *ST_Touches* (2)

		b		
		Interno	Limite	Esterno
a	Interno	F	*	*
	Limite	P	*	*
	Esterno	*	*	*

Tabella 49. Matrice per *ST_Touches* (3)

		b		
		Interno	Limite	Esterno
a	Interno	F	*	*
	Limite	*	P	*
	Esterno	*	*	*

Per ulteriori informazioni consultare “*ST_Touches*” a pagina 274.

ST_Overlaps

ST_Overlaps confronta le due geometrie della stessa dimensione. Tale funzione restituisce 1 (TRUE) se la relativa serie di intersezioni crea una geometria differente da entrambe ma con la stessa dimensione.

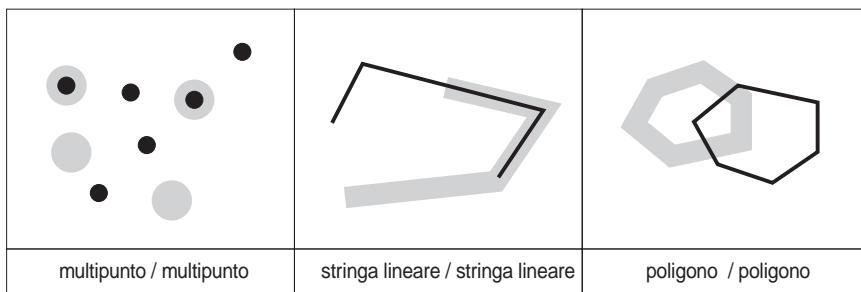


Figura 17. *ST_Overlaps*

La matrice modello nella Tabella 50 a pagina 150 fa riferimento alle sovrapposizioni poligono/poligono, multipunto/multipunto e multipoligono/multipoligono. Per queste combinazioni il predicato di

sovrapposizione restituisce TRUE se l'interno di entrambe le geometrie si interseca con gli altri interni ed esterni.

Tabella 50. Matrice per ST_Overlaps (1)

		b		
		Interno	Limite	Esterno
a	Interno	P	*	P
	Limite	*	*	*
	Esterno	P	*	*

La matrice modello nella Tabella 51 fa riferimento alle sovrapposizioni stringa lineare/stringa lineare e multistringa lineare/multistringa lineare. In questo caso l'intersezione delle geometrie deve creare una geometria con dimensione 1 (un'altra stringa lineare). Se la dimensione dell'intersezione degli interni è 1, il predicato ST_Overlaps dovrebbe restituire FALSE e il predicato ST_Crosses TRUE.

Tabella 51. Matrice per ST_Overlaps (2)

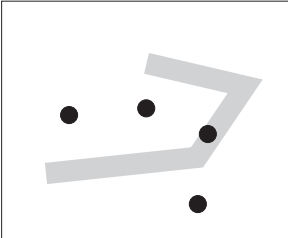

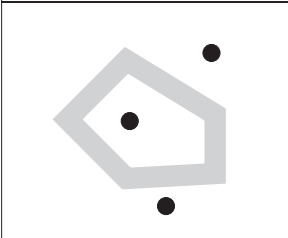

		b		
		Interno	Limite	Esterno
a	Interno	1	*	P
	Limite	*	*	*
	Esterno	P	*	*

Per ulteriori informazioni consultare "ST_Overlaps" a pagina 256.

ST_Crosses

ST_Crosses restituisce 1 (TRUE) se l'intersezione crea una geometria con dimensione inferiore alla dimensione massima delle due geometrie di origine e la serie di intersezioni è interna a entrambe le geometrie di origine.

ST_Crosses restituisce 1 (TRUE) solo per i confronti multipunto/poligono, multipunto/stringa lineare, stringa lineare/stringa lineare, stringa lineare/poligono e stringa lineare/multipoligono.

	
multipunto / stringa lineare	Stringa lineare / Stringa lineare
	
multipunto / poligono	stringa lineare / poligono

La matrice modello nella Tabella 52 fa riferimento alle sovrapposizioni multipunto/stringa lineare, multipunto/multistringa lineare, multipunto/poligono, multipunto/multipoligono, stringa lineare/poligono, stringa lineare/multipoligono. La matrice specifica che gli interni devono intersecarsi e che l'interno della geometria principale (geometria *a*) devono intersecarsi con l'esterno della geometria secondaria (geometria *b*).

Tabella 52. Matrice per *ST_Crosses* (1)

		b		
		Interno	Limite	Esterno
a	Interno	P	*	P
	Limite	*	*	*
	Esterno	*	*	*

La matrice modello nella Tabella 53 a pagina 152 fa riferimento alle sovrapposizioni stringa lineare/stringa lineare, stringa lineare/multistringa lineare e multistringa lineare/multistringa lineare. La matrice specifica come dimensione di intersezione degli interni 0 (intersezione con un punto). Se la dimensione di questa intersezione è 1 (intersezione con una stringa lineare), il predicato *ST_Crosses* restituisce FALSE; e il predicato *ST_Overlaps* restituisce TRUE.

Tabella 53. Matrice per ST_Crosses (2)

		b		
		Interno	Limite	Esterno
a	Interno	0	*	*
	Limite	*	*	*
	Esterno	*	*	*

Per ulteriori informazioni consultare “ST_Crosses” a pagina 203.

ST_Within

ST_Within restituisce 1 (TRUE) se la prima geometria rientra completamente nella seconda geometria. ST_Within restituisce l’opposto esatto di ST_Contains.

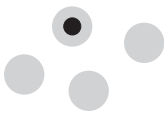
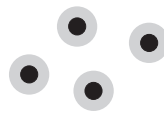


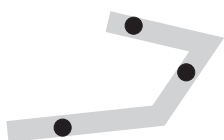




		
punto / multipunto	multipunto / multipunto	multipunto / poligono
		
punto / stringa lineare	multipunto / stringa lineare	stringa lineare / stringa lineare
		
punto / poligono	stringa lineare / poligono	poligono / poligono

Figura 18. Within

La matrice modello del predicato `ST_Within` specifica che gli interni di entrambe le geometrie devono intersecarsi mentre l'interno e il limite della geometria principale (geometria *a*) non deve intersecarsi con l'esterno della geometria secondaria (geometria *b*).

Tabella 54. Matrice per `ST_Within`

		b		
		Interno	Limite	Esterno
a	Interno	P	*	F
	Limite	*	*	F
	Esterno	*	*	*

Per ulteriori informazioni consultare “`ST_Within`” a pagina 277.

ST_Contains

`ST_Contains` restituisce 1 (TRUE) se la seconda geometria rientra completamente nella prima geometria. Il predicato `ST_Contains` restituisce l'opposto esatto del predicato `ST_Within`.



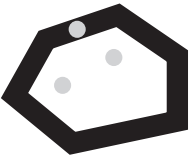






		
multipunto / punto	multipunto / multipunto	poligono / multipunto
		
stringa lineare / punto	stringa lineare / multipunto	stringa lineare / stringa lineare
		
poligono / punto	poligono / stringa lineare	poligono / poligono

Figura 19. ST_Contains

La matrice modello del predicato ST_Contains specifica che gli interni di entrambe le geometrie devono intersecarsi e l'interno e il limite della geometria secondaria (geometria *b*) non devono intersecarsi con l'esterno della geometria principale (geometria *a*).

Tabella 55. Matrice per ST_Contains

		b		
		Interno	Limite	Esterno
a	Interno	P	*	*
	Limite	*	*	*
	Esterno	F	F	*

Per ulteriori informazioni consultare "ST_Contains" a pagina 197.

ST_Relate

La funzione `ST_Relate` confronta le due geometrie e restituisce 1 (TRUE) se le geometrie soddisfano le condizioni specificate dalla stringa della matrice modello DE-91M; in caso contrario, la funzione restituisce 0 (FALSE). Per ulteriori informazioni consultare “`ST_Relate`” a pagina 268.

ST_Distance

La funzione `ST_Distance` rileva la distanza minima che separa due elementi separati. Se gli elementi non sono separati, la funzione rileverà una distanza minima uguale a 0.

Ad esempio, `ST_Distance` può rilevare la distanza minore che un aeroplano può coprire per unire due ubicazioni. La Figura 20 illustra tali informazioni.



Figura 20. Distanza minima tra due città. `ST_Distance` può rilevare le coordinate per le ubicazioni di Los Angeles e Chicago come input e restituire un valore che indica la distanza minima tra le due ubicazioni.

Per ulteriori informazioni consultare “`ST_Distance`” a pagina 210.

Le funzioni che generano nuove geometrie da quelle esistenti

DB2 Spatial Extender fornisce predicati e funzioni di trasformazione che generano nuove geometrie da quelle esistenti.

ST_Intersection

La funzione `ST_Intersection` restituisce la serie di intersezioni di due geometrie. La serie di intersezioni viene sempre restituita come raccolta con la dimensione minima delle geometrie di origine. Ad esempio, per una stringa lineare che si interseca con un poligono, la funzione di intersezione restituisce una multistringa lineare comprensiva della porzione di stringa lineare comune nell'interno e sul limite del poligono. La multistringa lineare contiene più di una stringa lineare se la stringa lineare di origine si interseca con il poligono con due o più segmenti discontinui. Se le geometrie non si intersecano o se

l'intersezione crea una geometria di dimensione inferiore alle geometrie di origine, viene restituita una geometria vuota.

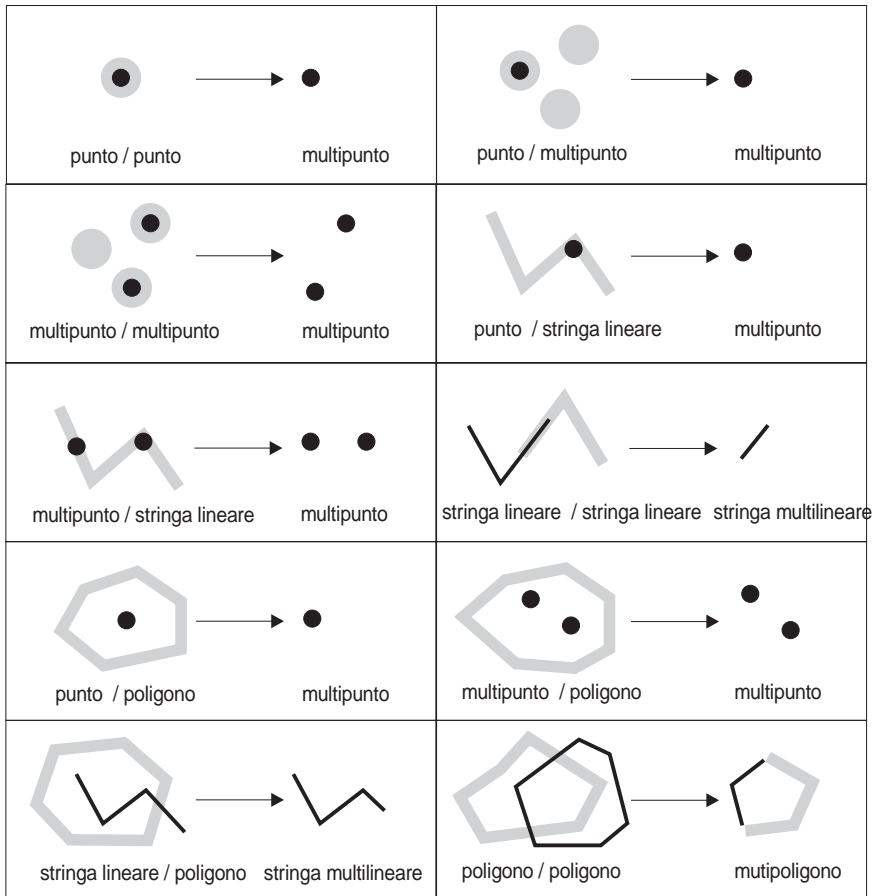


Figura 21. *ST_Intersection*. Esempi di funzione *ST_Intersection*.

Per ulteriori informazioni consultare “*ST_Intersection*” a pagina 229.

ST_Difference

La funzione *ST_Difference* restituisce la porzione della geometria principale che non presenta intersezioni con la geometria secondaria. Questa funzione rappresenta l'operatore AND NOT logico dell'area. La funzione *ST_Difference* viene utilizzata solo con le geometrie con dimensioni uguali e restituisce una raccolta con dimensione corrispondente alle geometrie di origine. Nel caso in cui le geometrie di origine sono uguali, viene restituita una geometria vuota.

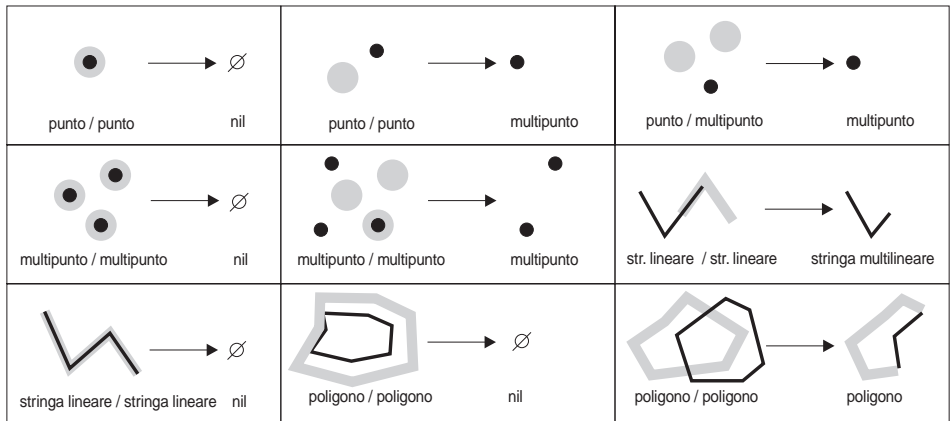


Figura 22. ST_Difference

Per ulteriori informazioni consultare “ST_Difference” a pagina 205.

ST_Union

La funzione ST_Union restituisce la serie di unioni di due geometrie. Tale funzione rappresenta l’operatore OR logico dell’area. Le geometrie di origine devono essere di dimensione uguale. ST_Union restituisce sempre il risultato come raccolta.

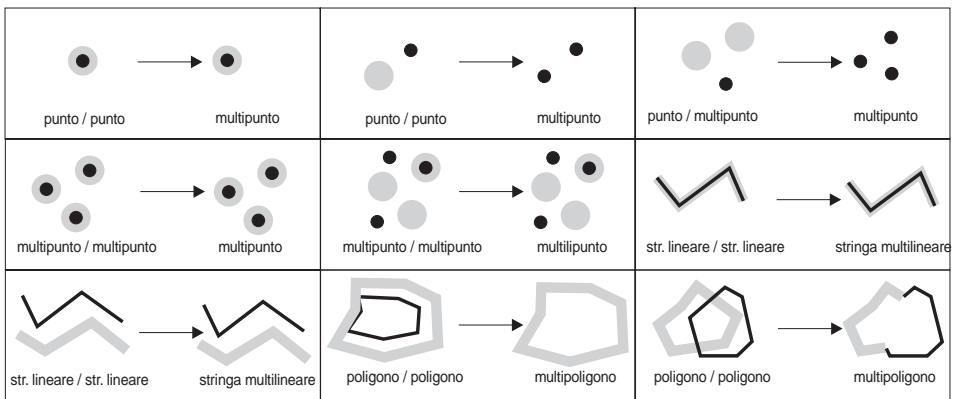


Figura 23. ST_Union

Per ulteriori informazioni consultare “ST_Union” a pagina 276.

ST_Buffer

La funzione ST_Buffer genera una geometria racchiudendo più volte una geometria a una distanza specifica. Un poligono deriva dalla creazione di un buffer di una geometria principale o dagli elementi di una raccolta chiusi in modo che tutti i poligoni del buffer si sovrappongono. Tuttavia, quando c’è

una distanza sufficiente tra gli elementi di una raccolta del buffer, si rileveranno i poligoni del buffer individuale, in tal caso la funzione ST_Buffer restituisce un multipoligono.

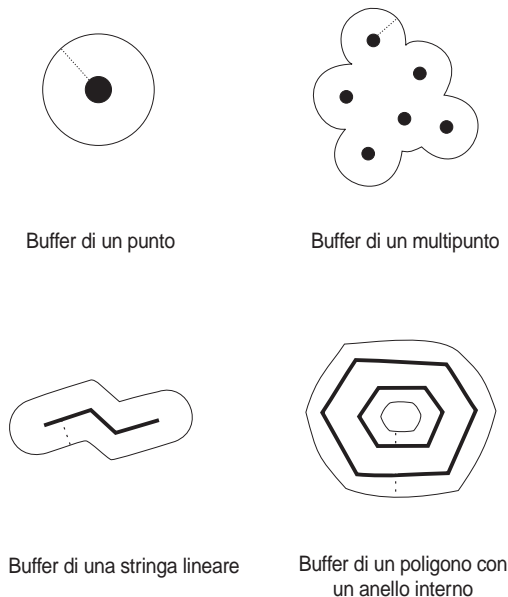


Figura 24. ST_Buffer

La funzione ST_Buffer accetta la distanza positiva e quella negativa, tuttavia, solo le geometrie con una dimensione 2 (poligoni e multipoligoni) sono valide per un buffer negativo. Il valore assoluto della distanza di buffer viene utilizzato ogni volta la dimensione della geometria di origine è inferiore a 2 (tutte le geometrie escluso il poligono e il multipoligono).

In generale, per gli anelli esterni, la distanza di buffer positivo genera anelli di poligono lontani dal centro della geometria di origine; le distanze di buffer negative generano anelli di poligono o multipoligono in prossimità del centro. Per gli anelli interni di un poligono o multipoligono, una distanza di buffer positiva genera un anello di buffer in prossimità del centro e una distanza di buffer negativa genera un anello di buffer lontano dal centro.

Il processo di creazione buffer unisce i poligoni che si sovrappongono. Le distanze negative superiori alla metà della larghezza massima dell'interno di un poligono creano una geometria vuota.

Per ulteriori informazioni consultare "ST_Buffer" a pagina 194.

LocateAlong

Per le geometrie con misure, è possibile rilevare l'ubicazione di una particolare misura con la funzione `LocateAlong`. `LocateAlong` restituisce l'ubicazione come multipunto. Se la dimensione della geometria di origine è 0 (ad esempio, un punto e un multipunto), è necessaria una corrispondenza esatta e i punti con un valore di misura corrispondente vengono restituiti come multipunto. Tuttavia, per le geometrie di origine la cui dimensione è superiore a 0, viene eseguita l'interpolazione dell'ubicazione. Ad esempio, se il valore della misura immesso è 5.5 e le misure dei vertici di una stringa lineare sono rispettivamente 3, 4, 5, 6 e 7, viene restituito il punto di interpolazione che si trova esattamente al centro tra i vertici con valori di misura 5 e 6.

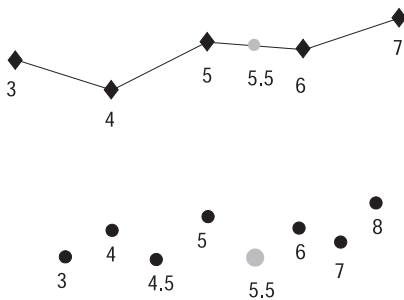


Figura 25. `LocateAlong`

Per ulteriori informazioni consultare “`LocateAlong`” a pagina 174.

LocateBetween

La funzione `LocateBetween` restituisce la serie di percorsi o di ubicazioni che rispecchiano la media dei due valori delle misure delle geometrie di origine. Se la dimensione della geometria di origine è 0, `LocateBetween` restituisce un multipunto contenente tutti i punti le cui misure rappresentano la media delle due misure di origine. Per le geometrie di origine la cui dimensione è superiore a 0, `LocateBetween` restituisce una multistringa lineare se è possibile eseguire l'interpolazione di un percorso; in caso contrario `LocateBetween` restituisce un multipunto contenente le ubicazioni dei punti. Un punto vuoto viene restituito ogni volta che `LocateBetween` non può eseguire l'interpolazione di un percorso o non può rilevare un'ubicazione tra le due misure. `LocateBetween` esegue una ricerca comprensiva delle geometrie; quindi le misure delle geometrie devono essere superiori o uguali alle misure *from* e inferiori o uguali a quelle *to*.

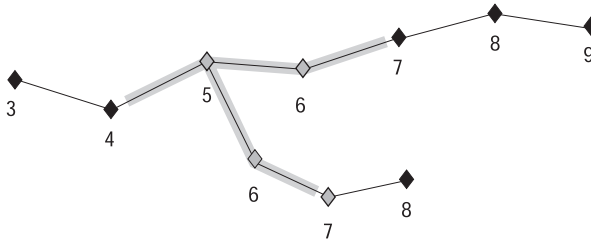


Figura 26. *LocateBetween*

Per ulteriori informazioni consultare “LocateBetween” a pagina 176.

ST_ConvexHull

La funzione `ST_ConvexHull` restituisce il poligono a involucro convesso di ciascuna geometria con almeno tre vertici che formano una figura convessa. Se i vertici della geometria non formano una figura convessa, `ST_ConvexHull` restituisce un valore nullo. `ST_ConvexHull` rappresenta la prima operazione nella rappresentazione a tasselli utilizzata per creare una rete TIN da una serie di punti.

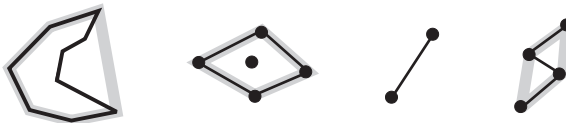


Figura 27. *ST_ConvexHull*

Per ulteriori informazioni consultare “`ST_ConvexHull`” a pagina 199.

ST_Polygon

Genera un poligono da una stringa lineare. Per ulteriori informazioni consultare “`ST_Polygon`” a pagina 267.

Le funzioni che convertono il formato dei valori di una geometria

DB2 Spatial Extender supporta tre formati di scambio dati GIS:

- Rappresentazione WKT
- Rappresentazione WKB
- Rappresentazione in formato binario ESRI

Rappresentazione WKT

DB2 Spatial Extender dispone di varie funzioni che generano geometrie dalle descrizioni di testo.

ST_WKTTToSQL

Crea una geometria dalla rappresentazione di un testo di qualsiasi

tipo di geometria. Non è necessario specificare alcun identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "ST_WKTToSQL" a pagina 280.

ST_GeomFromText

Crea una geometria dalla rappresentazione di un testo di qualsiasi tipo di geometria. E' necessario specificare un identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "ST_GeometryFromText" a pagina 217.

ST_PointFromText

Crea un punto dalla rappresentazione di testo di un punto. Per ulteriori informazioni consultare "ST_PointFromText" a pagina 259.

ST_LineFromText

Crea una stringa lineare dalla rappresentazione di testo di una stringa lineare. Per ulteriori informazioni consultare "ST_LineFromText" a pagina 242.

ST_PolyFromText

Crea un poligono dalla rappresentazione di testo di un poligono. Per ulteriori informazioni consultare "ST_PolyFromText" a pagina 264.

ST_MPointFromText

Crea un multipunto dalla rappresentazione di un multipunto. Per ulteriori informazioni consultare "ST_MPointFromText" a pagina 248.

ST_MLineFromText

Crea una multistringa lineare dalla rappresentazione di una multistringa lineare. Per ulteriori informazioni consultare "ST_MLineFromText" a pagina 245.

ST_MPolyFromText

Crea un multipoligono dalla rappresentazione di un multipoligono. Per ulteriori informazioni consultare "ST_MPolyFromText" a pagina 250.

La rappresentazione di testo è una stringa ASCII. Tale rappresentazione consente lo scambio della geometria in formato di testo ASCII. Queste funzioni non richiedono la definizione di alcuna struttura di programma speciale per associare una rappresentazione binaria. Quindi, è possibile utilizzare tali funzioni nel programma 3GL o 4GL.

La funzione ST_AsText converte il valore di una geometria esistente in una rappresentazione di testo. Per ulteriori informazioni consultare "ST_AsText" a pagina 191.

Per una descrizione dettagliata delle rappresentazioni WKT, consultare la sezione "Le rappresentazioni WKT OGC" a pagina 295.

Rappresentazione WKB

DB2 Spatial Extender dispone di varie funzioni che generano geometrie dalle rappresentazioni WKB (well-known binary).

ST_WKBToSQL

Crea una geometria da una rappresentazione WKB di qualsiasi tipo di geometria. Non è necessario specificare alcun identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "ST_WKBToSQL" a pagina 278.

ST_GeomFromWKB

Crea una geometria dalla rappresentazione WKB di qualsiasi tipo di geometria. E' necessario specificare un identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "ST_GeomFromWKB" a pagina 219.

ST_PointFromWKB

Crea un punto dalla rappresentazione WKB di un punto. Per ulteriori informazioni consultare "ST_PointFromWKB" a pagina 260.

ST_LineFromWKB

Crea una stringa lineare dalla rappresentazione WKB di una stringa lineare. Per ulteriori informazioni consultare "ST_LineFromWKB" a pagina 243.

ST_PolyFromWKB

Crea un poligono dalla rappresentazione WKB di un poligono. Per ulteriori informazioni consultare "ST_PolyFromWKB" a pagina 265.

ST_MPointFromWKB

Crea un multipunto dalla rappresentazione WKB di un multipunto. Per ulteriori informazioni consultare "ST_MPointFromWKB" a pagina 249.

ST_MLineFromWKB

Crea una multistringa lineare dalla rappresentazione WKB di una multistringa lineare. Per ulteriori informazioni consultare "ST_MLineFromWKB" a pagina 246.

ST_MPolyFromWKB

Crea un multipoligono dalla rappresentazione WKB di un multipoligono. Per ulteriori informazioni consultare "ST_MPolyFromWKB" a pagina 251.

La rappresentazione WKB rappresenta un flusso di byte contiguo. Tale rappresentazione consente lo scambio tra un client ODBC e un database SQL in formato binario. Queste funzioni della geometria richiedono che la definizione delle strutture C corrispondano alla rappresentazione binaria. Quindi sono state progettate per l'utilizzo in un programma 3GL e non sono adatte per un ambiente 4GL.

La funzione ST_AsBinary converte il valore di una geometria esistente in una rappresentazione WKB. Per ulteriori informazioni consultare "ST_AsBinary" a pagina 190.

Per una descrizione dettagliata delle rappresentazioni WKT, consultare la sezione "Le rappresentazioni WKB (well-known binary) OGC" a pagina 300.

Rappresentazione in formato ESRI

DB2 Spatial Extender dispone di varie funzioni che generano geometrie da una rappresentazione in formato ESRI. La rappresentazione in formato ESRI supporta le coordinate Z e le misure oltre alle rappresentazioni bidimensionali supportate dalle rappresentazione WKB e WKT.

ShapeToSQL

Crea una geometria da un formato di qualsiasi tipo di geometria. Non è necessario specificare alcun identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "ShapeToSQL" a pagina 186.

GeometryFromShape

Crea una geometria da un formato di qualsiasi tipo di geometria. E' necessario specificare un identificativo di sistema di riferimento spaziale. Per ulteriori informazioni consultare "GeometryFromShape" a pagina 167.

PointFromShape

Crea un punto dal formato punto. Per ulteriori informazioni consultare "PointFromShape" a pagina 183.

LineFromShape

Crea una stringa lineare dal formato multirighe. Per ulteriori informazioni consultare "LineFromShape" a pagina 172.

PolyFromShape

Crea un poligono da un formato multirighe. Per ulteriori informazioni consultare "PolyFromShape" a pagina 184.

MPointFromShape

Crea un multipunto da un formato multipunto. Per ulteriori informazioni consultare "MPointFromShape" a pagina 181.

MLineFromShape

Crea una multistringa lineare da un formato multirighe con più parti. Per ulteriori informazioni consultare "MLine FromShape" a pagina 179.

MPolyFromShape

Crea un multipoligono da un formato di un poligono con più parti. Per ulteriori informazioni consultare "MPolyFromShape" a pagina 182.

La sintassi generale di queste funzioni è uguale. Il primo argomento è la rappresentazione del formato immesso come tipo di dati BLOB. Il secondo argomento è l'identificativo di riferimento spaziale che verrà assegnato alla geometria. Ad esempio, la funzione `GeometryFromShape` presenta la seguente sintassi:

```
GeometryFromShape(shapegeometry, SRID)
```

Per associare la rappresentazione binaria, queste funzioni di formato richiedono la definizione delle strutture C. Quindi sono state progettate per l'utilizzo in un programma 3GL e non sono adatte per un ambiente 4GL.

La funzione `AsBinaryShape` converte il valore di una geometria in una rappresentazione del formato ESRI. Per ulteriori informazioni consultare "AsBinaryShape" a pagina 166.

Per una descrizione dettagliata delle rappresentazioni del formato, consultare la sezione "Le rappresentazioni in formato ESRI" a pagina 304.

Capitolo 14. Funzioni spaziali per le interrogazioni SQL

Questo capitolo elenca le funzioni disponibili che è possibile richiamare quando si interrogano i dati spaziali. Ogni funzione è descritta in una sezione che riporta la sintassi, il tipo di ritorno e gli esempi di codifica. Alcuni esempi in questo capitolo includono un'istruzione CREATE TABLE in cui le colonne vengono definite come colonne spaziali.

Le seguenti considerazioni si applicano alle funzioni spaziali:

- Gli esempi in questo capitolo vengono identificati con il nome libreria db2gse. Invece di identificare in modo esplicito ogni tipo e funzione spaziale con db2gse, è possibile impostare il percorso della funzione in modo da includere db2gse.
- Prima di inserire i dati in una colonna spaziale:
 - E' necessario aumentare il valore del parametro udf_mem_sz. L'impostazione iniziale consigliata è 2048. Se il valore 2048 risulta inadeguato, aumentare il valore del parametro udf_mem_sz con incrementi da 256.
 - Registrare la colonna spaziale come struttura. Per ulteriori informazioni sulla registrazione di una colonna spaziale come struttura, consultare la sezione "Capitolo 4. Definizione di colonne spaziali, registrazione come strutture e abilitazione di un geocoder per la gestione" a pagina 33.

AsBinaryShape

AsBinaryShape rileva un oggetto geometrico e restituisce BLOB.

Sintassi

```
db2gse.AsBinaryShape(g db2gse.ST_Geometry)
```

Tipo di ritorno

BLOB(1m)

Esempi

Il seguente frammento di codifica C illustra come la funzione AsBinaryShape converte i poligoni area della tabella SENSITIVE_AREAS in poligoni forma. Questi poligoni forma vengono trasferiti alla funzione draw_polygon dell'applicazione per la visualizzazione.

```
/* Create the SQL expression. */
strcpy(sqlstmt, "select db2gse.AsBinaryShape (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);
```

GeometryFromShape

GeometryFromShape rileva una forma e un'identità del sistema di riferimento spaziale per restituire un oggetto di geometria.

Sintassi

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

Il seguente frammento di codifica C contiene funzioni ODBC integrate alle funzioni SQL DB2 Spatial Extender che inseriscono i dati nella tabella LOTS.

La tabella LOTS è stata creata con due colonne: la colonna LOT_ID, che identifica in modo univoco ogni lotto, e la colonna LOT, che contiene la geometria di ciascun lotto.

```
CREATE TABLE LOTS ( lot_id integer,
                    lot      db2gse.ST_MultiPolygon);
```

La funzione GeometryFromShape converte le forme nella geometria DB2 Spatial Extender. L'intera istruzione INSERT viene copiata in shp_sql. L'istruzione INSERT contiene i contrassegni di parametro per accettare i dati LOT_ID e LOT in modo dinamico.

```
/* 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 (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);
```

EnvelopesIntersect

EnvelopesIntersect restituisce 1 (TRUE) se le curve di due figure geometriche si intersecano; altrimenti restituisce 0 (FALSE).

Sintassi

```
db2gse.EnvelopesIntersect(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La funzione `get_window` richiama le coordinate della finestra di visualizzazione dall'applicazione. Il parametro della finestra ha una struttura a forma di poligono contenente una stringa di coordinate che rappresentano il poligono di visualizzazione. La funzione `PolyFromShape` converte il formato della finestra di visualizzazione in un poligono DB2 Spatial Extender che viene utilizzato dalla funzione `EnvelopesIntersect` come curva di intersezione. Vengono restituiti tutti i poligoni `SENSITIVE_AREAS` che intersecano l'interno o i margini della finestra di visualizzazione. Ogni poligono viene richiamato dalla serie dei risultati e passato alla funzione `draw_polygon`.

```
/* 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.AsBinaryShape(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);
```

Is3d

Is3d rileva un oggetto di geometria e restituisce 1 (TRUE) se l'oggetto contiene le coordinate 3D; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.Is3d(g db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella THREEED_TEST, che contiene due colonne: la colonna GID di tipo integer e la colonna di geometria G1.

```
CREATE TABLE THREEED_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Le istruzioni INSERT inseriscono due punti nella tabella THREEED_TEST. Il primo punto non contiene le coordinate Z mentre il secondo le contiene.

```
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)))
```

La seguente istruzione SELECT elenca il contenuto della colonna GID con i risultati della funzione Is3d. La funzione restituisce 0 per la prima riga, che non contiene le coordinate Z, e 1 per la seconda riga che contiene le coordinate Z.

```
SELECT gid, db2gse.Is3d (g1) "Is it 3d?" FROM THREEED_TEST
```

Viene restituita la seguente serie di risultati.

gid	Is it 3d?
1	0
2	1

IsMeasured

IsMeasured rileva un oggetto di geometria e restituisce 1 (TRUE) se l'oggetto presenta misure; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.IsMeasured(g db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella MEASURE_TEST che contiene due colonne. La colonna GID identifica le righe in modo univoco e la colonna G1 memorizza le geometrie dei punti.

```
CREATE TABLE MEASURE_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono due record nella tabella MEASURE_TEST. Il primo record memorizza un punto che non presenta una misura. Il secondo record presenta una misura.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano la colonna GID con i risultati della funzione IsMeasured. La funzione IsMeasured restituisce 0 per la prima riga perché il punto non ha una misura. Restituisce 1 per la seconda riga perché il punto presenta misure.

```
SELECT gid, db2gse.IsMeasured (g1) "Has measures?" FROM MEASURE_TEST
```

gid	Has measures
1	0
2	1

LineFromShape

LineFromShape rileva una forma di tipo punto e un'identità del sistema di riferimento spaziale e restituisce una stringa lineare.

Sintassi

```
db2gse.Line FromShape(ShapeLineString Blob(1M), cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_LineString
```

Esempi

Il seguente frammento di codifica compila la tabella SEWERLINES con l'ID univoco, la classe della dimensione e la geometria di ogni rete fognaria.

L'istruzione CREATE TABLE crea la tabella SEWERLINES che contiene tre colonne. La prima colonna, SEWER_ID, identifica ciascuna rete fognaria in modo univoco. La seconda colonna, CLASS, di tipo integer identifica il tipo di rete fognaria, generalmente associata alla capacità. La terza colonna, SEWER, di tipo stringa lineare memorizza la geometria della rete fognaria.

```
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 rileva un oggetto di geometria e una misura e restituisce la serie di punti trovata nella misura come multipunto.

Sintassi

```
db2gse.LocateAlong(g db2gse.ST_Geometry, adistance Double)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella LOCATEALONG_TEST. LOCATEALONG_TEST ha due colonne: la colonna GID, che identifica ogni riga in modo univoco, e la colonna di geometria G1, che memorizza esempi di geometria.

```
CREATE TABLE LOCATEALONG_TEST (gid integer, g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono due righe. La prima è una multistringa lineare, le seconda è un multipunto.

```
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)))
```

Nella seguente istruzione SELECT e nella serie di risultati corrispondente, la funzione LocateAlong viene utilizzata per ricercare i punti le cui misure corrispondono a 6.5. La prima riga restituisce un multipunto contenente due punti. Invece, la seconda riga restituisce un punto vuoto. Per le funzioni lineari (la geometria con una dimensione maggiore di 0), LocateAlong può eseguire l'interpolazione del punto; tuttavia, per i multipunti, la misura di destinazione deve corrispondere esattamente.

```
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.

Nella seguente istruzione SELECT e nella serie di risultati corrispondente, la funzione LocateAlong restituisce multipunti per entrambe le righe. La misura di destinazione 7 corrisponde alle misure dei dati di origine della multistringa lineare e del multipunto.

```
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 rileva un oggetto di geometria e due ubicazioni delle misure e restituisce una geometria che rappresenta la serie di percorsi disconnessi tra le due ubicazioni delle misure.

Sintassi

```
db2gse.LocateBetween(g db2gse.ST_Geometry, a distance Double,  
another distance Double)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella LOCATEBETWEEN_TEST. LOCATEBETWEEN_TEST ha due colonne: la colonna GID, che identifica ogni riga in modo univoco, e la colonna della multistringa lineare G1, che memorizza esempi di geometria.

```
CREATE TABLE LOCATEBETWEEN_TEST (gid integer, g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono due righe nella tabella LOCATEBETWEEN_TEST. La prima riga è una multistringa lineare e la seconda è un multipunto.

```
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)))
```

La seguente istruzione SELECT e la corrispondente serie di risultati mostrano il modo in cui la funzione LocateBetween calcola le misure comprese tra 6.5 e 7.5 inclusi. La prima riga restituisce una multistringa lineare contenente due stringhe lineari. La seconda riga restituisce un multipunto perché i dati di origine erano multipunti. Se la dimensione dei dati di origine è 0 (punto o multipunto), è richiesta una corrispondenza esatta.

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.LocateBetween (g1,6.5,7.5))  
AS varchar(96)) "Geometry"  
FROM LOCATEBETWEEN_TEST
```

GID	Geometry
1	MULTILINESTRING M (27.01000000 19.38000000 6.50000000, 31.19000000 18.47000000 7.00000000,38.09000000 19.61000000 7.50000000),(27.40000000 22.1400

```
0000 6.50000000, 30.98000000 23.98000000 7.00000000,36.95000000 24.98000000 7.5  
00000000) 2 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000 23.9  
8000000 7.00000000)
```

2 record(s) selected.

M

M rileva un punto e restituisce una misura.

Sintassi

```
db2gse.M(p db2gse.ST_Point)
```

Tipo di ritorno

Double

Esempi

La seguente istruzione CREATE TABLE crea la tabella M_TEST. M_TEST ha due colonne: la colonna integer GID, che identifica ogni riga in modo univoco, e la colonna del punto PT1, che memorizza esempi di geometria.

```
CREATE TABLE M_TEST (gid integer, pt1 db2gse.ST_Point)
```

Le seguenti istruzioni INSERT inseriscono una riga che contiene un punto con misure e una riga che contiene un punto senza misure.

```
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)))
```

Nella seguente istruzione SELECT e nella serie di risultati corrispondente, la funzione M elenca i valori relativi alle misure dei punti. Poiché il primo punto non presenta misure, la funzione M restituisce NULL.

```
SELECT gid, db2gse.M (pt1) "The measure" FROM M_TEST
```

```
GID          The measure  
-----  
1              -  
2  +7.000000000000000E+000
```

2 record(s) selected.

MLine FromShape

MLine FromShape rileva una forma di tipo multistringa lineare e un'identità del sistema di riferimento spaziale e restituisce una multistringa lineare.

Sintassi

```
db2gse.MLineFromShape(ShapeMultiLineString Blob(1M), cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiLineString
```

Esempi

Il seguente frammento di codifica compila la tabella WATERWAYS utilizzando un ID univoco, un nome e una multistringa lineare.

La tabella WATERWAYS viene creata con le colonne ID e NAME che identificano ogni flusso o l'intero sistema fluviale memorizzato nella tabella. La colonna WATER è una multistringa lineare poiché questi sistemi fluviali sono spesso degli aggregati di altre stringhe lineari.

```
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 rileva una forma di tipo multipunto e un'identità del sistema di riferimento spaziale e restituisce un multipunto.

Sintassi

```
db2gse.MPointFromShape(ShapeMultiPoint (1M), srs db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiPoint
```

Esempi

Questo frammento di codifica compila una tabella SPECIES_SITINGS di tipo biologico.

La tabella SPECIES_SITINGS viene creata con tre colonne. Le colonne SPECIES e GENUS identificano le righe in modo univoco mentre il multipunto SITINGS memorizza le relative ubicazioni.

```
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 rileva una forma di tipo multipoligono e un'identità del sistema di riferimento spaziale e restituisce un multipoligono.

Sintassi

```
db2gse.MPolyFromShape(ShapeMultiPolygon Blob(1m), srs db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiPolygon
```

Esempi

Questo frammento di codifica compila la tabella LOTS.

La tabella LOTS memorizza `lot_id`, che identifica ogni LOT in modo univoco, e il multipoligono LOT che contiene la geometria lineare.

```
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 rileva una forma di tipo punto e un'identità del sistema di riferimento spaziale e restituisce un punto.

Sintassi

```
db2gse.PointFromShape(db2gse.ShapePoint blob(1M), srs db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

Questo frammento di programma compila la tabella HAZARDOUS_SITES.

I siti a rischio vengono memorizzati nella tabella HAZARDOUS_SITES creata con l'istruzione CREATE TABLE. La colonna LOCATION, definita come punto, memorizza un'ubicazione che rappresenta il centro geografico di ciascun sito a rischio.

```
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 rileva una forma di tipo poligono e un'identità del sistema di riferimento spaziale e restituisce un poligono.

Sintassi

db2gse.PolyFromShape (ShapePolygon Blob(1M), srs db2gse.coordref)

Tipo di ritorno

db2gse.ST_Polygon

Esempi

Questo frammento di programma compila la tabella SENSITIVE_AREAS. I punti interrogativi rappresentano i contrassegni dei parametri per i valori id, nome, dimensione, tipo e zona che verranno richiamati in fase di runtime.

La tabella SENSITIVE_AREAS contiene diverse colonne che descrivono le istituzioni interessate oltre alla colonna ZONE che memorizza la geometria del poligono dell'istituzione.

```
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 definisce un valore db2gse.ST_Geometry specificato dalla relativa rappresentazione WKB. Viene utilizzato automaticamente il valore SRID 0.

Sintassi

```
db2gse.ST_ShapeToSQL(ShapeGeometry blob(1M))
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

Il seguente frammento di codifica C contiene funzioni ODBC integrate alle funzioni SQL DB2 Spatial Extender che inseriscono i dati nella tabella LOTS. La tabella LOTS è stata creata con due colonne: la colonna lot_id, che identifica ciascun lotto in modo univoco, e la colonna del multipoligono LOT, che contiene la geometria di ogni lotto.

```
CREATE TABLE lots (lot_id integer,  
                   lot      db2gse.ST_MultiPolygon);
```

La funzione ShapeToSQL converte le forme nella geometria DB2 Spatial Extender. L'intera istruzione INSERT viene copiata in shp_sql. L'istruzione INSERT contiene i contrassegni di parametro per accettare i dati LOT_id e lot in modo dinamico.

```
/* 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 rileva un poligono o un multipoligono e restituisce la relativa area.

Sintassi

```
db2gse.ST_Area(s db2gse.ST_Surface)
```

Tipo di ritorno

Double

Esempi

Un ingegnere desidera consultare un elenco di aree edificabili. Per ottenere l'elenco, un tecnico GIS seleziona l'ID e l'area delle piantine di ciascun edificio.

Le piantine degli edifici vengono memorizzate nella tabella BUILDINGFOOTPRINTS creata con la seguente istruzione CREATE TABLE:

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,
                                   lot_id integer,
                                   footprint db2gse.ST_MultiPolygon);
```

Per soddisfare la richiesta dell'ingegnere, il tecnico utilizza la seguente istruzione SELECT per selezionare la chiave univoca, l'ID e l'area di ciascuna piantina dalla tabella BUILDINGFOOTPRINTS:

```
SELECT building_id, db2gse.ST_Area (footprint) "Area"
FROM BUILDINGFOOTPRINTS;
```

L'istruzione SELECT restituisce la seguente serie di risultati:

building_id	Area
506	+1.407680000000000E+003
1208	+2.557590000000000E+003
543	+1.807860000000000E+003
178	+2.086710000000000E+003
.	.
.	.
.	.

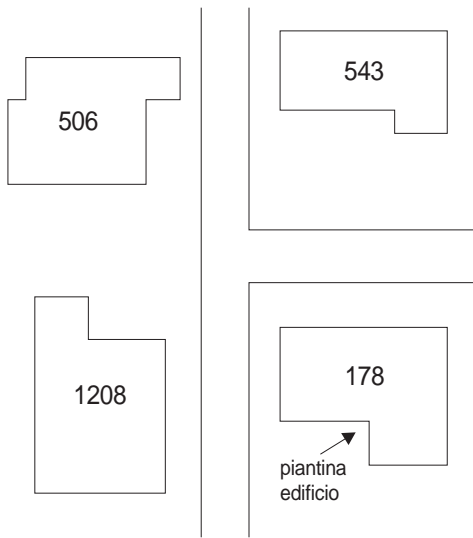


Figura 28. Utilizzo dell'area per la ricerca della piantina di un edificio. Quattro piantine etichettate con i relativi numeri ID vengono visualizzate nella parte laterale della strada adiacente.

ST_AsBinary

ST_AsBinary rileva un oggetto di geometria e restituisce la relativa rappresentazione WKB.

Sintassi

```
db2gse.ST_AsBinary(g db2gse.ST_Geometry)
```

Tipo di ritorno

BLOB(1m)

Esempi

Il seguente frammento di codifica illustra il modo in cui la funzione ST_AsBinary converte i multipoligoni delle piantine della tabella BUILDINGFOOTPRINTS nei multipoligoni WKB. Questi multipoligoni vengono trasferiti alla funzione draw_polygon dell'applicazione per la visualizzazione.

```
/* 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 rileva un oggetto di geometria e restituisce la relativa rappresentazione WKT.

Sintassi

db2gse.ST_AsText(g db2gse.ST_Geometry)

Tipo di ritorno

Varchar(4000)

Esempi

Nel seguente scenario, la funzione db2gse.ST_AsText converte il punto di ubicazione HAZARDOUS_SITES nella relativa descrizione di testo:

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,  
                                name      varchar(40),  
                                location  db2gse.ST_Point);
```

```
INSERT INTO HAZARDOUS_SITES  
VALUES (102,  
        'W. H. Kleenare Chemical Repository',  
        db2gse.ST_PointFromText('point (1020.12 324.02)',  
db2gse.coordref().srid));
```

```
SELECT site_id, name, cast(db2gse.ST_AsText(location) as varchar(40))  
       "Location"  
FROM HAZARDOUS_SITES;
```

L'istruzione SELECT restituisce la seguente serie di risultati:

SITE_ID	Name	Location
102	W. H. Kleenare Chemical Repository	POINT (1020.00000000 324.00000000)

ST_Boundary

ST_Boundary rileva un oggetto di geometria e restituisce il relativo limite combinato come oggetto di geometria.

Sintassi

```
db2gse.ST_Boundary(g db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

Nel seguente frammento di codifica viene creata una tabella denominata BOUNDARY_TEST. BOUNDARY_TEST contiene due colonne: GEOTYPE, definita come varchar, e G1, definita come geometria della superclasse. Le seguenti istruzioni INSERT inseriscono ciascuna geometria delle classi secondarie. La funzione ST_Boundary richiama il limite di ogni classe secondaria memorizzata nella colonna di geometria G1. La dimensione della figura geometrica risultante è sempre inferiore a quella della geometria di input. I punti e i multipunti determinano sempre un limite che corrisponde a una figura geometrica vuota di dimensione 1. Le stringhe e le multistringhe lineari restituiscono un limite per il multipunto di dimensione 0. Un poligono o un multipoligono restituisce sempre un limite per la multistringa lineare di dimensione 1.

```
CREATE TABLE BOUNDARY_TEST (GEOTYPE varchar(20), G1 db2gse.ST_Geometry)

INSERT INTO BOUNDARY_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
                              db2gse.coordref()..srid(0)))

INSERT INTO BOUNDARY_TEST
VALUES('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))

INSERT INTO BOUNDARY_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94, 10.02 20.01))',
                              db2gse.coordref()..srid(0)))

INSERT INTO BOUNDARY_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))

INSERT INTO BOUNDARY_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64), (9.55 23.75,15.36 30.11))',
```

```

db2gse.coordref()..srid(0))

INSERT INTO BOUNDARY_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
      25.02 34.15, 19.15 33.94,10.02 20.01)),
      ((51.71 21.73,73.36 27.04,71.52 32.87,
      52.43 31.90,51.71 21.73)))',
      db2gse.coordref()..srid(0))

SELECT GEOTYPE,
      CAST(db2gse.ST_AsText(db2gse.ST_Boundary (G1)) as varchar(280))
      "The boundary"
FROM BOUNDARY_TEST

```

GEOTYPE	The boundary
Point	POINT EMPTY
Linestring 25.64000000)	MULTIPOINT (10.02000000 20.01000000, 11.92000000
Polygon	MULTILINESTRING ((10.02000000 20.01000000, 19.15000000
	33.94000000, 25.02000000 34.15000000, 11.92000000 35.64000000, 10.02000000
	20.01000000))
Multipoint	POINT EMPTY
Multilinestring	MULTIPOINT (9.55000000 23.75000000, 10.02000000
	20.01000000, 11.92000000 25.64000000, 15.36000000 30.11000000)
Multipolygon	MULTILINESTRING ((51.71000000 21.73000000, 73.36000000
	27.04000000, 71.52000000 32.87000000, 52.43000000 31.90000000, 51.71000000
	21.73000000),(10.02000000 20.01000000, 19.15000000 33.94000000, 25.02000000
	34.15000000, 11.92000000 35.64000000, 10.02000000 20.01000000))

6 record(s) selected.

ST_Buffer

ST_Buffer rileva un oggetto di geometria e la distanza e restituisce l'oggetto geometrico che circonda l'oggetto di origine.

Sintassi

```
db2gse.ST_Buffer(g db2gse.ST_Geometry , adistance Double)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

L'ispettore regionale richiede un elenco dei luoghi pericolosi situati nel raggio di cinque miglia dalle zone protette come le scuole, gli asili e gli ospedali. Le zone protette vengono memorizzate nella tabella SENSITIVE_AREAS creata con la seguente istruzione CREATE TABLE. La colonna ZONE viene definita come un poligono, che viene memorizzato come perimetro di ciascuna zona protetta.

```
CREATE TABLE SENSITIVE_AREAS (id      integer,
                               name    varchar(128),
                               size    float,
                               type    varchar(10),
                               zone    db2gse.ST_Polygon);
```

I siti a rischio vengono memorizzati nella tabella HAZARDOUS_SITES creata con la seguente istruzione CREATE TABLE. La colonna LOCATION, definita come punto, memorizza un'ubicazione che rappresenta il centro geografico di ciascun sito a rischio.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);
```

Le tabelle SENSITIVE_AREAS e HAZARDOUS_SITES vengono unite dalla funzione db2gse.ST_Overlaps. Questa funzione restituisce 1 (TRUE) per tutte le righe SENSITIVE_AREAS i cui poligoni di zona si sovrappongono al raggio di cinque miglia del punto di ubicazione HAZARDOUS_SITES.

```
SELECT sa.name "Sensitive Areas", hs.name "Hazardous Sites"
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Overlaps(sa.zone, db2gse.ST_Buffer (hs.location,(5 * 5280)))
=1;
```

Nella Figura 29 a pagina 195, alcune zone protette in questo distretto amministrativo sono situate entro un raggio di cinque miglia dai luoghi a rischio. Questo raggio interseca la zona ospedaliera e la scuola. Mentre l'asilo si trova al sicuro lontano da questo raggio.

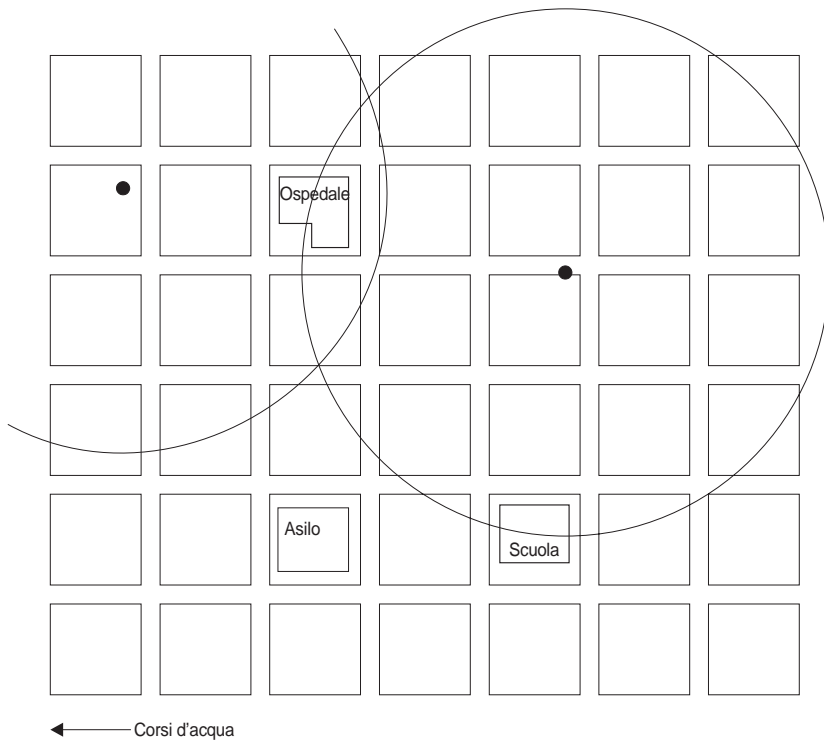


Figura 29. Un buffer con un raggio di cinque miglia viene applicato ad un punto

ST_Centroid

ST_Centroid rileva un poligono o un multipoligono e restituisce il relativo centro geometrico come punto.

Sintassi

```
db2gse.ST_Centroid(s db2gse.ST_Surface)
db2gse.ST_Centroid(ms db2gse.ST_MultiSurface)
```

Tipo di ritorno

Per la superficie: db2gse.ST_Point

Esempi

Il tecnico GIS della città desidera visualizzare i multipoligoni delle piantine degli edifici come singoli punti in un grafico di densità edilizia.

I progetti vengono memorizzati nella tabella BUILDINGFOOTPRINTS creata con la seguente istruzione CREATE TABLE:

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,
                                   lot_id integer,
                                   footprint db2gse.ST_MultiPolygon);
```

La funzione ST_Centroid restituisce il centro geometrico per ogni multipoligono delle piantine degli edifici. La funzione AsBinaryShape converte il punto del centro geometrico in una forma, la rappresentazione esterna riconosciuta dall'applicazione.

```
SELECT building_id,
       CAST(db2gse.AsBinaryShape(db2gse.ST_Centroid (footprint)) as blob(1m))
"Centroid"
FROM BUILDINGFOOTPRINTS;
```

ST_Contains

ST_Contains rileva due oggetti di geometria e restituisce 1 (TRUE) se il primo oggetto contiene il secondo; altrimenti restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Contains(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Nel seguente esempio sono riportate due tabelle. Una tabella contiene le piantine degli edifici di una città, mentre l'altra tabella contiene i relativi lotti. L'ingegnere desidera assicurarsi che tutte le piantine siano situate correttamente all'interno dei relativi lotti.

In entrambe le tabelle, il tipo di dati multipoligono memorizza la geometria delle piantine degli edifici e dei lotti. Il programma di progettazione del database ha selezionato i multipoligoni per entrambe le funzioni. E' possibile che i lotti risultino separati da condizioni geografiche naturali, ad esempio da un fiume, e le piantine possono essere costituite da più edifici.

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,  
                                lot_id      integer,  
                                footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (lot_id integer, lot db2gse.ST_MultiPolygon);
```

L'ingegnere seleziona innanzitutto gli edifici non completamente contenuti all'interno di un lotto.

```
SELECT building_id  
FROM BUILDINGFOOTPRINTS, LOTS  
WHERE db2gse.ST_Contains(lot, footprint) = 0;
```

La prima interrogazione restituirà un elenco di tutti gli ID degli edifici le cui piantine sono situate esternamente al poligono di un lotto. Tuttavia, queste informazioni non indicano se agli altri edifici è stato assegnato l'ID lotto corretto. Quindi la seconda interrogazione esegue un controllo dell'integrità dei dati nella colonna lot_id della tabella BUILDINGFOOTPRINTS.

```
SELECT bf.building_id "building id", bf.lot_id "buildings lot_id",  
       LOTS.lot_id "LOTS lot_id"  
FROM BUILDINGFOOTPRINTS bf, LOTS  
WHERE db2gse.ST_Contains(lot, footprint) = 1 AND LOTS.lot_id <> bf.lot_id;
```

Nella Figura 30 a pagina 198, le piantine degli edifici etichettate con i relativi ID si trovano all'interno dei lotti. Il perimetro del lotto viene rappresentato

con delle righe tratteggiate. Queste righe si estendono fino al punto centrale della strada e racchiudono completamente i lotti e le piantine degli edifici.

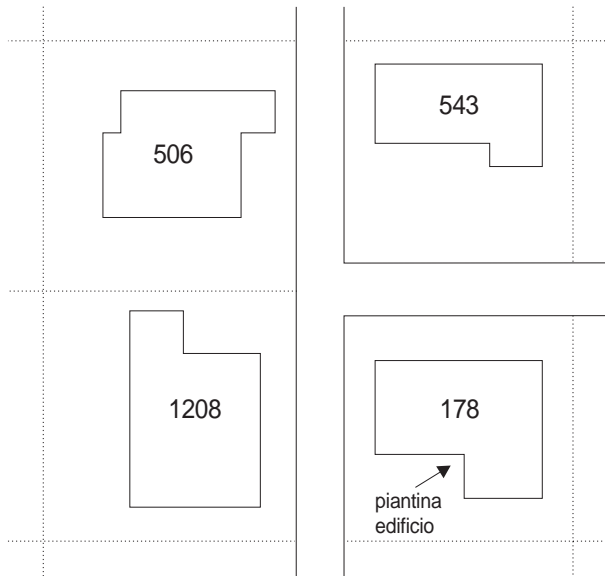


Figura 30. Utilizzo di ST_Contains per verificare che tutti gli edifici siano contenuti all'interno dei relativi lotti

ST_ConvexHull

ST_ConvexHull rileva un oggetto geometrico e restituisce un involucro convesso.

Sintassi

```
db2gse.ST_ConvexHull(g db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

L'esempio riporta la tabella CONVEXHULL_TEST che contiene due colonne: GEOTYPE e G1. La colonna GEOTYPE, di tipo varchar(20), memorizza il nome della classe secondaria della geometria nella colonna G1, che viene definita come geometria.

```
CREATE TABLE CONVEXHULL_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Ciascuna istruzione INSERT inserisce una geometria per ogni tipo di classe secondaria nella tabella CONVEXHULL_TEST.

```
INSERT INTO CONVEXHULL_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
      db2gse.coordref()..srid(0)))
```

```
INSERT INTO CONVEXHULL_TEST
VALUES('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
      11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO CONVEXHULL_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
      19.15 33.94,10.02 20.01))',
      db2gse.coordref()..srid(0)))
```

```
INSERT INTO CONVEXHULL_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
      11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO CONVEXHULL_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
      11.92 25.64),(9.55 23.75,15.36 30.11))',
      db2gse.coordref()..srid(0)))
```

```
INSERT INTO CONVEXHULL_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
      25.02 34.15, 19.15 33.94,10.02 20.01)),
```

```
((51.71 21.73,73.36 27.04,71.52 32.87,  
52.43 31.90,51.71 21.73)))',  
db2gse.coordref(..srid(0)))
```

L'istruzione SELECT elenca il nome della classe secondaria memorizzato nella colonna GEOTYPE e l'involucro convesso. L'involucro convesso generato dalla funzione ST_ConvexHull viene convertito in testo dalla funzione ST_AsText. Verrà successivamente convertito in varchar(256) perché output predefinito di ST_AsText è varchar(4000).

```
SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_ConvexHull(G1)))  
as varchar(256) "The convexhull"  
FROM CONVEXHULL_TEST
```

ST_CoordDim

ST_CoordDim restituisce le dimensioni delle coordinate del valore ST_Geometry. Per informazioni sulle dimensioni delle coordinate, consultare la sezione "Punti" a pagina 135.

Sintassi

```
db2gse.ST_CoordDim(g1 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La tabella `coorddim_test` viene creata con le colonne `geotype` e `g1`. La colonna `geotype` memorizza il nome della classe secondaria di geometria memorizzata nella colonna `g1`.

```
CREATE TABLE coorddim_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Le istruzioni `INSERT` inseriscono una classe secondaria di esempio nella tabella `coorddim_test`.

```
INSERT INTO coorddim_test VALUES(
    'Point', db2gse.ST_PointFromText('point (10.02 20.01)',
    db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
    'Linestring',
    db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,11.92 25.64)',
    db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
    'Polygon', db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,
    25.02 34.15, 19.15 33.94,10.02 20.01))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
    'Multipoint', db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
    11.92 25.64)', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
    'Multilinestring', db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,
    10.32 23.98,11.92 25.64),(9.55 23.75,15.36 30.11))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO coorddim_test VALUES(
    'Multipolygon',
    MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,25.02 34.15,
    19.15 33.94,10.02 20.01)),((51.71 21.73,73.36 27.04,71.52 32.87,
    52.43 31.90,51.71 21.73)))', db2gse.coordref()..srid(0))
)
```

L'istruzione SELECT elenca il nome della classe secondaria memorizzato nella colonna geotype con la dimensione delle coordinate corrispondente a geotype.

```
SELECT geotype, db2gse.ST_coordDim(g1)' coordinate_dimension'  
FROM coorddim_test
```

GEOTYPE	coordinate_dimension
ST_Point	2
ST_Linestring	2
ST_Polygon	2
ST_Multipoint	2
ST_Multilinestring	2
ST_Multipolygon	2

6 record(s) selected.

ST_Crosses

ST_Crosses rileva due oggetti di geometria e restituisce 1 (TRUE) se l'intersezione crea un oggetto geometrico la cui dimensione è inferiore a quella massima degli oggetti di origine. L'oggetto di intersezione contiene i punti interni a entrambe le geometrie di origine e non corrisponde a nessun oggetto di origine. Altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Crosses(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Il governo sta considerando una nuova normativa che stabilisce che tutte le discariche a rischio all'interno di un paese non possono essere situate in un raggio di cinque miglia dai corsi d'acqua. Il responsabile GIS regionale dispone di una rappresentazione accurata dei fiumi e dei flussi d'acqua memorizzati come multistringhe lineari nella tabella WATERWAYS. Tuttavia, questo responsabile ha soltanto una singola ubicazione di punto per ogni discarica a rischio.

```
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);
```

Per stabilire se l'ispettore regionale deve ricevere una notifica in caso di violazione della normativa proposta, è necessario che il responsabile GIS esegua il buffer delle zone a rischio e verifichi se i fiumi o i corsi d'acqua attraversano il poligono del buffer. Il predicato ST_Crosses confronta HAZARDOUS_SITES nel buffer con WATERWAYS. Pertanto, vengono restituiti solo i record i cui corsi d'acqua attraversano il raggio stabilito dalla normativa proposta.

```
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;
```

Nella Figura 31 a pagina 204, il buffer di cinque miglia delle zone delle discariche a rischio attraversa i fiumi o i corsi d'acqua dell'intero distretto regionale. I fiumi e i corsi d'acqua sono stati definiti come multistringa lineare. Quindi, la serie di risultati include tutti i segmenti delle stringhe lineari che fanno parte di quei segmenti che attraversano il raggio.

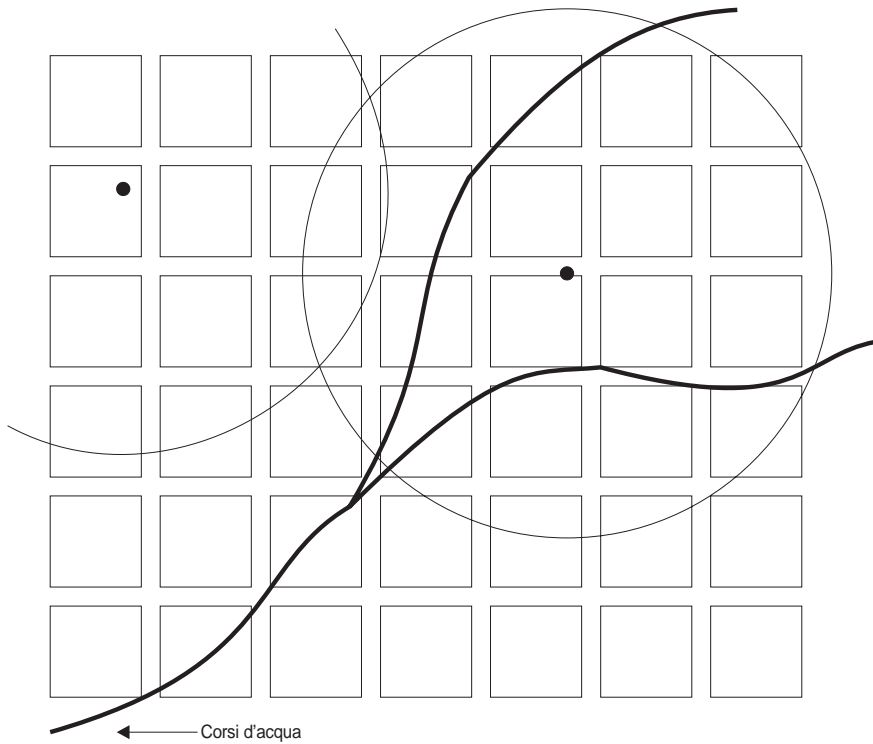


Figura 31. Utilizzo di `ST_Crosses` per la ricerca dei corsi d'acqua che attraversano una zona a rischio

ST_Difference

ST_Difference rileva due oggetti di geometria e restituisce un oggetto differente dagli oggetti di origine.

Sintassi

```
db2gse.ST_Difference(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

L'ingegnere desidera conoscere l'intera area della città non coperta da edifici. Quindi, l'ingegnere richiede la somma dell'area dei lotti successiva all'eliminazione dell'area edificabile.

```
CREATE TABLE BUILDINGFOOTPRINTS (building_id integer,  
                                lot_id      integer,  
                                footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (lot_id integer,  
                   lot     db2gse.ST_MultiPolygon);
```

L'ingegnere unifica le tabelle BUILDINGFOOTPRINTS e LOTS in lot_id. Quindi, calcola la somma dell'area della differenza dei lotti, sottraendo le piantine degli edifici.

```
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 rileva un oggetto di geometria e ne restituisce la dimensione.

Sintassi

```
db2gse.ST_Dimension(g1 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La tabella DIMENSION_TEST viene creata con le colonne GEOTYPE e G1. La colonna GEOTYPE memorizza il nome della classe secondaria di geometria memorizzata nella colonna G1.

```
CREATE TABLE DIMENSION_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Le istruzioni INSERT inseriscono una classe secondaria di esempio nella tabella DIMENSION_TEST.

```
INSERT INTO DIMENSION_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,
                                          11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
                                       19.15 33.94,10.02 20.01))',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
                                             11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
                                                  11.92 25.64),(9.55 23.75,15.36 30.11))',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO DIMENSION_TEST
VALUES('Multipolygon',
      db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
                                                25.02 34.15,19.15 33.94,10.02 20.01)),
                                ((51.71 21.73,73.36 27.04,71.52 32.87,
                                52.43 31.90,51.71 21.73)))',
                              db2gse.coordref()..srid(0)))
```

L'istruzione SELECT elenca il nome della classe secondaria memorizzato nella colonna GEOTYPE con la dimensione corrispondente a geotype.

```
SELECT geotype, db2gse.ST_Dimension(g1) "The dimension"  
FROM DIMENSION_TEST
```

Viene restituita la seguente serie di risultati.

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 rileva due oggetti di geometria e restituisce 1 (TRUE) se l'intersezione dei due oggetti produce una serie vuota; altrimenti restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Disjoint(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Una compagnia di assicurazioni desidera valutare la copertura assicurativa per un ospedale, un asilo o una scuola di una città. Una parte di questo processo comprende la determinazione dei rischi che le zone pericolose determinano per ciascuna istituzione. La compagnia di assicurazioni considera solo le istituzioni che non risultano a rischio di contaminazione. Il consulente GIS ingaggiato dalla compagnia di assicurazioni è stato incaricato di localizzare tutte le istituzioni che non si trovano nel raggio di cinque miglia da queste zone pericolose.

La tabella SENSITIVE_AREAS contiene diverse colonne che descrivono le istituzioni interessate oltre alla colonna ZONE che memorizza la geometria del poligono dell'istituzione.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name       varchar(128),
                                size       float,
                                type       varchar(10),
                                zone       db2gse.ST_Polygon);
```

La tabella HAZARDOUS_SITES memorizza l'identità dei siti nelle colonne SITE_ID e NAME, mentre l'effettiva ubicazione geografica di ogni sito viene memorizzata nella colonna LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                                name     varchar(128),
                                location  db2gse.ST_Point);
```

La seguente istruzione SELECT elenca i nomi di tutte le zone protette che non si trovano nel raggio di cinque miglia da una zona a rischio. La funzione ST_Intersects può sostituire la funzione ST_Disjoint in questa interrogazione se il risultato della funzione viene impostato su 0 piuttosto che su 1. Questo perché ST_Intersects e ST_Disjoint restituiscono il risultato opposto.

```
SELECT sa.name
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Disjoint(db2gse.ST_Buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

Nella Figura 32, le zone protette vengono confrontate al raggio di cinque miglia delle zone a rischio. L'asilo risulta l'unica zona protetta per la quale la funzione ST_Disjoint restituirà 1 (TRUE). La funzione ST_Disjoint restituisce 1 se le due geometrie non si intersecano.

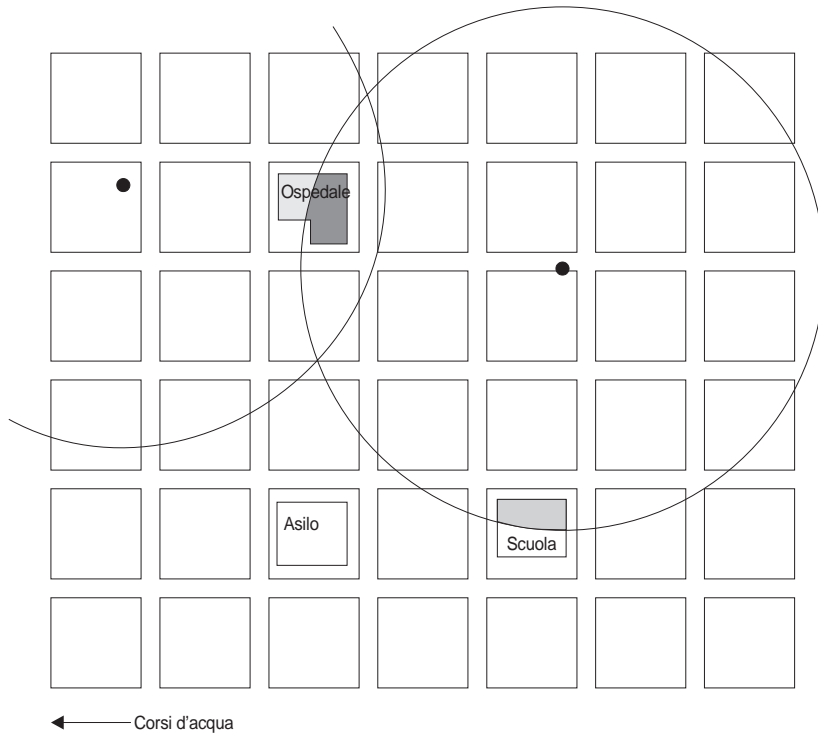


Figura 32. Utilizzo di ST_Disjoint per la ricerca degli edifici che non sono situati all'interno di un'area a rischio

ST_Distance

ST_Distance rileva due geometrie e restituisce la distanza più vicina.

Sintassi

```
db2gse.ST_Distance(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Double

Esempi

Un ingegnere desidera consultare un elenco di tutti gli edifici situati ad una determinata distanza (foot=piede) dai lotti.

La colonna BUILDING_ID della tabella BUILDINGFOOTPRINTS identifica ogni edificio in modo univoco. La colonna LOT_ID identifica il lotto a cui appartiene ciascun edificio. Il multipoligono delle piantine memorizza la geometria delle piantine di ciascun edificio.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
lot_id integer,  
footprint db2gse.ST_MultiPolygon);
```

La tabella LOTS memorizza l'ID lotto che identifica ogni LOT in modo univoco, e il multipoligono LOT che contiene la geometria lineare.

```
CREATE TABLE LOTS ( lot_id integer,  
lot db2gse.ST_MultiPolygon);
```

L'interrogazione restituisce un elenco di ID degli edifici situati entro il valore specificato di una determinata unità di misura dalla linea di demarcazione del lotto. La funzione ST_Distance segue un'unione spaziale tra le piantine e il limite dei multipoligoni dei lotti. Tuttavia, questa unione tra bf.lot_id e LOTS.lot_id garantisce che soltanto i multipoligoni che appartengono allo stesso lotto vengono confrontati dalla funzione ST_Distance.

```
SELECT bf.building_id  
FROM BUILDINGFOOTPRINTS bf, LOTS  
WHERE bf.lot_id = LOTS.lot_id AND  
db2gse.ST_Distance(bf.footprint, db2gse.ST_Boundary(LOTS.lot)) <= 1.0;
```

ST_Endpoint

ST_Endpoint rileva una stringa lineare e restituisce l'ultimo punto della stringa lineare.

Sintassi

```
db2gse.ST_Endpoint(c db2gse.ST_Curve)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

La tabella ENDPOINT_TEST memorizza la colonna integer GID che identifica ogni riga in modo univoco e la colonna stringa lineare LN1 che memorizza le stringhe lineari.

```
CREATE TABLE ENDPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Le istruzioni INSERT inseriscono le stringhe lineari nella tabella ENDPOINT_TEST. La prima non presenta misure o coordinate Z a differenza della seconda.

```
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)))
```

La seguente istruzione SELECT elenca la colonna GID con i risultati della funzione ST_Endpoint. La funzione ST_Endpoint genera un punto che viene convertito in testo dalla funzione ST_AsText. La funzione CAST viene utilizzata per ridurre il valore predefinito varchar(4000) della funzione ST_AsText in varchar(60).

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_Endpoint(ln1)) AS varchar(60))
"Endpoint"
FROM ENDPOINT_TEST
```

Viene restituita la seguente serie di risultati.

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 rileva un oggetto di geometria e restituisce la casella di limite come geometria.

Sintassi

```
db2gse.ST_Envelope(g db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La colonna GEOTYPE della tabella ENVELOPE_TEST memorizza il nome della classe secondaria di geometria memorizzata nella colonna G1.

```
CREATE TABLE ENVELOPE_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono ogni classe secondaria di geometria nella tabella ENVELOPE_TEST.

```
INSERT INTO ENVELOPE_TEST
VALUES('Point',
      db2gse.ST_PointFromText('point (10.02 20.01)',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES ('Linestring',
      db2gse.ST_LineFromText('linestring (10.01 20.01, 10.01 30.01,
                                      10.01 40.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES('Linestring',
      db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,11.92 25.64)',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES('Polygon',
      db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
                                      19.15 33.94,10.02 20.01))',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES('Multipoint',
      db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,11.92 25.64)',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES('Multilinestring',
      db2gse.ST_MLineFromText('multilinestring ((10.01 20.01,20.01 20.01,
                                      30.01 20.01), (30.01 20.01,40.01 20.01,50.01 20.01))',
                              db2gse.coordref()..srid(0)))
```

```
INSERT INTO ENVELOPE_TEST
VALUES('Multilinestring',
```

```

db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64), ( 9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0))

```

```

INSERT INTO ENVELOPE_TEST
VALUES('Multipolygon',
db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))',
db2gse.coordref()..srid(0))

```

L'istruzione SELECT elenca il nome della classe secondaria accanto al relativo involucro. Poiché la funzione ST_Envelope restituisce un punto, una stringa lineare o un poligono, l'output viene convertito in testo dalla funzione ST_AsText. La funzione CAST viene utilizzata per ridurre il valore predefinito varchar(4000) della funzione ST_AsText in varchar(280).

```

SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_Envelope(g1)) AS varchar(280))
"The envelope"
FROM ENVELOPE_TEST

```

Viene restituita la seguente serie di risultati.

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 confronta due oggetti di geometria e restituisce 1 (TRUE) se gli oggetti risultano identici; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Equals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Il tecnico GIS sospetta che alcuni dati della tabella BUILDINGFOOTPRINTS siano stati duplicati. Quindi, interroga la tabella per stabilire se i multipoligoni delle piantine risultano uguali.

La tabella BUILDINGFOOTPRINTS viene creata con la seguente istruzione. La colonna BUILDING_ID identifica gli edifici in modo univoco, la colonna LOT_ID identifica il lotto degli edifici e la colonna FOOTPRINT memorizza la geometria degli edifici.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
                                  lot_id      integer,  
                                  footprint   db2gse.ST_MultiPolygon);
```

La tabella BUILDINGFOOTPRINTS viene confrontata in modo spaziale a se stessa dal predicato ST_Equals, che restituisce 1 se rileva due multipoligoni uguali. La condizione bf1.building_id <> bf2.building_id viene utilizzata per eliminare il confronto della stessa geometria.

```
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 rileva un poligono e restituisce il relativo anello esterno come stringa lineare.

Sintassi

db2gse.ST_ExteriorRing(s db2gse.ST_Polygon)

Tipo di ritorno

db2gse.ST_LineString

Esempi

Un ornitologo che studia gli uccelli che vivono su varie isole dei mari del sud, sa che le zone ricche di cibo per una particolare specie di uccelli sono quelle limitate alla zona costiera. Per calcolare la portata dell'isola, l'ornitologo necessita del perimetro dell'isola. Sebbene su alcune isole vi siano diversi stagni, le linee di costa degli stagni sono abitate esclusivamente da un'altra specie di uccelli molto aggressiva. Pertanto, l'ornitologo richiede il perimetro dell'anello esterno delle isole.

Le colonne ID e NAME della tabella ISLANDS identificano ogni isola e la colonna LAND di tipo ST_Polygon memorizza la geometria di ogni isola.

```
CREATE TABLE ISLANDS (id integer,  
                        name varchar(32),  
                        land db2gse.ST_Polygon);
```

La funzione ST_ExteriorRing estrae l'anello esterno da ogni poligono delle isole come stringa lineare. La lunghezza della stringa lineare viene stabilita dalla funzione Length. Le lunghezze delle stringhe lineari vengono sommate dalla funzione SUM.

```
SELECT SUM(db2gse.ST_Length(db2gse.ST_ExteriorRing (land))) FROM ISLANDS;
```

Nella Figura 33 a pagina 216, gli anelli esterni delle isole rappresentano l'interfaccia ecologica che ciascuna isola condivide con il mare. Alcune isole hanno dei laghi, che vengono rappresentati dagli anelli interni dei poligoni.

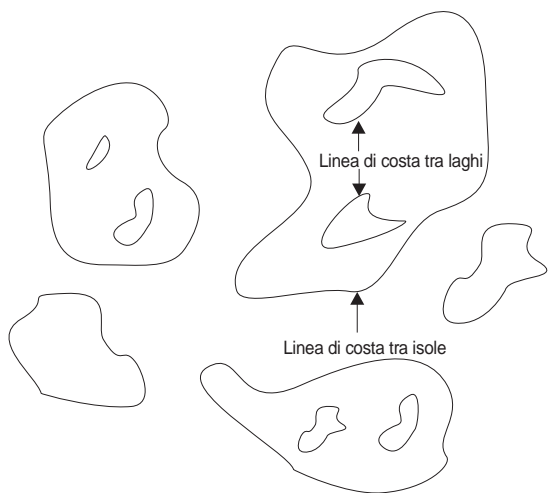


Figura 33. Utilizzo di `ST_ExteriorRing` per stabilire la lunghezza di una linea di costa dell'isola

ST_GeometryFromText

ST_GeometryFromText rileva una rappresentazione WKT e un'identità del sistema di riferimento spaziale e restituisce un oggetto di geometria.

Sintassi

```
db2gse.ST_GeometryFromText(geometryTaggedText Varchar(4000), cr
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La tabella GEOMETRY_TEST contiene la colonna integer `GID`, che identifica ciascuna riga in modo univoco, e la colonna `G1`, che memorizza la geometria.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Le istruzioni `INSERT` inseriscono i dati nelle colonne `GID` e `G1` della tabella `GEOMETRY_TEST`. La funzione `ST_GeometryFromText` converte la rappresentazione di testo di ogni geometria nella corrispondente classe secondaria DB2 Spatial Extender istanziabile.

```
INSERT INTO GEOMETRY_TEST
VALUES(1, db2gse.ST_GeometryFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES (2,
db2gse.ST_GeometryFromText('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_GeometryFromText('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_GeometryFromText('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_GeometryFromText('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_GeometryFromText('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 rileva una rappresentazione WKB e un'identità del sistema di riferimento spaziale e restituisce un oggetto di geometria.

Sintassi

db2gse.ST_GeomFromWKB(WKBGeometry Blob(1M), cr db2gse.coordref)

Tipo di ritorno

db2gse.ST_Geometry

Esempi

Il seguente frammento di codifica C contiene funzioni ODBC integrate alle funzioni SQL DB2 Spatial Extender che inseriscono i dati nella tabella LOTS.

La tabella LOTS è stata creata con due colonne: la colonna LOT_ID, che identifica in modo univoco ogni lotto, e la colonna LOT del multipoligono, che contiene la geometria di ciascun lotto.

```
CREATE TABLE LOTS ( lot_id integer,
                    lot      db2gse.ST_MultiPolygon);
```

La funzione ST_GeomFromWKB converte le rappresentazioni WKB nella geometria DB2 Spatial Extender. L'intera istruzione INSERT viene copiata nella stringa wkb_sql char. L'istruzione INSERT contiene i contrassegni di parametro per accettare i dati LOT_ID e LOT in modo dinamico.

```
/* 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_GeometryN

ST_GeometryN rileva una raccolta ed un indice integer e restituisce l'*ennesimo* oggetto di geometria nella raccolta.

Sintassi

```
db2gse.ST_GeometryN(g db2gse.ST_GeomCollection, n Integer)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

L'ingegnere desidera conoscere se le piantine degli edifici si trovano tutte all'interno del primo poligono del multipoligono del lotto.

La colonna BUILDING_ID della tabella BUILDINGFOOTPRINTS identifica ogni edificio in modo univoco. La colonna LOT_ID identifica il lotto a cui appartiene ciascun edificio. La colonna FOOTPRINT memorizza la geometria dell'edificio.

```
CREATE TABLE BUILDINGFOOTPRINTS ( building_id integer,  
                                  lot_id         integer,  
                                  footprint      db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer,  
                    lot     db2gse.ST_MultiPolygon);
```

L'interrogazione elenca BUILDINGFOOTPRINTS building_id e lot_id per tutte le piantine degli edifici che si trovano all'interno del primo poligono del lotto. La funzione ST_GeometryN restituisce un primo poligono del lotto nella schiera di multipoligoni.

```
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 rileva un oggetto ST_Geometry e restituisce il relativo tipo di geometria come stringa.

Sintassi

```
db2gse.ST_GeometryType (g db2gse.ST_Geometry)
```

Tipo di ritorno

Varchar(4000)

Esempi

La tabella GEOMETRYTYPE_TEST contiene la colonna di geometria G1.

```
CREATE TABLE GEOMETRYTYPE_TEST(g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono ogni classe secondaria di geometria nella colonna G1.

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeometryFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES (db2gse.ST_GeometryFromText('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_GeometryFromText('multipoint (10.02
20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeometryFromText('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_GeometryFromText('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)))
```

La seguente istruzione SELECT elenca il tipo di geometria di ogni classe secondaria memorizzata nella colonna di geometria G1.

```
SELECT db2gse.ST_GeometryType(g1) "Geometry type" FROM GEOMETRYTYPE_TEST
```

Viene restituita la seguente serie di risultati.

Geometry type

ST_Point
ST_LineString
ST_Polygon
ST_MultiPoint
ST_MultiLineString
ST_MultiPolygon

6 record(s) selected.

ST_InteriorRingN

Restituisce l'*n*-esimo anello interno di un poligono come stringa lineare. Gli anelli non sono organizzati in base all'orientamento geometrico. Sono organizzati in base alle regole definite dalle routine di verifica della geometria interna. Pertanto, l'ordine degli anelli non può essere predefinito.

Sintassi

ST_InteriorRingN(p ST_Polygon, n Integer)

Tipo di ritorno

db2gse.ST_LineString

Esempi

Un ornitologo che studia gli uccelli che vivono su varie isole dei mari del sud, sa che le zone ricche di cibo per una particolare specie di uccelli sono quelle limitate alla zona costiera. Su alcune isole si trovano diversi laghi. Le linee di costa dei laghi sono abitate esclusivamente da un'altra specie molto aggressiva. L'ornitologo riconosce che per ogni isola, se il perimetro dei laghi supera una determinata soglia, le specie aggressive risulteranno così numerose da minacciare l'altra specie. Pertanto, l'ornitologo richiede il perimetro aggregato dell'anello interno delle isole.

Nella Figura 34, gli anelli esterni delle isole rappresentano l'interfaccia ecologica che ciascuna isola condivide con il mare. Alcune isole hanno dei laghi, che vengono rappresentati dagli anelli interni dei poligoni.



Figura 34. Utilizzo di ST_InteriorRingN per stabilire la lunghezza delle linee costiere dei laghi di ciascuna isola

Le colonne ID e NAME della tabella ISLANDS identificano ogni isola e la colonna LAND memorizza la geometria di ogni isola.

```
CREATE TABLE ISLANDS (id    integer,
                        name  varchar(32),
                        land  db2gse.ST_Polygon);
```

Il seguente programma ODBC utilizza la funzione ST_InteriorRingN per estrarre l'anello interno (lago) da ogni poligono dell'isola come stringa lineare. Il perimetro della stringa lineare restituito dalla funzione Length viene sommato e visualizzato con l'ID dell'isola.

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sg.h"
#include "sgerr.h"
#include "sqlclil.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))) to the variable lake_perimeter */

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 Complete ...\n" );

}

static void check_sql_err (SQLHDBC handle, SQLHSTMT hstmt, LONG rc,
                          CHAR *str)
{

    SDWORD dbms_err = 0;
    SWORD length;
    UCHAR err_msg[SQL_MAX_MESSAGE_LENGTH], state[6];

    if (rc != SQL_SUCCESS)
    {
        SQLError (SQL_NULL_HENV, handle, hstmt, state, &dbms_err,
                 err_msg, SQL_MAX_MESSAGE_LENGTH - 1, &length);
        printf ("%s ERROR (%d): DBMS code:%d, SQL state: %s, message:
                \n %s\n", str, rc, dbms_err, state, err_msg);

        if (handle)
        {
            SQLDisconnect (handle);
            SQLFreeConnect (handle);
        }
        exit(1);
    }
}
}

```

ST_Intersection

ST_Intersection rileva due oggetti di geometria e restituisce la serie di intersezioni come oggetto di geometria.

Sintassi

```
db2gse.ST_Intersection(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

I vigili del fuoco devono esaminare le aree degli ospedali, delle scuole e degli asili che vengono intersecate dal raggio di una possibile area a rischio di contaminazione.

Le zone protette vengono memorizzate nella tabella SENSITIVE_AREAS creata con la seguente istruzione CREATE TABLE. La colonna ZONE viene definita come un poligono, che memorizza il perimetro di ciascuna zona protetta.

```
CREATE TABLE SENSITIVE_AREAS (id      integer,
                                name    varchar(128),
                                size     float,
                                type     varchar(10),
                                zone     db2gse.ST_Polygon);
```

I siti a rischio vengono memorizzati nella tabella HAZARDOUS_SITES creata con la seguente istruzione CREATE TABLE. La colonna LOCATION, definita come punto, memorizza un'ubicazione che rappresenta il centro geografico di ciascun sito a rischio.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                                name     varchar(128),
                                location db2gse.ST_Point);
```

La funzione buffer genera a un buffer di cinque miglia che circonda le ubicazioni delle discariche a rischio. La funzione ST_Intersection genera poligoni dall'intersezione dei poligoni delle discariche a rischio incluse nel buffer e aree protette. La funzione ST_Area restituisce l'area dei poligoni dell'intersezione che viene sommata per ciascuna zona a rischio dalla funzione SUM. La clausola GROUP BY indirizza l'interrogazione in modo da aggregare le aree intersecate in base al site_ID della zona a rischio.

```
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;
```

Nella Figura 35 a pagina 230, i cerchi rappresentano i poligoni di cui è stato eseguito il buffer che circoscrivono le zone a rischio. L'intersezione di questi

poligoni con quelli delle zone protette produce altri tre poligoni. L'ospedale nell'angolo sinistro superiore della figura viene intersecato due volte, mentre la scuola nell'angolo destro inferiore viene intersecata una sola volta.

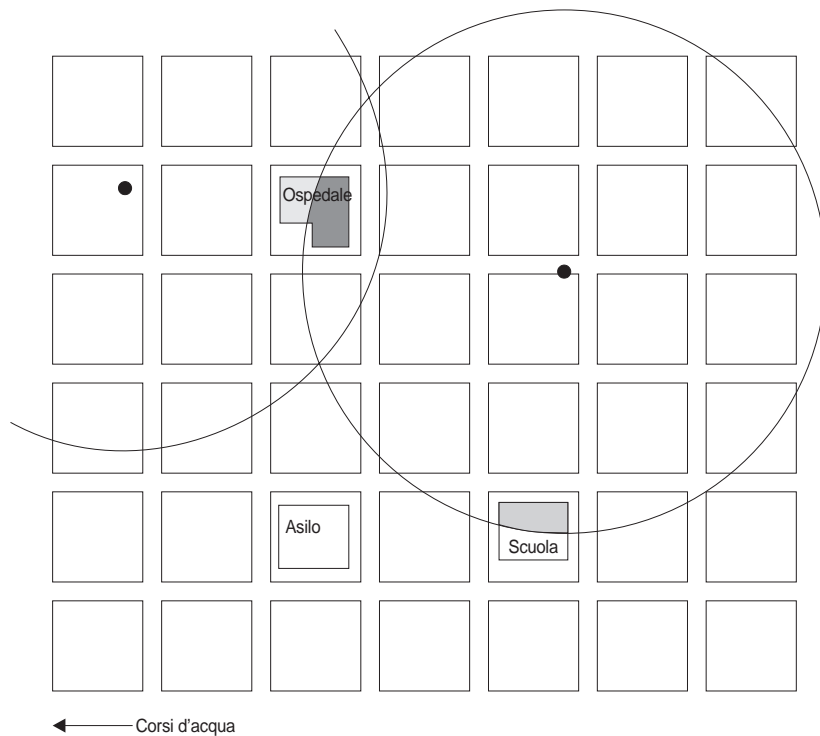


Figura 35. Utilizzo di `ST_Intersection` per determinare la dimensione dell'area a rischio di ogni edificio

ST_Intersects

ST_Intersects rileva due geometrie e restituisce 1 (TRUE), se l'intersezione delle due geometrie non determina una serie vuota. Altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Intersects(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

I vigili del fuoco richiedono un elenco di tutte le zone protette situate in un raggio di cinque miglia dalle aree a rischio.

Le zone protette vengono memorizzate nella tabella SENSITIVE_AREAS creata con la seguente istruzione CREATE TABLE. La colonna ZONE viene definita come un poligono, che memorizza il perimetro di ciascuna zona protetta.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

I siti a rischio vengono memorizzati nella tabella HAZARDOUS_SITES creata con la seguente istruzione CREATE TABLE. La colonna LOCATION, definita come punto, memorizza un'ubicazione che rappresenta il centro geografico di ciascun sito a rischio.

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

L'interrogazione restituisce un elenco di nomi delle zone protette e delle aree a rischio per quelle zone protette che intersecano il buffer di cinque miglia delle aree a rischio.

```
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 rileva una stringa o una multistringa lineare e restituisce 1 (TRUE) se è chiusa; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_IsClosed(c db2gse.ST_Curve)
db2gse.ST_IsClosed(mc db2gse.ST_MultiCurve)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella CLOSED_LINESTRING che contiene una singola colonna stringa lineare.

```
CREATE TABLE CLOSED_LINESTRING (ln1 db2gse.ST_LineString)
```

Le seguenti istruzioni INSERT inseriscono due record nella tabella CLOSED_LINESTRING. Il primo record non è una stringa lineare chiusa a differenza del secondo.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano i risultati della funzione ST_IsClosed. La prima riga restituisce 0 perché la stringa lineare non è chiusa, mentre la seconda riga restituisce 1 perché la stringa lineare è chiusa.

```
SELECT db2gse.ST_IsClosed(ln1) "Is it closed" FROM CLOSED_LINESTRING
```

```
Is it closed
-----
           0
           1
```

2 record(s) selected.

La seguente istruzione CREATE TABLE crea la tabella CLOSED_MULTILINESTRING che contiene una singola colonna multistringa lineare.

```
CREATE TABLE CLOSED_MULTILINESTRING (m1n1 db2gse.ST_MultiLineString)
```

Le seguenti istruzioni INSERT inseriscono due record in CLOSED_MULTILINESTRING, un record multistringa lineare non chiuso e un altro chiuso.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano i risultati della funzione ST_IsClosed. La prima riga restituisce 0 perché la multistringa lineare non è chiusa, mentre la seconda riga restituisce 1 perché la multistringa lineare è chiusa. Una multistringa lineare è chiusa se tutti gli elementi delle relative stringhe lineari sono chiusi.

```
SELECT db2gse.ST_IsClosed(mln1) "Is it closed" FROM CLOSED_MULTILINESTRING
```

```
Is it closed
```

```
-----
```

```
0
```

```
1
```

```
2 record(s) selected.
```

ST_IsEmpty

ST_IsEmpty rileva un oggetto di geometria e restituisce 1 (TRUE) se è vuoto; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_IsEmpty(g db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella EMPTY_TEST che contiene due colonne. La colonna GEOTYPE memorizza il tipo di dati della classe secondaria memorizzata nella colonna G1.

```
CREATE TABLE EMPTY_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono due record per le classe secondarie di geometria il punto, la stringa lineare e il poligono. Un record è vuoto e l'altro non è vuoto.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano il tipo di geometria dalla colonna GEOTYPE e i risultati della funzione 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 rileva una stringa lineare e restituisce 1 (TRUE) se è un anello (la stringa lineare è chiusa e semplice); altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_IsRing(c db2gse.ST_Curve)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella RING_LINestring che contiene una singola colonna stringa lineare denominata LN1.

```
CREATE TABLE RING_LINestring (ln1 db2gse.ST_LineString)
```

Le seguenti istruzioni INSERT inseriscono tre stringhe lineari nella colonna LN1. La prima riga contiene una stringa lineare non chiusa e quindi non è un anello. La seconda riga contiene una stringa lineare chiusa e semplice, quindi è un anello. La terza riga è una stringa lineare chiusa ma non semplice perché interseca il proprio interno, quindi non è un anello.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano i risultati della funzione ST_IsRing. La prima e la terza riga restituiscono 0. Questo perché le stringhe lineari non sono anelli a differenza della seconda riga che restituisce 1.

```
SELECT db2gse.ST_IsRing(ln1) "Is it ring" FROM RING_LINestring
```

```
Is it ring
-----
          0
          1
          0
```

```
3 record(s) selected.
```

ST_IsSimple

ST_IsSimple rileva un oggetto di geometria e restituisce 1 (TRUE) se l'oggetto è semplice; altrimenti, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_IsSimple(g db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella ISSIMPLE_TEST che contiene due colonne. La colonna PID, di tipo smallint, contiene l'identificativo univoco per ogni riga. La colonna di geometria G1 memorizza gli esempi di geometria semplici e non semplici.

```
CREATE TABLE ISSIMPLE_TEST (pid smallint, g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono due record nella tabella ISSIMPLE_TEST. Il primo è semplice perché è una stringa lineare che non interseca il proprio interno. Il secondo non è semplice perché interseca l'interno.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano i risultati della funzione ST_IsSimple. Il primo record restituisce 1 perché la stringa lineare è semplice, mentre il secondo record restituisce 0 perché la stringa lineare non è semplice.

```
SELECT ST_IsSimple(g1)
FROM ISSIMPLE_TEST
```

```
g1
-----
1
0
```

ST_IsValid

ST_IsValid rileva ST_Geometry e restituisce 1 (TRUE) se è valido, altrimenti restituisce 0 (FALSE). Una geometria inserita in un database DB2 è sempre valida perchè DB2 Spatial Extender convalida sempre i dati spaziali prima di accettarli. Tuttavia, è possibile che altri fornitori DBMS non eseguano la convalida, ma richiedano all'applicazione di eseguirla.

Sintassi

```
db2gse.ST_IsValid(g db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La tabella valid_test viene creata con le colonne geotype e g1. La colonna geotype memorizza il nome della classe secondaria di geometria memorizzata nella colonna g1.

```
CREATE TABLE valid_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

Le istruzioni INSERT inseriscono una classe secondaria di esempio nella tabella valid_test.

```
INSERT INTO valid_test VALUES(
    'Point', db2gse.ST_PointFromText('point (10.02 20.01)',
    db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Linestring',
    db2gse.ST_LineFromText('linestring (10.02 20.01,10.32 23.98,11.92 25.64)',
    db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Polygon', db2gse.ST_PolyFromText('polygon ((10.02 20.01,11.92 35.64,
    25.02 34.15, 19.15 33.94,10.02 20.01))', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multipoint', db2gse.ST_MPointFromText('multipoint (10.02 20.01,10.32 23.98,
    11.92 25.64)', db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multilinestring', db2gse.ST_MLineFromText('multilinestring ((10.02 20.01,
    10.32 23.98,11.92 25.64),(9.55 23.75,15.36 30.11))',
    db2gse.coordref()..srid(0))
)
```

```
INSERT INTO valid_test VALUES(
    'Multipolygon',
```

```

db2gse.ST_MPolyFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15,19.15 33.94,10.02 20.01)),((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))', db2gse.coordref()..srid(0))
)

```

L'istruzione SELECT elenca il nome della classe secondaria memorizzato nella colonna geotype con la dimensione corrispondente a geotype.

```

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 rileva una stringa o una multistringa lineare e restituisce la relativa lunghezza.

Sintassi

```
db2gse.ST_Length(c db2gse.ST_Curve)
db2gse.ST_Length(mc db2gse.ST_MultiCurve)
```

Tipo di ritorno

Double

Esempi

Un ecologista locale sta studiando i flussi di migrazione dei salmoni nei corsi d'acqua di un paese. Questo ecologista desidera sapere la lunghezza di tutti i fiumi e i corsi d'acqua che attraversano il paese.

La seguente istruzione CREATE TABLE crea la tabella WATERWAYS. Le colonne ID e NAME identificano ogni fiume e corso d'acqua memorizzato nella tabella. La colonna WATER è una mutistringa lineare poiché questi sistemi fluviali sono spesso degli aggregati di altre stringhe lineari.

```
CREATE TABLE WATERWAYS (id integer, name varchar(128),
water db2gse.ST_MultiLineString);
```

Le seguente istruzione SELECT utilizza la funzione ST_Length per restituire il nome e la lunghezza di ogni corso d'acqua all'interno del paese.

```
SELECT name, db2gse.ST_Length(water) "Length"
FROM WATERWAYS;
```

La Figura 36 a pagina 241 visualizza i fiumi e i corsi d'acqua che attraversano il perimetro del paese.

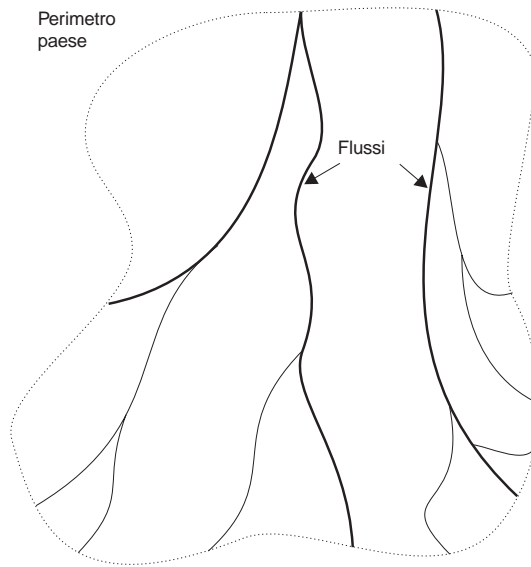


Figura 36. Utilizzo di ST_Length per determinare la lunghezza totale dei corsi d'acqua di un paese

ST_LineFromText

ST_LineFromText rileva una rappresentazione WKT di tipo stringa lineare e un'identità del sistema di riferimento spaziale e restituisce una stringa lineare.

Sintassi

```
db2gse.ST_LineFromText(lineStringTaggedText Varchar(4000), cr  
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_LineString
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella LINESTRING_TEST che contiene una singola colonna stringa lineare LN1.

```
CREATE TABLE LINESTRING_TEST (ln1 db2gse.ST_LineString)
```

La seguente istruzione INSERT inserisce una stringa lineare nella colonna LN1 utilizzando la funzione ST_LineFromText.

```
INSERT INTO LINESTRING_TEST  
VALUES (db2gse.ST_LineFromText('linestring(10.01 20.03,20.94 21.34,  
35.93 19.04)', db2gse.coordref()..srid(0)))
```

ST_LineFromWKB

ST_LineFromWKB rileva una rappresentazione WKB di tipo stringa lineare e un'identità del sistema di riferimento spaziale e restituisce una stringa lineare.

Sintassi

db2gse.ST_LineFromWKB(WKBLineString Blob(1M), cr db2gse.coordref)

Tipo di ritorno

db2gse.ST_LineString

Esempi

Il seguente frammento di codifica compila la tabella SEWERLINES con l'ID univoco, la classe della dimensione e la geometria di ogni rete fognaria.

La tabella SEWERLINES viene creata con tre colonne. La prima colonna, SEWER_ID, identifica ciascuna rete fognaria in modo univoco. La seconda colonna, CLASS, di tipo integer identifica il tipo di rete fognaria, generalmente associata alla capacità. La terza colonna, SEWER, di tipo stringa lineare memorizza la geometria della rete fognaria.

```
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 rileva una rappresentazione WKT di tipo multistringa lineare e un'identità del sistema di riferimento spaziale e restituisce una multistringa lineare.

Sintassi

```
db2gse.ST_MLineFromText(multiLineStringTaggedText String, cr
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiLineString
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella MLINESTRING_TEST. MLINESTRING_TEST ha due colonne: la colonna smallint GID, che identifica ogni riga in modo univoco, e la colonna della multistringa lineare ML1.

```
CREATE TABLE ST_MLINESTRING_TEST (gid smallint, ml1 db2gse.ST_MultiLineString)
```

La seguente istruzione INSERT inserisce la multistringa lineare con la funzione ST_MLineFromText.

```
INSERT INTO MLINESTRING_TEST
VALUES (1, db2gse.ST_MLineFromText('multilinestring((10.01 20.03,10.52 40.11,
30.29 41.56,31.78 10.74),
(20.93 20.81, 21.52 40.10))',
db2gse.coordref()..srid(0)))
```

ST_MLineFromWKB

ST_MLineFromWKB rileva una rappresentazione WKB di tipo multistringa lineare e un'identità del sistema di riferimento spaziale e restituisce una multistringa lineare.

Sintassi

```
db2gse.ST_MLineFromWKB(WKBMultiLineString Blob(1M), cr  
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiLineString
```

Esempi

Il seguente frammento di codifica compila la tabella WATERWAYS utilizzando un ID univoco, un nome e una multistringa lineare.

La tabella WATERWAYS viene creata con le colonne ID e NAME che identificano ogni flusso o l'intero sistema fluviale memorizzato nella tabella. La colonna WATER è una mutistringa lineare poiché questi sistemi fluviali sono spesso degli aggregati di altre stringhe lineari.

```
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 rileva una rappresentazione WKT di tipo multipunto e un'identità del sistema di riferimento spaziale e restituisce un multipunto.

Sintassi

```
db2gse.ST_MPointFromText(multiPointTaggedText Varchar(4000), cr  
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiPoint
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella MULTIPOINT_TEST con una singola colonna multipunto, MPT1.

```
CREATE TABLE MULTIPOINT_TEST (mpt1 db2gse.ST_MultiPoint)
```

La seguente istruzione INSERT inserisce un multipunto nella colonna MPT1 utilizzando la colonna ST_MPointFromText.

```
INSERT INTO MULTIPOINT_TEST  
VALUES (1, db2gse.ST_MPointFromText('multipoint(10.01 20.03,10.52 40.11,  
30.29 41.56,31.78 10.74)',  
db2gse.coordref()..srid(0)))
```

ST_MPointFromWKB

ST_MPointFromWKB rileva una rappresentazione WKB di tipo multipunto e un'identità del sistema di riferimento spaziale per restituire un multipunto.

Sintassi

db2gse.ST_MPointFromWKB(WKBMultiPoint Blob(1M), cr db2gse.coordref)

Tipo di ritorno

db2gse.ST_MultiPoint

Esempi

Questo frammento di codifica compila la tabella SPECIES_SITINGS.

La tabella SPECIES_SITINGS viene creata con tre colonne. Le colonne SPECIES e GENUS identificano le righe in modo univoco mentre il multipunto SITINGS memorizza le relative ubicazioni.

```
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 rileva una rappresentazione WKT di tipo multipoligono e un'identità del sistema di riferimento spaziale e restituisce un multipoligono.

Sintassi

```
db2gse.ST_MPolyFromText(multiPolygonTaggedText Varchar(4000), cr
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_MultiPolygon
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella MULTIPOLYGON_TEST, che contiene una singola colonna, MPL1.

```
CREATE TABLE MULTIPOLYGON_TEST (mpl1 db2gse.ST_MultiPolygon)
```

La seguente istruzione INSERT inserisce un multipoligono nella colonna MPL1 utilizzando la funzione ST_MPolyFromText.

```
INSERT INTO MULTIPOLYGON_TEST VALUES (
db2gse.ST_MPolyFromText('multipolygon(((10.01 20.03,10.52 40.11,
30.29 41.56,31.78 10.74,10.01 20.03),(21.23 15.74,21.34 35.21,28.94 35.35,
29.02 16.83, 21.23 15.74)),((40.91 10.92,40.56 20.19,
50.01 21.12,51.34 9.81, 40.91 10.92)))',
db2gse.coordref()..srid(0)))
```

ST_MPolyFromWKB

ST_MPolyFromWKB rileva una rappresentazione WKB di tipo multipoligono e un'identità del sistema di riferimento spaziale e restituisce un multipoligono.

Sintassi

db2gse.ST_MPolyFromWKB(WKBMultiPolygon Blob(1M), cr db2gse.coordref)

Tipo di ritorno

db2gse.ST_MultiPolygon

Esempi

Questo frammento di codifica compila la tabella LOTS.

La tabella LOTS memorizza LOT_ID, che identifica ogni lotto in modo univoco, e il multipoligono LOT che contiene la geometria lineare.

```
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 rileva una raccolta e restituisce il numero di geometrie della raccolta.

Sintassi

```
db2gse.ST_NumGeometries(g db2gse.ST_GeomCollection)
```

Tipo di ritorno

Integer

Esempi

L'ingegnere della città desidera conoscere il numero effettivo degli edifici associati a ciascuna piantina.

Le piantine degli edifici vengono memorizzate nella tabella

BUILDINGFOOTPRINTS creata con la seguente istruzione CREATE TABLE:

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                   lot_id      integer,
                                   footprint   db2gse.ST_MultiPolygon);
```

La seguente istruzione SELECT utilizza la funzione ST_NumGeometries per elencare BUILDING_ID che identifica ciascun edificio in modo univoco e il numero di edifici in ogni piantina.

```
SELECT building_id, db2gse.ST_NumGeometries (footprint) "Number of buildings"
FROM BUILDINGFOOTPRINTS;
```

ST_NumInteriorRing

ST_NumInteriorRing rileva un poligono e restituisce il numero di anelli interni in esso contenuti.

Sintassi

db2gse.NumInteriorRing(p db2gse.ST_Polygon)

Tipo di ritorno

Integer

Esempi

Un ornitologo che desidera studiare gli uccelli che vivono su varie isole dei mari del sud, sa che le zone ricche di cibo per una particolare specie di uccelli sono quelle limitate alle isole in cui sono presenti laghi con acqua dolce. Quindi, desidera conoscere le isole che contengono uno o più laghi.

La seguente istruzione CREATE TABLE crea la tabella ISLANDS. Le colonne ID e NAME della tabella ISLANDS identificano ogni isola e la colonna LAND memorizza la geometria di ogni isola.

```
CREATE TABLE ISLANDS (id integer, name varchar(32), land db2gse.ST_Polygon);
```

Poiché gli anelli interni rappresentano i laghi, la funzione ST_NumInteriorRing viene utilizzata per elencare solo le isole che presentano almeno un anello interno.

```
SELECT name FROM ISLANDS WHERE db2gse.ST_NumInteriorRing(land) > 0;
```

ST_NumPoints

ST_NumPoints rileva una stringa lineare e restituisce il relativo numero di punti.

Sintassi

```
db2gse.ST_NumPoints(l db2gse.ST_LineString)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella NUMPOINTS_TEST. La colonna GEOTYPE contiene il tipo di geometria memorizzato nella colonna G1.

```
CREATE TABLE NUMPOINTS_TEST (geotype varchar(12), g1 db2gse.ST_Geometry)
```

La seguente istruzione INSERT inserisce una stringa lineare.

```
INSERT INTO NUMPOINTS_TEST VALUES( linestring,  
db2gse.ST_LineFromText('linestring (10.02 20.01, 23.73 21.92)',  
db2gse.coordref()..srid(0)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente elencano il tipo di geometria e il numero di punti contenuti.

```
SELECT geotype, db2gse.ST_NumPoints(g1)  
FROM NUMPOINTS_TEST
```

```
GEOTYPE      Number of points  
-----  
ST_linestring      2  
1 record(s) selected.
```

ST_OrderingEquals

ST_OrderingEquals confronta le due geometrie e restituisce 1 (TRUE) se le geometrie sono uguali e le coordinate sono nello stesso ordine; in caso contrario restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_OrderingEquals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

La seguente istruzione CREATE TABLE crea la tabella LINESTRING_TEST, che contiene due colonne di stringa lineare, L1 e L2.

```
CREATE TABLE LINESTRING_TEST (l1 integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString);
```

La seguente istruzione INSERT inserisce due stringhe lineari uguali, le cui coordinate presentano lo stesso ordine, nelle colonne L1 e L2.

```
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)));
```

La seguente istruzione INSERT inserisce due stringhe lineari uguali, le cui coordinate non presentano lo stesso ordine, nelle colonne L1 e L2.

```
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)));
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano il modo in cui la funzione ST_Equals restituisce 1 (TRUE) indipendente dall'ordine delle coordinate. La funzione ST_OrderingEquals restituisce 0 (FALSE) se le geometrie non sono uguali e non presentano lo stesso ordine di coordinate.

```
SELECT l1d, db2gse.ST_Equals(l1,l2) equals, db2gse.ST_OrderingEquals(l1,l2)  
OrderingEquals  
FROM linestring_test
```

l1d	equals	OrderingEquals
1	1	1
2	1	0

ST_Overlaps

ST_Overlaps rileva due oggetti di geometria e restituisce 1 (TRUE) se l'intersezione dei risultati degli oggetti determinano un oggetto di geometria della stessa dimensione, ma diverso dagli oggetti di origine; in caso contrario, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Overlaps(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

L'ispettore regionale richiede un elenco dei siti di discariche a rischio nel cui raggio di cinque miglia sono situate aree protette.

La seguente istruzione CREATE TABLE crea la tabella SENSITIVE_AREAS. La tabella SENSITIVE_AREAS contiene diverse colonne che descrivono le istituzioni interessate oltre alla colonna ZONE che memorizza la geometria del poligono dell'istituzione.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

La tabella HAZARDOUS_SITES memorizza l'identità dei siti nelle colonne SITE_ID e NAME, mentre l'effettiva ubicazione geografica di ciascun sito viene memorizzata nella colonna di punto LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   db2gse.ST_Point);
```

Nella seguente istruzione SELECT, le tabelle SENSITIVE_AREAS e HAZARDOUS_SITES vengono unite dalla funzione ST_Overlaps. Questa funzione restituisce 1 (TRUE) per tutte le righe nella tabella SENSITIVE_AREAS i cui poligoni di zona superano il raggio di cinque miglia compreso nel buffer del punto di ubicazione di HAZARDOUS_SITES.

```
SELECT hs.name
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa
WHERE db2gse.ST_Overlaps (buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

Nella Figura 37 a pagina 257, l'ospedale e la scuola superano il raggio di cinque miglia dei due siti delle discariche a rischio, mentre l'asilo non lo supera.

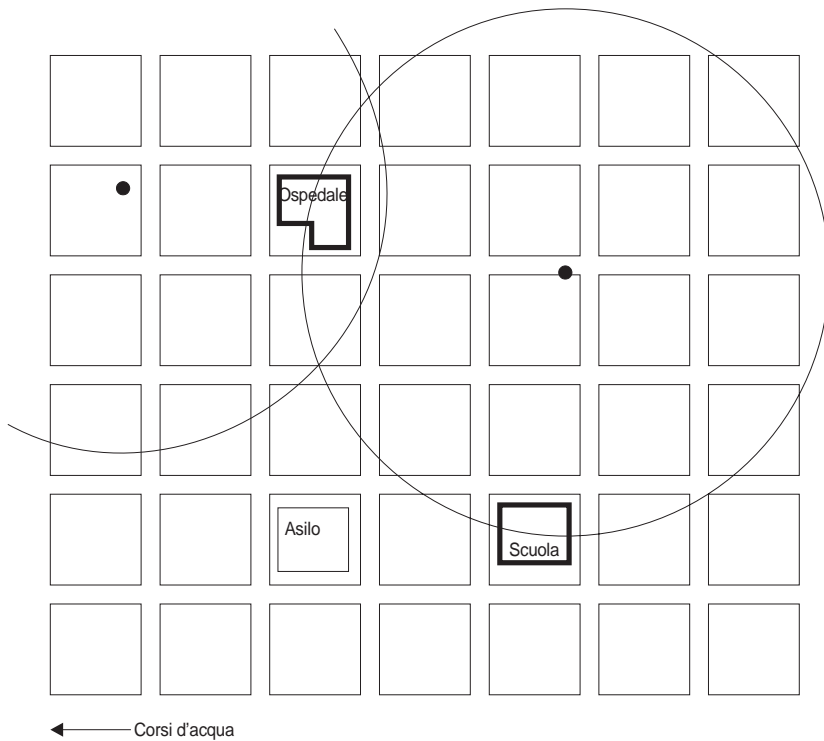


Figura 37. Utilizzare `ST_Overlaps` per determinare gli edifici che si trovano almeno parzialmente all'interno di una discarica a rischio

ST_Perimeter

ST_Perimeter restituisce il perimetro di ST_Surface.

Sintassi

```
db2gse.ST_Perimeter(s db2gse.ST_Surface)
db2gse.ST_Perimeter(ms db2gse.ST_MultiSurface)
```

Tipo di ritorno

Double

Esempi

Un ecologista sta studiando gli uccelli costieri e ha bisogno di stabilire la linea costiera per i laghi all'interno di una determinata area. I laghi vengono memorizzati come multipoligoni nella tabella WATERBODIES creata con la seguente istruzione CREATE TABLE.

```
CREATE TABLE WATERBODIES (wbid integer,
                           waterbody db2gse.ST_MultiPolygon);
```

Nella seguente istruzione SELECT, la funzione ST_Perimeter restituisce il perimetro che circonda ciascuna parte di acqua, mentre la funzione SUM aggrega i perimetri per restituirne il totale.

```
SELECT SUM(db2gse.ST_Perimeter(waterbody))
FROM waterbodies;
```

ST_PointFromText

ST_PointFromText rileva una rappresentazione WKT di tipo punto e un'identità del sistema di riferimento spaziale e restituisce un punto.

Sintassi

```
db2gse.ST_PointFromText(pointTaggedText Varchar(4000), cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella POINT_TEST, che contiene una singola colonna point, PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

Prima che l'istruzione INSERT inserisca il punto nella colonna PT1, la funzione ST_PointFromText converte le coordinate di testo del punto nel formato punto.

```
INSERT INTO POINT_TEST VALUES (  
    db2gse.ST_PointFromText ('point(10.01 20.03)',  
        db2gse.coordref()..srid(0))
```

ST_PointFromWKB

ST_PointFromWKB rileva una rappresentazione WKB di tipo punto e un'identità del sistema di riferimento spaziale per restituire un punto.

Sintassi

```
db2gse.ST_PointFromWKB(WKBPoint Blob(1M), srs SRID)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

Questo frammento di codifica compila la tabella HAZARDOUS_SITES.

I siti a rischio vengono memorizzati nella tabella HAZARDOUS_SITES creata con la seguente istruzione CREATE TABLE. La colonna LOCATION, definita come punto, memorizza un'ubicazione che rappresenta il centro geografico di ciascun sito a rischio.

```
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_Point

ST_Point restituisce ST_Point, una coordinata x, una coordinata y e un riferimento spaziale.

Sintassi

```
db2gse.ST_Point(X Double, Y Double, srs SRID)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella POINT_TEST, che contiene una singola colonna point, PT1.

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

La funzione ST_Point converte le coordinate di punto in una geometria di punto prima che l'istruzione INSERT lo inserisca nella colonna PT1.

```
INSERT INTO point_test VALUES(  
    db2gse.ST_Point(10.01,20.03, db2gse.coordref()..srid(0))  
)
```

ST_PointN

ST_PointN rileva una stringa lineare e un indice integer e restituisce un punto che rappresenta l'ennesimo vertice nel percorso della stringa lineare.

Sintassi

```
db2gse.ST_PointN(l db2gse.ST_Curve, n Integer)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella POINTN_TEST, che contiene due colonne: la colonna GID, che identifica in modo univoco ciascuna riga, e la colonna della stringa lineare LN1.

```
CREATE TABLE POINTN_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Le seguenti istruzioni INSERT inseriscono due valori di stringhe lineari. La prima stringa lineare non presenta misure o coordinate Z, mentre la seconda presenta entrambi.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondenti elencano la colonna GID ed il secondo vertice di ciascuna stringa lineare. La prima riga diviene un punto senza misura o coordinata Z e la seconda riga diviene un punto con una misura ed una coordinata Z. La funzione ST_PointN restituisce punti con una misura ed una coordinata Z, se sono presenti nella stringa lineare di origine.

```
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 rileva un poligono o un multipoligono e restituisce ST_Point.

Sintassi

```
db2gse.ST_PointOnSurface(s db2gse.ST_Surface)
db2gse.ST_PointOnSurface(ms db2gse.ST_MultiSurface)
```

Tipo di ritorno

db2gse.ST_Point

Esempi

L'ingegnere edile deve creare un punto etichetta per ciascuna piantina degli edifici.

Le piantine degli edifici vengono memorizzate nella tabella BUILDINGFOOTPRINTS creata con la seguente istruzione CREATE TABLE:

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

La funzione ST_PointOnSurface genera un punto che si trova nella superficie delle piantine degli edifici. La funzione ST_PointOnSurface restituisce un punto che viene convertito dalla funzione AsBinaryShape in una forma costituita da una stringa di caratteri da 1 megabyte per l'utilizzo da parte dell'applicazione.

```
SELECT CAST(db2gse.AsBinaryShape(db2gse.ST_PointOnSurface(footprint)) as
          blob(1m))
FROM BUILDINGFOOTPRINTS;
```

ST_PolyFromText

ST_PolyFromText rileva una rappresentazione WKT di tipo poligono e un'identità del sistema di riferimento spaziale e restituisce un poligono.

Sintassi

```
db2gse.ST_PolyFromText(polygonTaggedText Varchar(4000), cr  
db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Polygon
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella POLYGON_TEST con una singola colonna poligono.

```
CREATE TABLE POLYGON_TEST (p11 db2gse.ST_Polygon)
```

La seguente istruzione INSERT inserisce un poligono nella colonna poligono utilizzando la funzione ST_PolyFromText.

```
INSERT INTO POLYGON_TEST VALUES (1,  
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 rileva una rappresentazione WKB di tipo poligono e un'identità del sistema di riferimento spaziale per restituire un poligono.

Sintassi

db2gse.ST_PolyFromWKB(WKBPolygon Blob(1M), SRID Integer)

Tipo di ritorno

db2gse.ST_Polygon

Esempi

Questo frammento di codifica compila la tabella SENSITIVE_AREAS.

La tabella SENSITIVE_AREAS contiene diverse colonne che descrivono le istituzioni interessate oltre alla colonna ZONE che memorizza la geometria del poligono dell'istituzione.

```
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 genera ST_Polygon da ST_LineString e un'identificativo del sistema di riferimento spaziale.

Sintassi

```
db2gse.ST_Polygon(l db2gse.ST_LineString, cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Polygon
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella POLYGON_TEST, che contiene una singola colonna, P1.

```
CREATE TABLE POLYGON_TEST (p1 db2gse.ST_polygon)
```

La seguente istruzione INSERT converte un anello (una stringa lineare chiuso e semplice) in un poligono e lo inserisce nella colonna P1 utilizzando la funzione ST_LineFromText all'interno della funzione ST_Polygon.

```
INSERT INTO POLYGON_TEST VALUES (  
db2gse.ST_Polygon(db2gse.ST_LineFromText('linestring(10.01 20.03,20.94  
21.34,35.93 10.04,10.01 20.03)', db2gse.coordref()..srid(0))),  
db2gse.coordref()..srid(0))  
)
```

ST_Relate

ST_Relate confronta le due geometrie e restituisce 1 (TRUE) se le geometrie soddisfano le condizioni specificate dalla stringa della matrice modello DE-9IM; in caso contrario, viene restituito 0 (FALSE). Per le informazioni relative alle matrici modello DE-9IM, consultare “Funzioni predicato” a pagina 143.

Sintassi

```
db2gse.ST_Relate(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry,  
patternMatrix String)
```

Tipo di ritorno

Integer

Esempi

La matrice modello DE-9IM è un dispositivo per il confronto delle geometrie. Vi sono diversi tipi di matrici. Ad esempio, la matrice modello *equals* indica se vi sono due geometrie uguali.

In questo esempio, una matrice modello *equals*, mostrata nella Tabella 56, viene letta da sinistra verso destra e dall’alto verso il basso in una stringa (“T*F**FFF*”).

Tabella 56. Matrice modello *equals*

		b		
		Interno	Limite	Esterno
a	Interno	P	*	F
	Limite	*	*	F
	Esterno	F	F	*

Quindi, viene creata la tabella RELATE_TEST con la seguente istruzione CREATE TABLE.

```
CREATE TABLE RELATE_TEST (rid integer, g1 db2gse.ST_Geometry,  
g2 db2gse.ST_Geometry, g3 db2gse.ST_Geometry);
```

Le seguenti istruzioni INSERT inseriscono una classe secondaria di esempio nella tabella RELATE_TEST.

```
INSERT INTO RELATE_TEST VALUES(  
1,  
db2gse.ST_PointFromText('point (10.02 20.01)',  
db2gse.coordref()..srid(0),  
db2gse.ST_PointFromText('point (10.02 20.01)',  
db2gse.coordref()..srid(0),  
db2gse.ST_PointFromText('point (30.01 20.01)',  
db2gse.coordref()..srid(0)  
)
```

La seguente istruzione SELECT ed la serie di risultati corrispondente elencano il nome della classe secondaria memorizzato nella colonna GEOTYPE con la dimensione corrispondente a geotype.

```
SELECT rid, relate(g1,g2) equals, relate(g1,g3) not_equals
FROM relate_test
```

RID	equals	not_equals
1	1	0

1 record(s) selected.

ST_SRID

ST_SRID rileva un oggetto di geometria e ne restituisce l'identità del sistema di riferimento spaziale.

Sintassi

```
db2gse.ST_SRID(g1 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Durante l'installazione di DB2 Spatial Extender viene creata la tabella SPATIAL_REFERENCES. Quando viene creata una geometria, SRID della geometria viene immesso nella tabella SPATIAL_REFERENCES. La funzione ST_SRID restituisce il valore di tale immissione.

Ad esempio, un tipo di geometria viene utilizzato in un'istruzione CREATE TABLE:

```
CREATE TABLE SRID_TEST(g1 db2gse.ST_Geometry)
```

Nella seguente istruzione INSERT, una geometria di punto ubicata alla coordinata 10.01,50.76 viene inserita nella colonna della geometria G1. Quando la geometria di punto è stata creata dalla funzione ST_PointFromText, vi è stato assegnato il valore srid di 1.

```
INSERT INTO SRID_TEST
VALUES (db2gse.ST_PointFromText('point(10.01 50.76)',
    db2gse.coordref()..srid(0)))
```

La funzione ST_SRID restituisce l'identità del sistema di riferimento spaziale della geometria appena immessa, come illustrato dalla seguente istruzione SELECT e dalla serie di risultati corrispondente.

```
SELECT db2gse.ST_SRID(g1) FROM SRID_TEST
```

```
g1
-----
1
```

ST_StartPoint

ST_StartPoint rileva una stringa lineare e restituisce il primo punto della stringa lineare.

Sintassi

```
db2gse.ST_StartPoint(c db2gse.ST_Curve)
```

Tipo di ritorno

```
db2gse.ST_Point
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella STARTPOINT_TEST. STARTPOINT_TEST ha due colonne: la colonna integer GID, che identifica le righe della tabella in modo univoco, e la colonna della stringa lineare LN1.

```
CREATE TABLE STARTPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

Le seguenti istruzioni INSERT inseriscono le stringhe lineari nella colonna LN1. La prima stringa lineare non presenta misure o coordinate Z, mentre la seconda le presenta entrambi.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente mostrano il modo in cui la funzione ST_StartPoint estrae il primo punto di ciascuna stringa lineare. La funzione ST_AsText converte il punto nel relativo formato di testo. Il primo punto nell'elenco non presenta una misura e una coordinata Z, mentre il secondo punto presenta entrambi, come la stringa lineare di origine.

```
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 rileva due oggetti di geometria e restituisce un oggetto di geometria che è la differenza simmetrica degli oggetti di origine.

Sintassi

```
db2gse.ST_SymmetricDiff(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

L'ispettore regionale deve determinare le aree protette e il raggio di cinque miglia delle zone a rischio che non viene intersecato.

La seguente istruzione CREATE TABLE crea la tabella SENSITIVE_AREAS, contenente diverse colonne che descrivono le istituzioni interessate. La tabella SENSITIVE_AREAS contiene inoltre la colonna ZONE, in cui è memorizzata la geometria del poligono dell'istituzione.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

La seguente istruzione CREATE TABLE crea la tabella HAZARDOUS_SITES, che memorizza l'identità dei siti nelle colonne SITE_ID e NAME, mentre l'effettiva ubicazione geografica di ciascun sito viene memorizzata nella colonna di punto LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                               name       varchar(128),
                               location   point);
```

La funzione ST_Buffer genera a un buffer di cinque miglia che circonda le ubicazioni delle discariche a rischio. La funzione ST_SymmetricDiff genera poligoni dall'intersezione dei poligoni delle discariche a rischio incluse nel buffer e aree protette. La funzione ST_Area restituisce l'area dei poligoni dell'intersezione per ciascuna zona a rischio.

```
SELECT sa.name, hs.name,
       db2gse.ST_Area(db2gse.ST_SymmetricDiff (db2gse.ST_Buffer(hs.location,
(5 * 5280)),sa.zone))
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa
```

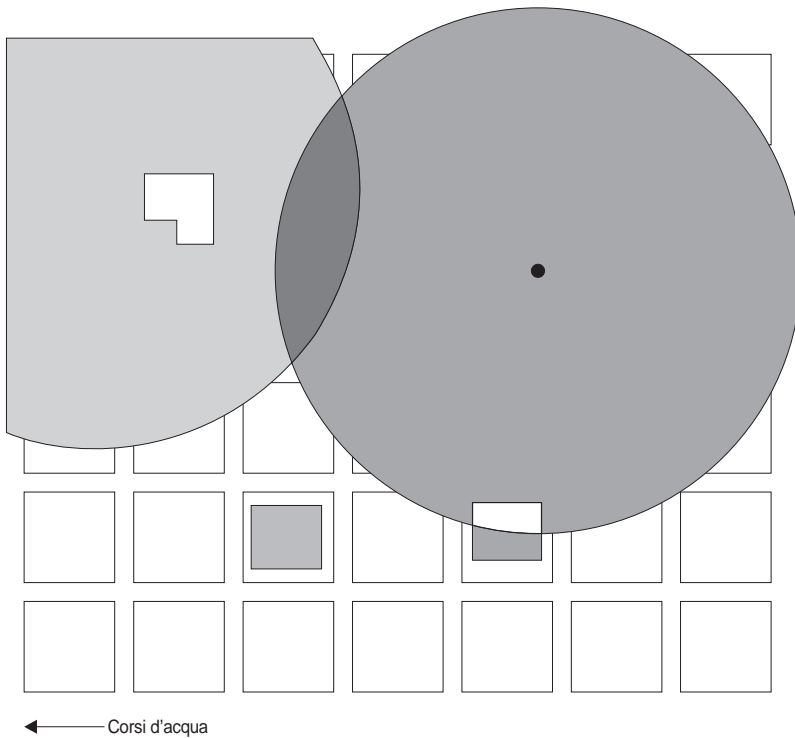


Figura 38. Utilizzare ST_SymmetricDiff per determinare le aree di discariche a rischio che non contengono aree protette (edifici abitati)

Nella Figura 38, la differenza simmetrica delle discariche a rischio e delle aree protette determina la sottrazione delle aree intersecate.

ST_Touches

ST_Touches restituisce 1 (TRUE) se nessuno dei punti comuni alle geometrie interseca gli interni di entrambe le geometrie; in caso contrario, restituisce 0 (FALSE). Almeno una geometria deve essere una stringa lineare, un poligono, una multistringa lineare o un multipoligono.

Sintassi

```
db2gse.ST_Touches(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Il tecnico GIS deve fornire un elenco di tutte le reti fognarie le cui estremità intersecano un'altra rete fognaria.

La seguente istruzione CREATE TABLE crea la tabella SEWERLINES, che contiene tre colonne. La prima colonna, SEWER_ID, identifica ciascuna rete fognaria in modo univoco. La seconda colonna, CLASS, di tipo integer identifica il tipo di rete fognaria, generalmente associata alla capacità. La terza colonna, SEWER, di tipo stringa lineare memorizza la geometria della rete fognaria.

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer, sewer  
db2gse.ST_LineString);
```

La seguente istruzione SELECT restituisce un elenco ordinato di SEWER_IDS adiacenti.

```
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 assegna una geometria a un sistema di riferimento spaziale diverso da quello correntemente assegnato alla geometria.

Sintassi

```
db2gse.ST_Transform(g db2gse.ST_Geometry, cr db2gse.coordref)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella TRANSFORM_TEST, che contiene due colonne di stringa lineare, L1 e L2.

```
CREATE TABLE TRANSFORM_TEST (tid integer, l1 db2gse.ST_LineString,  
l2 db2gse.ST_LineString)
```

La seguente istruzione INSERT inserisce una stringa lineare in l1 con un SRID di 102.

```
INSERT INTO TRANSFORM_TEST VALUES (1, db2gse.ST_LineFromText('linestring  
(10.01 40.43, 92.32 29.89)',  
db2gse.coordref()..srid(102)),NULL)
```

La funzione ST_Transform converte la stringa lineare di L1 dal riferimento di coordinata assegnato a SRID 102 nel riferimento di coordinata assegnato a SRID 105. La seguente istruzione UPDATE memorizza la stringa lineare trasformata nella colonna l2.

```
UPDATE TRANSFORM_TEST SET l2 = db2gse.ST_Transform(l1,  
db2gse.coordref()..srid(105))
```

ST_Union

ST_Union rileva due oggetti di geometria e restituisce un oggetto di geometria che è l'unione degli oggetti di origine.

Sintassi

```
db2gse.ST_Union(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella SENSITIVE_AREAS, contenente diverse colonne che descrivono le istituzioni interessate. La tabella SENSITIVE_AREAS contiene inoltre la colonna ZONE, in cui è memorizzata la geometria del poligono dell'istituzione.

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

La seguente istruzione CREATE TABLE crea la tabella HAZARDOUS_SITES, che memorizza l'identità dei siti nelle colonne SITE_ID e NAME. L'effettiva ubicazione geografica di ciascun sito viene memorizzata nella colonna di punto LOCATION.

```
CREATE TABLE HAZARDOUS_SITES (site_id integer, name varchar(128),
                               location db2gse.ST_Point);
```

La seguente istruzione SELECT utilizza la funzione ST_Buffer per generare un buffer di cinque miglia che circonda le ubicazioni delle discariche a rischio. La funzione ST_Union genera poligoni dall'unione dei poligoni delle discariche a rischio e delle aree protette. La funzione ST_Area restituisce l'unione dell'area dei poligoni.

```
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 rileva due oggetti di geometria e restituisce 1 (TRUE) se il primo oggetto si trova nel secondo; in caso contrario, restituisce 0 (FALSE).

Sintassi

```
db2gse.ST_Within(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

Tipo di ritorno

Integer

Esempi

Nel seguente esempio sono riportate due tabelle. La prima tabella, BUILDINGFOOTPRINTS, contiene le piantine degli edifici della città. La seconda tabella, LOTS, contiene i lotti della città. L'ingegnere desidera assicurarsi che tutte le piantine siano situate correttamente all'interno dei relativi lotti.

In entrambe le tabelle, il tipo di dati multipoligono memorizza la geometria delle piantine degli edifici e dei lotti. Il programma di progettazione del database ha selezionato multipoligoni per entrambe le funzioni, poiché i lotti possono risultare separati da condizioni geografiche naturali, ad esempio da un fiume, e le piantine degli edifici possono essere costituite da più edifici.

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                     lot_id      integer,
                                     footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (  lot_id integer, lot db2gse.ST_MultiPolygon );
```

Utilizzando la seguente istruzione SELECT, l'ingegnere edile seleziona innanzitutto gli edifici non completamente contenuti all'interno di un lotto.

```
SELECT building_id
   FROM BUILDINGFOOTPRINTS, LOTS
  WHERE db2gse.ST_Within(footprint,lot) = 0;
```

Sebbene la prima interrogazione restituirà un elenco di tutti i BUILDING_ID le cui piantine sono situate all'esterno di un poligono di lotto, questa non determinerà se agli altri edifici è assegnato l'ID di lotto corretto. La seconda istruzione SELECT esegue un controllo dell'integrità dei dati nella colonna LOT_ID della tabella BUILDINGFOOTPRINTS.

```
SELECT bf.building_id "building id",
       bf.lot_id      "buildings lot_id",
       LOTS.lot_id    "LOTS lot_id"
   FROM BUILDINGFOOTPRINTS bf, LOTS
  WHERE db2gse.ST_Within(footprint,lot) = 1 AND
        LOTS.lot_id <> bf.lot_id;
```

ST_WKBTToSQL

ST_WKBTToSQL costruisce un valore ST_Geometry in base alla rappresentazione WKB. Viene utilizzato automaticamente un valore SRID di 0.

Sintassi

```
db2gse.ST_WKBTToSQL(WKBGeometry Blob(1M))
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella LOTS, che contiene due colonne: la colonna LOT_ID, che identifica ciascun lotto in modo univoco, e la colonna LOT del multipoligono, che contiene la geometria di ciascun lotto.

```
CREATE TABLE lots (lot_id integer,  
                   lot      db2gse.ST_MultiPolygon);
```

Il seguente frammento di codifica C contiene funzioni ODBC integrate alle funzioni SQL DB2 Spatial Extender che inseriscono i dati nella tabella LOTS.

La funzione ST_WKBTToSQL converte le rappresentazioni WKB nella geometria DB2 Spatial Extender. L'intera istruzione INSERT viene copiata in una stringa wkb_sql char. L'istruzione INSERT contiene i contrassegni di parametro per accettare i dati LOT_ID e LOT in modo dinamico.

```
/* 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_WKBTToSQL(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 costruisce il valore ST_Geometry in base alla relativa rappresentazione WKT. Viene utilizzato automaticamente un valore SRID di 0.

Sintassi

```
db2gse.ST_WKTTToSQL(geometryTaggedText Varchar(4000))
```

Tipo di ritorno

```
db2gse.ST_Geometry
```

Esempi

La seguente istruzione CREATE TABLE crea la tabella GEOMETRY_TEST, che contiene due colonne: la colonna GID di tipo integer, che identifica ciascuna riga in modo univoco, e la colonna G1, che memorizza la geometria.

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

Le seguenti istruzioni INSERT inseriscono i dati nelle colonne GID e G1 della tabella GEOMETRY_TEST. La funzione ST_WKTTToSQL converte la rappresentazione di testo di ciascuna geometria nella corrispondente classe secondaria DB2 Spatial Extender istanziabile.

```
INSERT INTO GEOMETRY_TEST VALUES(
1, db2gse.ST_WKTTToSQL ('point (10.02 20.01)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
2, db2gse.ST_WKTTToSQL('linestring (10.01 20.01, 10.01 30.01, 10.01 40.01)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
3, db2gse.ST_WKTTToSQL('polygon ((10.02 20.01, 11.92 35.64, 25.02 34.15,
19.15 33.94, 10.02 20.01))')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
4, db2gse.ST_WKTTToSQL('multipoint (10.02 20.01,10.32 23.98,11.92 25.64)')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
5, db2gse.ST_WKTTToSQL('multilinestring ((10.02 20.01,      10.32 23.98,
11.92 25.64),(9.55 23.75,15.36 30.11))')
)
```

```
INSERT INTO GEOMETRY_TEST VALUES(
6, db2gse.ST_WKTTToSQL('multipolygon (((10.02 20.01, 11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
((51.71 21.73, 73.36 27.04, 71.52 32.87, 52.43 31.90, 51.71 21.73)))')
)
```

ST_X

ST_X rileva un punto e restituisce la relativa coordinata X.

Sintassi

ST_X(p ST_Point)

Tipo di ritorno

Double

Esempi

La seguente istruzione CREATE TABLE crea la tabella X_TEST, che contiene due colonne: la colonna GID, che identifica la riga in modo univoco, e la colonna di punto PT1.

```
CREATE TABLE X_TEST (gid integer, pt1 db2gse.ST_Point)
```

Le seguenti istruzioni INSERT inseriscono due righe. Una è un punto senza una misura o una coordinata Z. L'altra colonna presenta una misura e una coordinata Z.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente elencano la colonna GID e la coordinata Doppio X dei punti.

```
SELECT gid, db2gse.ST_X(pt1) "The X coordinate" FROM X_TEST
```

```
GID          The X coordinate  
-----  
1  +1.002000000000000E+001  
2  +1.002000000000000E+001
```

2 record(s) selected.

ST_Y

ST_Y rileva un punto e restituisce la relativa coordinata Y.

Sintassi

```
db2gse.ST_Y(p db2gse.ST_Point)
```

Tipo di ritorno

Double

Esempi

La seguente tabella CREATE TABLE crea la tabella Y_TEST, che contiene due colonne: la colonna GID, che identifica la riga in modo univoco, e la colonna di punto PT1.

```
CREATE TABLE Y_TEST (gid integer, pt1 db2gse.ST_Point)
```

Le istruzioni INSERT inseriscono due righe. Una è un punto senza una misura o una coordinata Z. L'altra colonna presenta una misura e una coordinata Z.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente elencano la colonna GID e la coordinata Doppio Y dei punti.

```
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 rileva un punto e restituisce la relativa coordinata Z.

Sintassi

Z(p db2gse.ST_Point)

Tipo di ritorno

Double

Esempi

La seguente istruzione CREATE TABLE crea la tabella Z_TEST, che contiene due colonne: la colonna GID che identifica la riga in modo univoco, e la colonna di punto PT1.

```
CREATE TABLE Z_TEST (gid integer, pt1 db2gse.ST_Point)
```

Le seguenti istruzioni INSERT inseriscono due righe. Una è un punto senza una misura o una coordinata Z. L'altra colonna presenta una misura e una coordinata Z.

```
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)))
```

La seguente istruzione SELECT e la serie di risultati corrispondente elencano la colonna GID e la coordinata Doppio Z dei punti. La prima riga è NULL, poiché il punto non presenta una coordinata Z.

```
SELECT gid, db2gse.Z(pt1) "The Z coordinate" FROM Z_TEST
```

GID	The Z coordinate
1	-
2	+5.00000000000000E+000

2 record(s) selected.

Capitolo 15. Sistemi di coordinate

Questo capitolo fornisce informazioni relative al sistema di riferimento spaziale (SRS) e i valori delle coordinate utilizzate per interpretare i dati spaziali.

- “Panoramica dei sistemi di coordinate”
- “Unità lineari supportate” a pagina 287
- “Unità angolari supportate” a pagina 288
- “Sferoidi supportati” a pagina 288
- “Dati geodetici supportati” a pagina 290
- “Meridiani principali supportati” a pagina 292
- “Proiezioni della carta supportate” a pagina 292
- “Proiezioni coniche” a pagina 293
- “Proiezioni planari e azimutali” a pagina 293
- “Parametri di proiezione della carta” a pagina 293

Panoramica dei sistemi di coordinate

La rappresentazione WKT dei sistemi di riferimento spaziale fornisce una rappresentazione standard delle relative informazioni. Il modello delle definizioni della rappresentazione WKT si basa sul modello dei dati del sistema di coordinate POSC/EPSC.

Un sistema di riferimento spaziale rappresenta un sistema di coordinate geografiche (latitudine-longitudine), di proiezione (X,Y) o geocentriche (X,Y,Z). Il sistema di coordinate è composto da più oggetti. Ciascun oggetto presenta una parola chiave in lettere maiuscole (ad esempio, DATUM o UNIT) seguita dai parametri di definizione delimitati da virgola dell'oggetto tra parentesi. Alcuni oggetti comprendono altri oggetti e presentano quindi una struttura nidificata.

Nota: L'implementazione può eseguire la sostituzione delle parentesi standard () con le parentesi quadre [] ed è necessario che sia abilitata per la lettura di entrambe le forme di parentesi.

La definizione EBNF (Extended Backus Naur Form) relativa alla rappresentazione della stringa di un sistema di coordinate che utilizza parentesi quadre è la seguente (fare riferimento alle note sopra riportate relative all'uso delle parentesi):

```

<coordinate system> = <projected cs>
| <geographic cs> | <geocentric cs>
<projected cs> = PROJCS["<name>", <geographic cs>, <projection>, {<parameter>,*
    <linear unit>}]
<projection> = PROJECTION["<name>"]
<parameter> = PARAMETER["<name>", <value>]
<value> = <number>

```

Un sistema di coordinate della serie di dati è identificata dalla parola chiave PROJCS se i dati sono contenuti nelle coordinate di conversione (da GEOGCS per i dati delle coordinate geografiche o da GEOCCS per i dati delle coordinate geocentriche). La parola chiave PROJCS è seguita da tutte le "parti" che definiscono il sistema di coordinate di proiezione. La prima parte di un oggetto è sempre il nome. Il nome del sistema di coordinate di proiezione è seguito da vari oggetti: il sistema di coordinate geografiche, la proiezione corrispondente, uno o più parametri e l'unità di misura lineare. Tutti i sistemi di coordinate di proiezione si basano su un sistema di coordinate geografiche, questa sezione descrive prima le parti specifiche di un sistema di coordinate di proiezione. Ad esempio, la zona 10N di UTM in NAD83 è definita come segue:

```

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]]

```

Il nome e vari oggetti definiscono l'oggetto del sistema di coordinate geografiche per le seguenti voci: datum, prime meridian e angular unit.

```

<geographic cs> = GEOGCS["<name>", <datum>, <prime meridian>, <angular unit>]
<datum> = DATUM["<name>", <spheroid>]
<spheroid> = SPHEROID["<name>", <semi-major axis>, <inverse flattening>]
<semi-major axis> = <number>
    (semi-major axis is measured in meters and must be > 0.)
<inverse flattening> = <number>
<prime meridian> = PRIMEM["<name>", <longitude>]
<longitude> = <number>

```

La stringa del sistema di coordinate geografiche per la zona 10 di UTM 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]]

```

L'oggetto UNIT può rappresentare le unità di misura lineari e angolari:

```

<angular unit> = <unit>
<linear unit> = <unit>
<unit> = UNIT["<name>", <conversion factor>]
<conversion factor> = <number>

```

Il fattore di conversione specifica il numero di metri (per l'unità di misura lineare) o il numero di radianti (per un'unità di misura angolare) ed è necessario che sia superiore a zero.

La rappresentazione della stringa completa della zona 10N di UTM è la seguente:

```

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]]

```

Un sistema di coordinate geocentriche è simile a un sistema di coordinate geografiche:

```

<geocentric cs> = GEOCCS["<name>", <datum>, <prime meridian>, <linear unit>]

```

Unità lineari supportate

Tabella 57. Unità lineari supportate

Unità	Fattore di conversione
Metri	1,0
Piedi (Internazionale)	0,3048
Piedi (Stati Uniti)	12/39,37
Piedi (americano modificato)	12,0004584/39,37
Piedi (di Clarke)	12/39,370432
Piedi (indiano)	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

Tabella 57. Unità lineari supportate (Continua)

Unità	Fattore di conversione
Fathom	1.8288
Nautical Mile	1852.0

Unità angolari supportate

Tabella 58. Unità angolari supportate

Unità	Fattore di conversione
Radiante	1,0
Grado decimale	$p/180$
Minuto decimale	$(p/180)/60$
Secondo decimale	$(p/180)/36000$
Gon	$p/200$
Grad	$p/200$

Sferoidi supportati

Tabella 59. Sferoidi supportati

Nome	Asse semimaggiore	Livellamento inverso
Airy	6377563.396	299.3249646
Airy modificato	6377340.189	299.3249646
Australian	6378160	298.25
Bessel	6377397.155	299.1528128
Bessel modificato	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

Tabella 59. Sferoidi supportati (Continua)

Nome	Asse semimaggiore	Livellamento inverso
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
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
Sphere (radius = 1.0)	1	0
Sphere (radius = 6371000 m)	6371000	0
Sphere (radius =6370997 m)	6370997	0
Struve 1860	6378297	294.73
Walbeck	6376896	302.78
War Office	6378300.583	296
WGS 1960	6378165	298.3

Tabella 59. Sferoidi supportati (Continua)

Nome	Asse semimaggiore	Livellamento inverso
WGS 1966	6378145	298.25
WGS 1972	6378135	298.26
WGS 1984	6378137	298.257223563

Dati geodetici supportati

Tabella 60. Dati geodetici supportati

Adindan	Lisbon
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-Geographische Institute
Barbados	Mhast
Batavia	Minna
Beduaram	Monte Mario
Beijing 1954	M'poraloko
Reseau National Belge 1950	NAD Michigan
Reseau National Belge 1972	North American Datum 1927
Bermuda 1957	North American Datum 1983
Bern 1898	Nahrwan 1967
Bern 1938	Naparima 1972
Bogota	Nord de Guerre
Bukit Rimpah	NGO 1948
Camacupa	Nord Sahara 1959
Campo Inchauspe	NSWC 9Z-2
Cape	Nouvelle Triangulation Francaise

Tabella 60. Dati geodetici supportati (Continua)

Carthage	New Zealand Geodetic Datum 1949
Chua	OS (SN) 1980
Conakry 1905	OSGB 1936
Corrego Alegre	OSGB 1970 (SN)
Cote d'Ivoire	Padang 1884
Datum 73	Palestine 1923
Deir ez Zor	Pointe Noire
Deutsche Hauptdreiecksnetz	Provisional South American Datum 1956
Douala	Pulkovo 1942
European Datum 1950	Qatar
European Datum 1987	Qatar 1948
Egypt 1907	Qornoq
European Reference System 1989	RT38
Fahud	South American Datum 1969
Gandajika 1970	Sapper Hill 1943
Garoua	Schwarzeck
Geocentric Datum of Australia 1994	Segora
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

Tabella 60. Dati geodetici supportati (Continua)

Leigon	Yoff
Liberia 1964	Zanderij

Meridiani principali supportati

Tabella 61. Meridiani principali supportati

Ubicazione	Coordinate
Greenwich	0° 0' 0"
Bern	7° 26' 22.5" E
Bogota	74° 4' 51.3" W
Brussel	4° 22' 4.71" E
Ferro	17° 40' 0" W
Jakarta	106° 48' 27.79" E
Lisbona	9° 7' 54.862" W
Madrid	3° 41' 16.58" W
Parigi	2° 20' 14.025" E
Roma	12° 27' 8.4" E
Stoccolma	18° 3' 29" E

Proiezioni della carta supportate

Tabella 62. Proiezioni della carta supportate

Proiezioni cilindriche	Proiezioni pseudocilindriche
Behrmann	Parabolica Craster
Cassini	Eckert I
Equiareale cilindrico	Eckert II
Equirettangolare	Eckert III
Stereografica di Gall	Eckert IV
Gauss-Kruger	Eckert V
Mercatore	Eckert VI
Cilindrica Miller	Quartico polare piatto McBryde-Thomas
Obliquo	Mercatore (Hotine) Mollweide
Plate-Carée	Robinson

Tabella 62. Proiezioni della carta supportate (Continua)

Proiezioni cilindriche	Proiezioni pseudocilindriche
Times	Sinusoidale (Sansom-Flamsteed)
Mercator trasversale	Winkel I

Proiezioni coniche

Tabella 63. Proiezioni coniche

Equiareale conico Albers	Trimetrico Chamberlin
Conica conforme obliqua bipolare	Due punti equidistanti
Bonne	Equiareale Hammer-Aitoff
Conica equidistante	Van der Grinten I
Conica conforme Lambert	Varie
Policonica	Alaska serie E
Conica semplice	Griglia Alaska (Stereografica modificata da Snyder)

Proiezioni planari e azimutali

- Equidistante azimutale
- Prospettica laterale verticale generale
- Gnomonica
- Equiareale azimutale di Lambert
- Ortografica
- Stereografica polare
- Stereografica

Parametri di proiezione della carta

Tabella 64. Parametri di proiezione della carta

Parametro	Descrizione
meridiano_principale	La linea della longitudine scelta come origine delle coordinate x.
fattore_di scala	Utilizzato generalmente per ridurre la distorsione in una proiezione di carta.

Tabella 64. Parametri di proiezione della carta (Continua)

Parametro	Descrizione
parallelo_standard_1	Una linea di latitudine che non presenta distorsioni. Utilizzata anche per la "latitudine di una scala reale".
parallela_standard_2	Una linea di latitudine che non presenta distorsioni.
longitudine_del_centro	La longitudine che definisce il punto centrale della proiezione.
latitudine_del_centro	La latitudine che definisce il punto centrale della proiezione.
latitudine_di_origine	La latitudine scelta come origine delle coordinate y.
falso_spostamento_a_est	Aggiunto alle coordinate x. Utilizzato per fornire valori positivi.
falso_spostamento_a_nord	Aggiunto alle coordinate y. Utilizzato per fornire valori positivi.
azimut	L'angolo orientale rispetto al nord che definisce la linea centrale di una proiezione obliqua.
longitudine_del_punto_1	La longitudine del primo punto necessario per una proiezione.
latitudine_del_punto_1	La latitudine del primo punto necessario per una proiezione.
longitudine_del_punto_2	La longitudine del secondo punto necessario per una proiezione.
latitudine_del_punto_2	La latitudine del secondo punto necessario per una proiezione.
longitudine_del_punto_3	La longitudine del terzo punto necessario per una proiezione.
latitudine_del_punto_3	La latitudine del terzo punto necessario per una proiezione.
numero_landsat	Il numero di un satellite Landsat.
numero_percorso	Il numero del percorso orbitale di un particolare satellite.
altezza_del_punto_di_prospettiva	L'altezza sulla terra del punto di prospettiva della proiezione.
fipszone	Numero di zona State Plane Coordinate System.
zona	Numero di zona UTM.

Capitolo 16. Formati file dei dati spaziali

Questo capitolo illustra le rappresentazioni note di DB2 Spatial Extender. Le rappresentazioni sono descritte come *note* in quanto vengono fornite da ESRI e non sono specifiche di DB2 Spatial Extender. Per l'importazione e l'esportazione dei dati spaziali sono necessari i seguenti tre tipi di valori spaziali:

- Le rappresentazioni WKT (well-known text) OGC (Open GIS Consortium)
- Le rappresentazioni WKB (well-known binary)
- Le rappresentazioni in formato ESRI

Le rappresentazioni WKT OGC

DB2 Spatial Extender dispone di varie funzioni che generano geometrie dalle descrizioni di testo:

ST_GeomFromText

Crea una geometria dalla rappresentazione di un testo di qualsiasi tipo di geometria.

ST_PointFromText

Crea un punto dalla rappresentazione di testo di un punto.

ST_LineFromText

Crea una stringa lineare dalla rappresentazione di testo di una stringa lineare.

ST_PolyFromText

Crea un poligono dalla rappresentazione di testo di un poligono.

ST_MPointFromText

Crea un multipunto dalla rappresentazione di testo di un multipunto.

ST_MLineFromText

Crea una multistringa lineare dalla rappresentazione di testo di una multistringa lineare.

ST_MPolyFromText

Crea un multipoligono dalla rappresentazione di testo di un multipoligono.

La rappresentazione di testo è una stringa con formato testo ASCII che consente lo scambio di geometrie in formato testo ASCII. E' possibile utilizzare queste funzioni in un programma con linguaggio di terza o quarta generazione (3GL o 4GL) in quanto non richiedono le definizioni delle

strutture di programma speciale. La funzione ST_AsText converte una geometria esistente in una rappresentazione di testo.

Ciascun tipo di geometria presenta una rappresentazione WKT che è possibile utilizzare per creare nuove istanze di quel tipo e per convertire le istanze esistenti nel formato testo per la visualizzazione alfanumerica.

La rappresentazione WKT di una geometria è definita come riportato di seguito: {}* denota 0 o più ripetizioni dei token compreso tra parentesi; le parentesi non vengono visualizzate nell'elenco token di output.

```
<Geometry Tagged Text> :=
| <Point Tagged Text>
| <LineString Tagged Text>
| <Polygon Tagged Text>
| <MultiPoint Tagged Text>
| <MultiLineString Tagged Text>
| <MultiPolygon Tagged Text>

<Point Tagged Text> :=
POINT <Point Text>

<LineString Tagged Text> :=
LINESTRING <LineString Text>

<Polygon Tagged Text> :=
POLYGON <Polygon Text>

<MultiPoint Tagged Text> :=
MULTIPOINT <Multipoint Text>

<MultiLineString Tagged Text> :=
MULTILINESTRING <MultiLineString Text>

<MultiPolygon Tagged Text> :=
MULTIPOLYGON <MultiPolygon Text>

<Point Text> := EMPTY
| <Point>
| Z <PointZ>
| M <PointM>
| ZM <PointZM>

<Point> := <x> <y>
<x> := double precision literal
<y> := double precision literal
<PointZ> := <x> <y> <z>
<x> := double precision literal
<y> := double precision literal
<z> := double precision literal
<PointM> := <x> <y> <m>
<x> := double precision literal
<y> := double precision literal
<m> := double precision literal
```

```

<PointZM> := <x> <y> <z> <m>
<x> := double precision literal
<y> := double precision literal
<z> := double precision literal
<m> := double precision literal

<LineString Text> := EMPTY
| ( <Point Text > {, <Point Text> }* )
| Z ( <PointZ Text > {, <PointZ Text> }* )
| M ( <PointM Text > {, <PointM Text> }* )
| ZM ( <PointZM Text > {, <PointZM Text> }* )

<Polygon Text> := EMPTY
| ( <LineString Text > {,< LineString Text > }* )

<Multipoint Text> := EMPTY
| ( <Point Text > {, <Point Text > }* )

<MultiLineString Text> := EMPTY
| ( <LineString Text > {,< LineString Text>}* )

<MultiPolygon Text> := EMPTY
| ( < Polygon Text > {, < Polygon Text > }* )

```

La sintassi della funzione di base è:

```
function (<text description>,<SRID>)
```

L'identificativo di riferimento spaziale (SRID) e la chiave primaria della tabella SPATIAL_REFERENCES identifica il sistema di riferimento spaziale della geometria memorizzata nella tabella SPATIAL_REFERENCES. Prima di inserire una geometria in una colonna spaziale, è necessario che il relativo SRID corrisponda all'SRID della colonna spaziale.

La descrizione di testo è composta da tre componenti di base compresi tra singoli apici, ad esempio:

```
<'tipo di geometria'>
['tipo di coordinate'] ['elenco di coordinate']
```

dove:

tipo di geometria

Comprende uno dei seguenti tipi: punto, stringa lineare, poligono, multipunto, multistringa lineare o multipoligono.

tipo di coordinate

Specifica se la geometria presenta coordinate Z o misure. Lasciare vuoto questo campo se la geometria non presenta tali valori. In caso contrario, impostare il tipo di coordinate su Z per le geometrie che contengono coordinate Z, M per le geometrie con misure e ZM per le geometrie che presentano entrambi i valori.

elenco di coordinate

Definisce i vertici della geometria. Gli elenchi di coordinate sono delimitati da virgole e racchiusi tra parentesi. Le geometrie con componenti multipli richiedono gruppi di parentesi per racchiudere ciascuna parte del componente. Se la geometria è vuota, la parola chiave EMPTY sostituisce la coordinata.

Tabella 65 illustra un elenco completo degli esempi di tutte le rappresentazioni di testo possibili.

Tabella 65. Tipi di geometria e delle relative rappresentazioni di testo.

Tipo di geometria	Descrizione del testo	Commento
punto	point empty	punto vuoto
punto	point z empty	punto vuoto con coordinata z
punto	point m empty	punto vuoto con misura
punto	point zm empty	punto vuoto con coordinata z e misura
punto	point (10.05 10.28)	punto
punto	point z (10.05 10.28 2.51)	punto con coordinata z
punto	point m (10.05 10.28 4.72)	punto con misura
punto	point zm (10.05 10.28 2.51 4.72)	punto con coordinata z e misura
stringa lineare	linestring empty	stringa lineare vuota
stringa lineare	linestring z empty	stringa lineare vuota con coordinate z
stringa lineare	linestring m empty	stringa lineare vuota con misure
stringa lineare	linestring zm empty	stringa lineare vuota con coordinate z e misure
stringa lineare	linestring (10.05 10.28 , 20.95 20.89)	stringa lineare
stringa lineare	linestring z (10.05 10.28 3.09, 20.95 31.98 4.72, 21.98 29.80 3.51)	stringa lineare con coordinate z
stringa lineare	linestring m (10.05 10.28 5.84, 20.95 31.98 9.01, 21.98 29.80 12.84)	stringa lineare con misure
stringa lineare	linestring zm ()	stringa lineare con coordinate z e misure
poligono	polygon empty	poligono vuoto

Tabella 65. Tipi di geometria e delle relative rappresentazioni di testo. (Continua)

Tipo di geometria	Descrizione del testo	Commento
poligono	polygon z empty	poligono vuoto con coordinate z
poligono	polygon m empty	poligono vuoto con misure
poligono	polygon zm empty	poligono vuoto con coordinate z e misure
poligono	polygon ((10 10, 10 20, 20 20, 20 15, 10 10))	poligono
poligono	polygon z (())	poligono con coordinate z
poligono	polygon m (())	poligono con misure
poligono	polygon zm (())	poligono con coordinate z e misure
multipunto	multipoint empty	multipunto vuoto
multipunto	multipoint z empty	multipunto vuoto con coordinate z
multipunto	multipoint m empty	multipunto vuoto con misure
multipunto	multipoint zm empty	multipunto vuoto con coordinate z e misure
multipunto	multipoint empty	multipunto vuoto
multipunto	multipoint (10 10, 20 20)	multipunto con due punti
multipunto	multipoint z (10 10 2, 20 20 3)	multipunto con coordinate z
multipunto	multipoint m (10 10 4, 20 20 5)	multipunto con misure
multipunto	multipoint zm (10 10 2 4, 20 20 3 5)	multipunto con coordinate z e misure
multistringa lineare	multilinestring empty	multistringa lineare vuota
multistringa lineare	multilinestring z empty	multistringa lineare vuota con coordinate z
multistringa lineare	multilinestring m empty	multistringa lineare con misure
multistringa lineare	multilinestring zm empty	multistringa lineare con coordinate z e misure
multistringa lineare	multilinestring (())	multistringa lineare
multistringa lineare	multilinestring z (())	multistringa lineare con coordinate z

Tabella 65. Tipi di geometria e delle relative rappresentazioni di testo. (Continua)

Tipo di geometria	Descrizione del testo	Commento
multistringa lineare	multilinestring m (())	multistringa lineare con misure
multistringa lineare	multilinestring zm (())	multistringa lineare con coordinate z e misure
multipoligono	multipolygon empty	multipoligono vuoto
multipoligono	multipolygon z empty	multipoligono vuoto con coordinate z
multipoligono	multipolygon m empty	multipoligono vuoto con misure
multipoligono	multipolygon z	multipoligono vuoto con coordinate z e misure
multipoligono	multipolygon ((()))	multipoligono
multipoligono	multipolygon z ((()))	multipoligono con coordinate z
multipoligono	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)))	multipoligono con misure
multipoligono	multipolygon zm ((()))	multipoligono con coordinate z e misure

Le rappresentazioni WKB (well-known binary) OGC

DB2 Spatial Extender dispone di varie funzioni che generano geometrie dalle rappresentazioni in formato binario:

ST_GeomFromWKB

Crea una geometria dalla rappresentazione WKB di qualsiasi tipo di geometria.

ST_PointFromWKB

Crea un punto dalla rappresentazione WKB di un punto.

ST_LineFromWKB

Crea una stringa lineare dalla rappresentazione WKB di una stringa lineare.

ST_PolyFromWKB

Crea un poligono dalla rappresentazione WKB di un poligono.

ST_MPointFromWKB

Crea un multipunto dalla rappresentazione WKB di un multipunto.

ST_MLineFromWKB

Crea una multistringa lineare dalla rappresentazione WKB di una multistringa lineare.

ST_MPolyFromWKB

Crea un multipoligono dalla rappresentazione WKB di un multipoligono.

La rappresentazione WKB rappresenta un flusso di byte contiguo. Tale rappresentazione consente lo scambio tra un client ODBC e un database SQL in formato binario. Poiché queste funzioni della geometria richiedono che la definizione delle strutture di linguaggio di programmazione C corrispondano alla rappresentazione binaria, sono state progettate per l'utilizzo in un programma (3GL). Inoltre non sono adatte per un ambiente (4GL). La funzione `ST_AsBinary` converte un valore della geometria esistente in una rappresentazione WKB.

La rappresentazione WKB per la geometria si ottiene mediante la serializzazione di un'istanza geometrica in una sequenza di tipi numerici. Questi tipi si rilevano dal gruppo (unsigned integer, double), quindi ciascun tipo numerico è disposto in serie come sequenza di byte. I tipi vengono serializzati utilizzando una delle due rappresentazioni binarie standard definite per i tipi numerici (NDR, XDR). Una tag a un byte che precede i byte serializzati descrive la codifica binaria specifica (NDR o XDR) utilizzata per un flusso di byte della geometria. L'unica differenza tra le due codifiche di geometria è uno degli ordinamenti byte: la codifica XDR è Big Endian; la codifica NDR è Little Endian.

Definizioni del tipo numerico

Un *unsigned integer* è un tipo di dati a 32 bit (4 byte) che codifica un valore integer non negativo nell'intervallo [0, 4294967295].

Un *double* è un tipo di dati di precisione doppia a 64 bit (8 byte) che codifica un numero di precisione doppia utilizzando il formato di precisione doppia IEEE 754.

Queste definizioni sono comuni per XDR e NDR.

Codifica XDR (Big Endian) dei tipi numerici

La rappresentazione XDR di un unsigned integer è Big Endian (prima il byte più rilevante).

La rappresentazione XDR di un double è Big Endian (il bit di segno è il primo byte).

Codifica NDR (Little Endian) dei tipi numerici

La rappresentazione NDR di un unsigned integer è Little Endian (prima i byte minori).

La rappresentazione NDR di un double è Little Endian (il bit di segno è l'ultimo byte).

Conversione tra NDR e XDR

La conversione tra i tipi di dati NDR e XDR per gli unsigned integer e i double è un'operazione semplice. Tale operazione comprende l'inversione dell'ordine di byte all'interno di ciascun unsigned integer o double del flusso di byte.

Descrizione dei flussi di byte WKBGeometry

Questa sezione descrive la rappresentazione WKB per la geometria. Il blocco di creazione di base è il flusso di byte relativo a un punto che comprende due double. I flussi di byte per le altre geometrie vengono creati utilizzando i flussi di byte delle geometrie già definite.

```
// Basic Type definitions
// byte : 1 byte
// uint32 : 32 bit unsigned integer (4 bytes)
// double : double precision number (8 bytes)

// 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];
}

WKBGeometry {
  union {
    WKBPoint      point;
    WKBLineString linestring;
    WKBPolygon     polygon;
    WKBMultiPoint mpoint;
    WKBMultiLineString mlinestring;
    WKBMultiPolygon mpolygon;
  }
};

```

La seguente figura illustra una rappresentazione NDR.

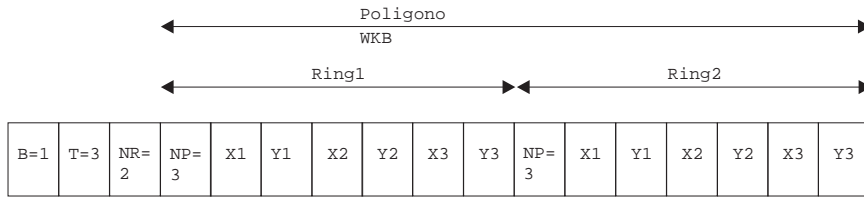


Figura 39. Rappresentazione nel formato NDR. (B=1) di tipo poligono (T=3) con 2 lineari (NR=2), ciascun anello con 3 punti (NP=3).

Asserzioni relative alla rappresentazione WKB

La rappresentazione WKB della geometria è progettata per rappresentare le istanze dei tipi di geometria descritti in Geometry Object Model e in OpenGIS Abstract Specification.

Queste asserzioni fanno riferimento agli anelli, i poligoni e i multipoligoni:

Anelli lineari

Gli anelli sono semplici e chiusi, quindi non possono intersecarsi tra loro.

Poligoni

Due anelli lineari nei contorni di un poligono non possono intersecarsi. Gli anelli lineari nei contorni di un poligono possono intersecarsi solo in un punto singolo, come tangente.

Multipoligoni

Gli interni di due poligoni, elementi di un multipoligono, non possono intersecarsi. I contorni di due poligoni, elementi di un multipoligono, possono toccarsi solo in un numero limitato di punti.

Le rappresentazioni in formato ESRI

DB2 Spatial Extender dispone di varie funzioni che generano geometrie dalle rappresentazioni in formato ESRI: La rappresentazione WKB GIS, oltre a supportare la rappresentazione bidimensionale, supporta anche le misure e le coordinate Z facoltative. Le seguenti funzioni generano geometrie da un formato ESRI:

ST_GeometryFromShape

Crea una geometria dalla rappresentazione del formato di qualsiasi tipo di geometria.

ST_PointFromShape

Crea un punto dalla rappresentazione del formato di un punto.

ST_LineFromShape

Crea una stringa lineare dalla rappresentazione del formato di una stringa lineare.

ST_PolyFromShape

Crea un poligono dalla rappresentazione del formato di un poligono.

ST_MPointFromShape

Crea un multipunto dalla rappresentazione del formato di un multipunto.

ST_MLineFromShape

Crea una multistringa lineare dalla rappresentazione del formato di una multistringa lineare.

ST_MPolyFromShape

Crea un multipoligono dalla rappresentazione del formato di un multipoligono.

La sintassi generale di queste funzioni è uguale. Il primo argomento è la rappresentazione del formato immesso come tipo di dati BLOB (binary large object). Il secondo argomento è il valore intero dell'identificativo di riferimento spaziale da assegnare alla geometria. La funzione `GeometryFromShape` presenta la seguente sintassi:

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), cr
db2gse.coordref)
```

Poiché queste funzioni del formato richiedono che la definizione delle strutture di linguaggio di programmazione C corrispondano alla rappresentazione binaria, sono state progettate per l'utilizzo in un programma 3GL e non sono adatte per un ambiente 4GL. La funzione `AsShape` converte il valore della geometria in una rappresentazione del formato ESRI.

Un tipo di formato 0 indica un formato nullo, privo di dati geometri.

Valore	Tipo formato
0	Null Shape
1	Punto
3*	Multiriga
5	Poligono
8	Multipunto
11	PuntoZ
13	MultirigaZ
15	PoligonoZ
18	MultipuntoZ

Valore	Tipo formato
21	PuntoM
23	MultirigaM
25	PoligonoM
28	MultipuntoM

Nota: * I tipi di formato che non sono specificati (2, 4, 6 etc.) sono riservati per un utilizzo futuro.

Tipi di formato nell'area XY

Punto

Un punto comprende una coppia di coordinate di precisione doppia nell'ordine X, Y.

Tabella 66. Contenuto del flusso di byte del punto

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	1	Integer	1	Little
Byte 4	P	P	Double	1	Little
Byte 12	Y	Y	Double	1	Little

Multipunto

Un Multipunto comprende una raccolta di punti. La casella di delimitazione è memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

Tabella 67. Contenuto del flusso di byte del multipunto

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Multirighe

Un Multirighe è una serie ordinata di vertici che comprende una o più parti. Una parte è una sequenza collegata di due o più punti. E' possibile che i punti non siano collegati tra loro. Non è rilevante la presenza di intersezioni tra le parti.

Poiché questa specifica consente la successione di punti con coordinate identiche, i programmi di lettura del file shape devono gestire tali casi. Inoltre, le parti con lunghezza zero che potrebbero derivare non sono consentite.

I campi relativi a un Multirighe sono:

Box La casella limite relativa al Multirighe memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts

Il numero delle parti nel Multirighe.

NumPoints

Il numero totale dei punti per tutte le parti.

Parts Una serie di lunghezze NumParts. Ogni Multiriga memorizza l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Points Una serie di lunghezze NumPoints. I punti per ciascuna parte del Multiriga vengono memorizzati da un'estremità all'altra. I punti relativi alla parte 2 seguono i punti relativi alla parte 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie dei punti tra le parti.

Tabella 68. Contenuto del flusso di byte del Multiriga

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Nota: $X = 44 + 4 * \text{NumParts}$.

Poligono

Un poligono consiste di uno o più anelli. Un anello è una sequenza collegata di quattro o più punti che forma un loop chiuso privo di intersezioni. Un poligono può contenere più anelli esterni. L'ordine dei vertici o l'orientamento di un anello indica quale lato dell'anello è l'interno del poligono. Il contorno di un punto a destra di un observer che si trova sull'anello secondo il senso dell'ordine dei vertici rappresenta l'interno del poligono. I vertici delle righe

che definiscono i fori dei poligoni sono in senso antiorario. I vertici per un poligono singolo con anelli, quindi, sono sempre in senso orario. Gli anelli di un poligono sono denominati parti.

Poiché questa specifica consente la successione di punti con coordinate identiche, i programmi di lettura del file shape devono gestire tali casi. Inoltre, le parti con area o lunghezza zero che potrebbero derivare non sono consentite.

I campi relativi a un poligono sono:

Box La casella limite relativa al poligono memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts
Il numero di anelli nel poligono.

NumPoints
Il numero totale di punti per tutti gli anelli.

Parts Una serie di lunghezze NumParts. Per ciascun anello, viene memorizzato l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Punti Una serie di lunghezze NumPoints. I punti per ciascun anello del poligono vengono memorizzati da un'estremità all'altra. I punti relativi all'anello 2 seguono i punti relativi all'anello 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascun anello. Non è presente alcun delimitatore nella serie dei punti tra gli anelli.

Note importanti relative alle forme Poligono:

- Gli anelli sono chiusi (il primo e l'ultimo vertice di un anello DEVE corrispondere).
- L'ordine degli anelli in una serie di punti non è rilevante.
- I poligoni memorizzati in un file shape devono essere puliti. Un poligono pulito è un poligono che presenta le seguenti caratteristiche:
 - Privo di intersezioni. Un segmento che appartiene a un anello non può intersecarsi con un segmento appartenente a un altro anello. Gli anelli di un poligono possono toccarsi nei vertici ma non possono presentare intersezioni con segmenti. I segmenti colineari sono considerati segmenti intersecanti.
 - L'interno del poligono sul lato "corretto" della linea che lo definisce. Il contorno di un punto a destra di un observer che si trova sull'anello secondo il senso dell'ordine dei vertici rappresenta l'interno del poligono. I vertici per un poligono

singolo con anelli, quindi, sono sempre in senso orario. Gli anelli che definiscono i fori in questi poligoni presentano un orientamento antiorario.

I poligoni "sporchi" sono quelli che presentano anelli che definiscono fori nel poligono anche con senso orario, quindi con sovrapposizione di interni.

Un esempio di istanza di poligono:

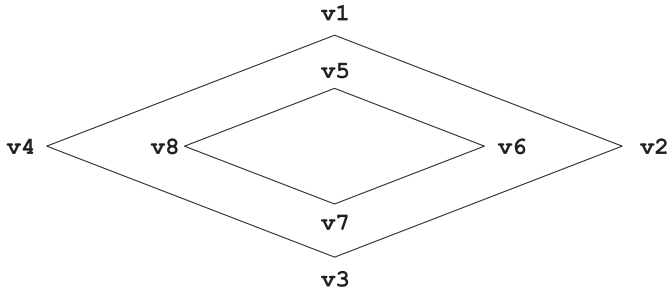


Figura 40. Un poligono con un foro e otto vertici

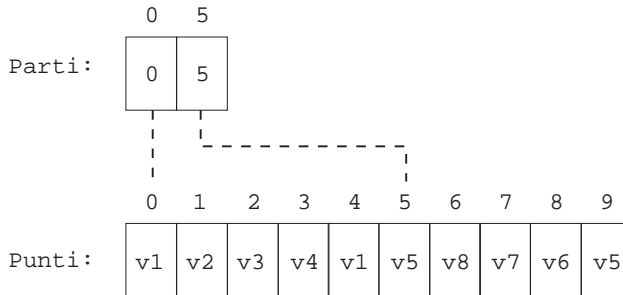


Figura 41. Contenuto del flusso di byte del poligono. NumParts corrisponde a 2 e NumPoints corrisponde a 10. Tenere presente che l'ordine dei punti del poligono a forma di ciambella (foro) viene invertito.

Tabella 69. Contenuto del flusso di byte del poligono

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Nota: $X = 44 + 4 * \text{NumParts}$.

Tipi di formato misura nell'area XY

PuntoM

Un PuntoM comprende una coppia di coordinate di precisione doppia nell'ordine X, Y con una misura M.

Tabella 70. Contenuto del flusso di byte di PuntoM

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	21	Integer	1	Little
Byte 4	P	P	Double	1	Little
Byte 12	Y	Y	Double	1	Little
Byte 20	M	M	Double	1	Little

MultipuntoM

I campi relativi a un MultipuntoM sono:

Box La casella limite relativa al MultipuntoM memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumPoints

Il numero di punti.

Points Una serie di punti di lunghezza NumPoints.

NumMs

Il numero di misure. NumMs può presentare solo due valori zero se nel campo non è presente alcuna misura oppure uguale a NumPoints se sono presenti misure.

M Range

Le misure minime e massime relative al MultipuntoM memorizzate nell'ordine Mmin, Mmax.

M Array

Una serie di misure di lunghezza NumPoints.

Tabella 71. Contenuto del flusso di byte di MultipuntoM

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Tabella 71. Contenuto del flusso di byte di MultipuntoM (Continua)

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte X+4*	Mmin	Mmin	Double	1	Little
Byte X+12*	Mmax	Mmax	Double	1	Little
Byte X+20*	Marray	Marray	Double	NumPoints	Little

Note:

1. $X = 40 + (16 * \text{NumPoints})$
2. * facoltativo

MultirigaM

Un file shape MultirigaM consiste di una o più parti. Una parte è una sequenza collegata di due o più punti. La connessione tra le parte non è rilevante. Le parti non possono intersecarsi.

I campi relativi a un MultirigaM sono:

Box La casella limite relativa al MultirigaM memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts

Il numero delle parti nel MultirigaM.

NumPoints

Il numero totale dei punti per tutte le parti.

Parts Una serie di lunghezze NumParts. Viene memorizzato, per ciascuna parte, l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Points Una serie di lunghezze NumPoints. I punti per ciascuna parte del MultirigaM vengono memorizzati da un'estremità all'altra. I punti relativi alla parte 2 seguono i punti relativi alla parte 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie dei punti tra le parti.

NumMs

Il numero di misure. NumMs può presentare solo due valori zero se nel campo non è presente alcuna misura oppure uguale a NumPoints se sono presenti misure.

M Range

Le misure minime e massime relative al MultirigaM memorizzate nell'ordine Mmin, Mmax.

M Array

Una serie di lunghezze NumPoints. Le misure per ciascuna parte del

MultirigaM vengono memorizzati da un'estremità all'altra. Le misure relative alla parte 2 seguono le misure relative alla parte 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie di misure tra le parti.

Tabella 72. Contenuto del flusso di byte di MultirigaM

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Note:

1. $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$.
2. * facoltativo

PoligonoM

Un PoligonoM comprende un numero di anelli. Un anello è un loop chiuso privo di intersezioni. Tenere presente che le intersezioni sono calcolate nell'area XY e *non* nell'area XYM. Un PoligonoM può contenere più anelli esterni. Gli anelli di un PoligonoM sono denominati parti.

I campi relativi a un PoligonoM sono:

Box La casella limite relativa al PoligonoM memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts

Il numero di anelli nel PoligonoM.

NumPoints

Il numero totale di punti per tutti gli anelli.

Parts

Una serie di lunghezze NumParts. Per ciascun anello, viene memorizzato l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Points Una serie di lunghezze NumPoints. I punti per ciascun anello di PoligonoM vengono memorizzati da un'estremità all'altra. I punti relativi all'anello 2 seguono i punti relativi all'anello 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascun anello. Non è presente alcun delimitatore nella serie dei punti tra gli anelli.

NumMs

Il numero di misure. NumMs può presentare solo due valori zero se nel campo non è presente alcuna misura oppure uguale a NumPoints se sono presenti misure.

M Range

Le misure minime e massime relative al PoligonoM memorizzate nell'ordine Mmin, Mmax.

M Array

Una serie di lunghezze NumPoints. Le misure per ciascun anello di PoligonoM vengono memorizzate da un'estremità all'altra. Le misure relative alla parte 2 seguono le misure relative all'anello 1 e così via. La serie delle parti contiene l'indice della serie della misura iniziale per ciascun anello. Non è presente alcun delimitatore nella serie di misure tra gli anelli.

Note importanti relative alle forme PoligonoM:

- Gli anelli sono chiusi (il primo e l'ultimo vertice di un anello devono corrispondere).
- L'ordine degli anelli in una serie di punti non è rilevante.

Tabella 73. Contenuto del flusso di byte di PoligonoM

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Note:

1. $X = 44 + (4 * NumParts)$, $Y = X + (16 * NumPoints)$.
2. * facoltativo

Tipi di forma nell'area XYZ**PuntoZ**

Un PuntoZ comprende coordinate di precisione doppia o tripla nell'ordine X, Y, Z più una misura.

Tabella 74. Contenuto del flusso di byte di PuntoZ

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	11	Integer	1	Little
Byte 4	P	P	Double	1	Little
Byte 12	Y	Y	Double	1	Little
Byte 20	Z	Z	Double	1	Little
Byte 28	Measure	M	Double	1	Little

MultipuntoZ

Un MultipuntoZ rappresenta una serie di PuntoZ, come segue:

- La casella di delimitazione è memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.
- L'intervallo Z limite viene memorizzato nell'ordine Zmin, Zmax.
L'intervallo M limite viene memorizzato nell'ordine Mmin, Mmax.

Tabella 75. Contenuto del flusso di byte di MultipuntoZ

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Note:

1. $X = 40 + (16 * \text{NumPoints}); Y = X + 16 + (8 * \text{NumPoints})$
2. * facoltativo

MultirigaZ

Un MultirigaZ consiste da una o più parti. Una parte è una sequenza collegata di due o più punti. La connessione tra le parte non è rilevante. Le parti non possono intersecarsi.

I campi relativi a un MultirigaZ sono:

Box La casella limite relativa al MultirigaZ memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts

Il numero delle parti nel MultirigaZ.

NumPoints

Il numero totale dei punti per tutte le parti.

Parts Una serie di lunghezze NumParts. Viene memorizzato, per ciascuna parte, l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Points Una serie di lunghezze NumPoints. I punti per ciascuna parte del MultirigaZ vengono memorizzati da un'estremità all'altra. I punti relativi alla parte 2 seguono i punti relativi alla parte 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie dei punti tra le parti.

Z Range

I valori Z minimi e massimi relativi al MultirigaZ memorizzati nell'ordine Zmin, Zmax.

Z Array

Una serie di lunghezze NumPoints. I valori Z per ciascuna parte del MultirigaZ vengono memorizzati da un'estremità all'altra. I valori Z della parte 2 seguono i valori Z della parte 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie Z tra le parti.

NumMs

Il numero di misure. NumMs può presentare solo due valori zero se nel campo non è presente alcuna misura oppure uguale a NumPoints se sono presenti misure.

M Range

Le misure minime e massime relative al MultirigaZ memorizzate nell'ordine Mmin, Mmax.

M Array

Una serie di lunghezze NumPoints. Le misure per ciascuna parte del MultirigaZ vengono memorizzate da un'estremità all'altra. Le misure relative alla parte 2 seguono le misure relative alla parte 1 e così via. La serie delle parti contiene l'indice della serie della misura iniziale per ciascuna parte. Non è presente alcun delimitatore nella serie di misure tra le parti.

Tabella 76. Contenuto del flusso di byte MultirigaZ

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Note:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$, $Z = Y + 16 + (8 * \text{NumPoints})$
2. * facoltativo

PoligonoZ

Un PoligonoZ comprende un numero di anelli. Un anello è un loop chiuso privo di intersezioni. Un PoligonoZ può contenere più anelli esterni. Gli anelli di un PoligonoZ sono denominati parti.

I campi relativi a un PoligonoZ sono:

Box La casella limite relativa al PoligonoZ memorizzata nell'ordine Xmin, Ymin, Xmax, Ymax.

NumParts

Il numero di anelli nel PoligonoZ.

NumPoints

Il numero totale di punti per tutti gli anelli.

Parts Una serie di lunghezze NumParts. Per ciascun anello, viene memorizzato l'indice del relativo primo punto nella serie di punti. Gli indici della serie fanno riferimento a 0.

Points Una serie di lunghezze NumPoints. I punti per ciascun anello di PoligonoZ vengono memorizzati da un'estremità all'altra. I punti relativi all'anello 2 seguono i punti relativi all'anello 1 e così via. La serie delle parti contiene l'indice della serie del punto iniziale per ciascun anello. Non è presente alcun delimitatore nella serie dei punti tra gli anelli.

Z Range

I valori Z minimi e massimi relativi ad arc memorizzati nell'ordine Zmin, Zmax.

Z Array

Una serie di lunghezze NumPoints. I valori Z per ciascun anello di PoligonoZ vengono memorizzati da un'estremità all'altra. I valori Z dell'anello 2 seguono i valori Z dell'anello 1 e così via. La serie delle parti contiene l'indice della serie del valore Z iniziale per ciascun anello. Non è presente alcun delimitatore nella serie di valori Z tra gli anelli.

NumMs

Il numero di misure. NumMs può presentare solo due valori zero se nel campo non è presente alcuna misura oppure uguale a NumPoints se sono presenti misure.

M Range

Le misure minime e massime relative al PoligonoZ memorizzate nell'ordine Mmin, Mmax.

M Array

Una serie di lunghezze NumPoints. Le misure per ciascun anello di PoligonoZ vengono memorizzati da un'estremità all'altra. Le misure relative alla parte 2 seguono le misure relative all'anello 1 e così via. La serie delle parti contiene l'indice della serie della misura iniziale per ciascun anello. Non è presente alcun delimitatore nella serie di misure tra gli anelli.

Note importanti relative alle forme PoligonoZ:

- Gli anelli sono chiusi (il primo e l'ultimo vertice di un anello DEVE corrispondere).
- L'ordine degli anelli in una serie di punti non è rilevante.

Tabella 77. Contenuto del flusso di byte di PoligonoZ

Posizione	Campo	Valore	Tipo	Numero	Ordine
Byte 0	Tipo formato	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

Parte 3. Appendici

Informazioni particolari

E' possibile che negli altri paesi l'IBM non offra i prodotti, le funzioni o i servizi illustrati in questo documento. Consultare il rappresentante IBM locale per informazioni sui prodotti o sui servizi disponibili nel proprio paese. Ogni riferimento relativo a prodotti, programmi o servizi IBM non implica che solo quei prodotti, programmi o servizi IBM possono essere utilizzati. In sostituzione a quelli forniti dall'IBM, è possibile usare prodotti, programmi o servizi funzionalmente equivalenti che non comportino violazione dei diritti di proprietà intellettuale o di altri diritti dell'IBM. E' comunque responsabilità dell'utente valutare e verificare la possibilità di utilizzare altri prodotti, programmi o servizi non IBM.

L'IBM può avere brevetti o domande di brevetti in corso relativi a quanto trattato nella presente pubblicazione. La fornitura di questa pubblicazione non implica la concessione di alcuna licenza su di essi. Chi desiderasse ricevere informazioni relative alle licenze può rivolgersi per iscritto a:

IBM Director of Commercial Relations
IBM Corporation
Schoenaicher Str. 220
D-7030 Boeblingen
Deutschland

Il seguente paragrafo non è valido per il Regno Unito o per tutti i paesi le cui leggi nazionali siano in contrasto con le disposizioni in esso contenute:
L'INTERNATIONAL BUSINESS MACHINES CORPORATION FORNISCE QUESTA PUBBLICAZIONE NELLO STATO IN CUI SI TROVA SENZA ALCUNA GARANZIA, ESPLICITA O IMPLICITA, IVI INCLUSE EVENTUALI GARANZIE DI COMMERCIALIZZABILITA' ED IDONEITA' AD UNO SCOPO SPECIFICO. Alcuni stati non consentono la rinuncia a garanzie esplicite o implicite in determinate transazioni, quindi, la presente dichiarazione potrebbe non essere a voi applicabile.

Questa pubblicazione potrebbe contenere imprecisioni tecniche o errori tipografici. Le informazioni incluse in questo documento vengono modificate su base periodica; tali modifiche verranno incorporate nelle nuove edizioni della pubblicazione. L'IBM si riserva il diritto di apportare miglioramenti e/o modifiche al prodotto o al programma descritto nel manuale in qualsiasi momento e senza preavviso.

Tutti i riferimenti a siti Web non dell'IBM sono forniti unicamente a scopo di consultazione. I materiali contenuti in tali siti Web non fanno parte di questo prodotto e l'utente si assume ogni rischio relativo al loro utilizzo.

L'IBM può utilizzare o divulgare le informazioni ricevute dagli utenti secondo le modalità ritenute appropriate, senza alcun obbligo nei loro confronti.

Coloro che detengono la licenza su questo programma e desiderano avere informazioni allo scopo di consentire: (i) uno scambio di informazioni tra programmi indipendenti e altri (compreso questo) e (ii) l'uso reciproco di tali informazioni, dovrebbero rivolgersi a:

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

Queste informazioni possono essere rese disponibili, secondo condizioni contrattuali appropriate, compreso, in alcuni casi, il pagamento in addebito.

Il programma su licenza descritto in questo manuale e tutto il materiale su licenza ad esso relativo sono forniti dall'IBM nel rispetto delle condizioni previste dalla licenza d'uso.

Qualsiasi informazione relativa alle prestazioni è stata verificata in un ambiente controllato. Di conseguenza l'utilizzo del prodotto in ambienti operativi diversi può comportare risultati sensibilmente diversi. Alcune rilevazioni possono essere state effettuate su sistemi a livello di sviluppo e non si garantisce in alcun modo, dunque, che siano uguali alle rilevazioni eseguite sui vari sistemi disponibili. Inoltre, è possibile che ad alcune di queste rilevazioni si sia pervenuti tramite estrapolazione. I risultati reali potrebbero variare. E' necessario che gli utenti confrontino i dati in base agli ambienti utilizzati.

Le informazioni relative a prodotti non IBM sono state ottenute dai fornitori di tali prodotti. L'IBM non ha verificato tali prodotti e non può garantire l'accuratezza delle prestazioni. Eventuali commenti relativi alle prestazioni dei prodotti non IBM devono essere indirizzati ai fornitori di tali prodotti.

Le dichiarazioni relative a futuri intenti o obiettivi IBM sono soggette a modifiche senza preavviso.

Questa pubblicazione contiene esempi di dati e prospetti usati quotidianamente nelle operazioni aziendali. Pertanto può contenere nomi di persone, società, marchi e prodotti. Tutti i nomi contenuti nel manuale sono fittizi e ogni riferimento a nomi ed indirizzi reali è puramente casuale.

LICENZA RELATIVA AI DIRITTI D'AUTORE:

Queste informazioni contengono programmi applicativi di esempio in lingua originale che illustrano le tecniche di programmazione su diverse piattaforme operative. Potete copiare, modificare e distribuire questi esempi di programmi sotto qualsiasi forma senza alcun pagamento alla IBM, allo scopo di sviluppare, utilizzare, commercializzare o distribuire i programmi applicativi in modo conforme alle API (Application Programming Interface) a seconda della piattaforma operativa per cui tali esempi di programmi sono stati scritti. Questi esempi non sono stati testati approfonditamente tenendo conto di tutte le condizioni possibili. La IBM, quindi, non può garantire o assicurare l'affidabilità, la praticità o il funzionamento di questi programmi.

Marchi

I termini di seguito riportati, contrassegnati da un asterisco (*), sono marchi della 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
AS/400	OS/2
BookManager	OS/390
CICS	OS/400
C Set++	PowerPC
C/370	QBIC
DATABASE 2	QMF
DataHub	RACF
DataJoiner	RISC System/6000
DataPropagator	RS/6000
DataRefresher	S/370
DB2	SP
DB2 Connect	SQL/DS
DB2 Extenders	SQL/400
DB2 OLAP Server	System/370
DB2 Universal Database	System/390
Distributed Relational Database Architecture	SystemView
DRDA	VisualAge
eNetwork	VM/ESA
Extended Services	VSE/ESA
FFST	VTAM
First Failure Support Technology	WebExplorer
	WIN-OS/2

I termini seguenti sono marchi di altre società:

Microsoft, Windows, Windows NT sono marchi o marchi registrati della Microsoft Corporation.

Java e tutti i marchi a base Java sono marchi della Sun Microsystems, Inc. negli Stati Uniti e/o altri paesi.

Tivoli e NetView sono marchi della Tivoli Systems, Inc. negli Stati Uniti e/o altri paesi.

UNIX è un marchi registrato negli Stati Uniti e in altri paesi autorizzati esclusivamente attraverso la X/Open Company Limited.

Nomi di altre prodotti, società e servizi indicati da un doppio asterisco (**), possono essere marchi di altre società.

Indice analitico

A

- abilitazione del database per le operazioni spaziali
 - descrizione 10
 - discussione 24
 - scelte di menu del Centro di controllo DB2 24
 - AIX
 - il punto in cui vengono memorizzate le definizioni macro per le costanti 67
 - installazione di DB2 Spatial Extender 19
 - ubicazione di memorizzazione dei dati di riferimento 23
 - anelli lineari 304
 - applicazioni
 - indicazioni per la scrittura 57
 - procedure memorizzate 67
 - ArcExplorer
 - scaricamento 21
 - utilizzo di un'interfaccia 9, 53
 - AsBinaryShape 164, 166
- ## B
- busta 121, 134
- ## C
- Centro di controllo DB2
 - finestra Creare indici spaziali 51
 - finestra Creare riferimento spaziale 29, 31
 - finestra Creare struttura spaziale
 - registrazione di una colonna della tabella come struttura 36
 - registrazione di una colonna della vista come struttura 38
 - finestra Eseguire geocoder 43, 44
 - Finestra Esportare dati spaziali 50
 - finestra Importare dati spaziali 46, 48, 49
 - richiamo di DB2 Spatial Extender da 22
 - classe 132
 - codifica NDR 301, 302
 - codifica XDR 301, 302

- colonne spaziali 41
- Contenuto del flusso di byte del multipunto 306
- Contenuto del flusso di byte del Multiriga 307
- Contenuto del flusso di byte del poligono 309
- Contenuto del flusso di byte del punto 306
- Contenuto del flusso di byte di MultipuntoM 310
- Contenuto del flusso di byte di MultipuntoZ 314
- Contenuto del flusso di byte di MultirigaM 312
- Contenuto del flusso di byte di PoligonoM 313
- Contenuto del flusso di byte di PoligonoZ 318
- Contenuto del flusso di byte di PuntoM 310
- Contenuto del flusso di byte di PuntoZ 314
- Contenuto del flusso di byte MultirigaZ 316
- coordinate
 - coordinate X
 - descrizione 25
 - proprietà delle geometrie 132
 - coordinate Y
 - descrizione 25
 - proprietà delle geometrie 132
 - coordinate Z
 - descrizione 26
 - proprietà delle geometrie 132
- descrizione 6
- coordinate X
 - descrizione 25
 - proprietà delle geometrie 132
- coordinate Y
 - descrizione 25
 - proprietà delle geometrie 132
- coordinate Z
 - descrizione 26
 - proprietà delle geometrie 132

D

- database
 - abilitazione per le operazioni spaziali
 - db2gse.gse_enable_db 77
 - discussione 24
 - scelte di menu del Centro di controllo DB2 24
 - attivazione per le operazioni spaziali
 - programma di esempio 58
 - disattivazione del supporto per le operazioni spaziali
 - programma di esempio 58
 - supporto di disabilitazione per le operazioni spaziali
 - db2gse.gse_disable_db 72
- dati di attributo 5
- dati di origine 5
- dati di riferimento 23
- dati geodetici 290
- dati spaziali
 - derivati da altri dati spaziali
 - discussione 7
 - funzioni spaziali da cui derivano i dati 155
 - derivati da dati di attributo 7
- esportazione
 - db2gse.gse_export_shape 84
 - descrizione 45
 - Finestra Esportare dati spaziali 49
 - programma di esempio 64
- formati file
 - rappresentazioni in formato ESRI 163, 304
 - rappresentazioni WKB (well-known binary) 162, 300
 - rappresentazioni WKT (well-known text) 160, 295
- importazione
 - db2gse.gse_import_sde 86
 - db2gse.gse_import_shape 89
 - descrizione 45
 - discussione 8
 - finestra Importare dati spaziali 46, 48
 - programma di esempio 60

- dati spaziali (*Continua*)
 - struttura 6
- DB2 Spatial Extender
 - applicazioni
 - indicazioni per la scrittura 57
 - procedure memorizzate 67
 - attività, riepiloghi
 - programma di esempio 57
 - attività, riepilogo
 - eseguite dalle procedure memorizzate 68
 - panoramica 10
 - scenario 12
 - configurazione 17
 - funzioni spaziali 165
 - installazione
 - requisiti hardware e software 17
 - su AIX 19
 - su Windows NT 19
 - verifica 20
 - interfacce per 9
 - messaggi di errore, di avvertenza e informativi 107
 - procedure memorizzate 67
 - programma di esempio
 - compilazione ed esecuzione 20
 - descrizione 57
 - richiamo dal Centro di controllo DB2 22
 - risorse
 - dati di riferimento 23
 - per le operazioni spaziali 24
 - riepilogo 23
 - utilizzo 3
 - viste del catalogo 115
- DB2GSE.COORD_REF_SYS 115
- DB2GSE.GEOMETRY_COLUMNS 116
- db2gse.gse_disable_autogc 70
- db2gse.gse_disable_db 72
- db2gse.gse_disable_sref 73
- db2gse.gse_enable_autogc 74
- db2gse.gse_enable_db 77
- db2gse.gse_enable_idx 78
- db2gse.gse_enable_sref 81
- db2gse.gse_export_shape 84
- db2gse.gse_import_sde 86
- db2gse.gse_import_shape 89
- db2gse.gse_register_gc 91
- db2gse.gse_register_layer 93
- db2gse.gse_run_gc 100
- db2gse.gse_unregist_gc 103
- db2gse.gse_unregist_layer 104
- DB2GSE.SPATIAL_GEOCODER 116
- DB2GSE.SPATIAL_REF_SYS 117
- db2iupdt (programma di utilità dell'istanza DB2) 21
- dimensione 134
- E**
- EBNF (Extended Backus Naur) 285
- EnvelopesIntersect 148, 168
- esterno 130, 133
- F**
- false M
 - specifica 28, 32
- false X
 - specifica 28, 31
- false Y
 - specifica 28, 31
- false Z
 - specifica 28, 31
- fattori di scala
 - specifica 28, 31
- fattori offset
 - specifica 27, 31
- finestra Creare indici spaziali 51
- finestra Creare riferimento spaziale 29, 31
- finestra Creare struttura spaziale
 - registrazione di una colonna della tabella come struttura 36
 - registrazione di una colonna della vista come struttura 38
- finestra Eseguire geocoder 43, 44
- Finestra Esportare dati spaziali 50
- finestra Importare dati spaziali 46, 48, 49
- flussi di byte WKBBGeometry 302
- formati
 - nell'area XY 306
 - nell'area XYZ 314
- funzioni geografiche
 - descrizione 3
 - rappresentate da dati 4
 - tipi di dati associati 34
- funzioni spaziali
 - .ST_Area 188
 - AsBinaryShape 164, 166
 - classificazione delle categorie in base alle operazioni eseguite 53
 - EnvelopesIntersect 148, 168
 - GeometryFromShape 163
 - Is3d 132, 170
 - IsMeasured 133, 171
 - LineFromShape 163, 172
 - LocateAlong 159, 174
- funzioni spaziali (*Continua*)
 - LocateBetween 159, 176
 - M 136, 178
 - MLine FromShape 179
 - MLineFromShape 163
 - MPointFromShape 163, 181
 - MPolyFromShape 163, 182
 - PointFromShape 163, 183
 - PolyFromShape 163, 184
 - predicati 54
 - ShapeToSQL 163, 186
 - ST_Area 139, 142
 - ST_AsBinary 163, 190
 - ST_AsText 161, 191
 - ST_Boundary 133, 192
 - ST_Buffer 157, 194
 - ST_Centroid 139, 142, 196
 - ST_Contains 153, 197
 - ST_Convexhull 199
 - ST_ConvexHull 160
 - ST_CoordDim 136, 201
 - ST_Crosses 150, 203
 - ST_Difference 156, 205
 - ST_Dimension 135, 206
 - ST_Disjoint 146, 208
 - ST_Distance 155, 210
 - ST_Endpoint 137, 211
 - ST_Envelope 134, 212
 - ST_Equals 145, 214
 - ST_ExteriorRing 139, 215
 - ST_GeometryFromText 217
 - ST_GeometryN 140, 221
 - ST_GeometryType 132, 222
 - ST_GeomFromText 161, 295
 - ST_GeomFromWKB 162, 219, 300
 - ST_InteriorRingN 139, 224
 - ST_Intersection 155, 229
 - ST_Intersects 147, 231
 - ST_IsClosed 137, 140, 232
 - ST_IsEmpty 133, 234
 - ST_IsRing 137, 236
 - ST_IsSimple 133, 237
 - ST_IsValid 132, 238
 - ST_Length 137, 140, 240
 - ST_LineFromText 161, 242, 295
 - ST_LineFromWKB 162, 243, 300
 - ST_MLineFromText 161, 245, 295
 - ST_MLineFromWKB 162, 246, 301
 - ST_MPointFromText 161, 248, 295
 - ST_MPointFromWKB 162, 249, 300

funzioni spaziali (*Continua*)

- ST_MPolyFromText 161, 250, 295
- ST_MPolyFromWKB 162, 251, 301
- ST_NumGeometries 139, 252
- ST_NumInteriorRing 139, 253
- ST_NumPoints 137, 254
- ST_OrderingEquals 146, 255
- ST_Overlaps 149, 256
- ST_Perimeter 139, 258
- ST_Point 136, 261
- ST_PointFromText 136, 259, 295
- ST_PointFromWKB 162, 260, 300
- ST_PointN 137, 262
- ST_PointOnSurface 139, 263
- ST_PolyFromText 161, 264, 295
- ST_PolyFromWKB 162, 265, 300
- ST_Polygon 160, 267
- ST_Relate 155, 268
- ST_SRID 135, 270
- ST_StartPoint 137, 271
- ST_SymmetricDiff 272
- ST_Touches 148, 274
- ST_Transform 135, 275
- ST_Union 157, 276
- ST_Within 152, 277
- ST_WKBTToSQL 162, 278
- ST_WKTTToSQL 160, 280
- ST_X 136, 281
- ST_Y 136, 282

tipi

- associate alle proprietà delle geometrie 131
- associati alle geometrie istanziabili 135
- funzioni che confrontano le geometrie 143
- funzioni che generano geometrie 155
- funzioni che mostrano relazioni tra geometrie 143
- funzioni predicato 143
- scambio di dati 160
- utilizzo per l'impostazione degli indici spaziali 55
- Z 136

G

geocoder

- abilitazione automatica geocoder discussione 33
- finestra Creare struttura spaziale 36

geocoder (*Continua*)

- attivazione geocoding automatico descrizione 42
- programma di esempio 61
- disabilitazione geocode automatico db2gse.gse_disable_autogc 70
- disabilitazione geocoding automatico programma di esempio 62
- disattivazione automatica del geocoding finestra Eseguire geocoder 44
- esecuzione in modalità batch descrizione 42
- finestra Eseguire geocoder 43
- programma di esempio 60, 62
- geocoder non predefiniti descrizione 41
- utilizzo db2gse.gse_unregist_gc per annullare la registrazione 103
- geocoder predefiniti 41
- vista del catalogo DB2GSE.SPATIAL_GEOCODER 116
- geocoder predefiniti 41
- geocoders
 - attivazione geocoding automatico db2gse.gse_enable_autogc 74
 - esecuzione in modalità batch db2gse.gse_run_gc 100
 - geocoder non predefiniti utilizzo di db2gse.gse_register_gc per registrare 91
- geocoding
 - batch 42
 - descrizione 7, 41
 - incrementale 42
 - precisione 15
- geocoding automatico 42
- geocoding batch 42
- geocoding incrementale 42
- geometrie
 - corrispondenza con i tipi di dati spaziali 131
 - descrizione 129
 - griglie di indici spaziali 121
 - multipoligoni 131, 141
 - multipunti 131, 139

geometrie (*Continua*)

- multistringhe lineari 131, 140
 - poligoni 131, 138
 - proprietà
 - busta 121, 134
 - classe 132
 - coordinate X 132
 - coordinate Y 132
 - coordinate Z 132
 - dimensione 134
 - esterno 130, 133
 - interno 130, 133
 - limite 130, 133
 - misure 132
 - semplice o non semplice 133
 - SRID (spatial reference system identifier) 135
 - vuoto o non vuoto 133
 - punti 131, 135
 - stringhe lineari 131, 136
 - GeometryFromShape 163, 167
 - GIS (geographic information system)
 - creazione 10
 - descrizione 3
 - uso 11
- ## I
- indici con struttura ad albero B 120
 - indici griglia 51
 - indici spaziali 119
 - creare
 - db2gse.gse_enable_idx 78
 - determinazione della dimensione di griglia 52, 126
 - Finestra Creare indici spaziali 51
 - creazione
 - programma di esempio 61
 - impostazione 55
 - indici griglia 51
 - modalità di generazione 121
 - uso 125
 - informazioni spaziali
 - descrizione 3
 - richiamo e analisi
 - impostazione degli indici spaziali 55
 - interfacce da utilizzare 9, 53
 - programma di esempio 63
 - tipi di funzioni spaziali da utilizzare 53
 - utilizzo delle funzioni di predicato spaziali 54

- installazione di DB2 Spatial Extender
 - requisiti hardware e software 17
 - su AIX 19
 - su Windows NT 19
 - verifica 20
- interfacce per DB2 Spatial Extender 9
- interno 130, 133
- interrogazioni
 - impostazione degli indici spaziali 55
 - interfacce per l'inoltro 9, 53
 - tipi di funzioni spaziali da utilizzare 53
 - utilizzo delle funzioni di predicato spaziali 54
- Is3d 132, 170
- IsMeasured 133, 171
- J**
- Java 2 Runtime Environment (JRE)
 - v1.2.2 21
- L**
- limite 130, 133
- LineFromShape 163, 172
- LocateAlong 159, 174
- LocateBetween 159, 176
- M**
- M 136, 178
- matrici modello 144
- meridiani principali 292
- messaggi 107
- messaggi di avvertenza 107
- messaggi di errore 107
- messaggi informativi 107
- misure
 - descrizione 26, 132
 - proprietà delle geometrie 132
- MLine FromShape 179
- MLineFromShape 163
- modello di sistema coordinate
 - POSC/EPSPB 285
- MPointFromShape 163, 181
- MPolyFromShape 163, 182
- multipoligoni 131, 141
- multipunti 131, 139
- multistringhe lineari 131, 140
- P**
- Parametri di proiezione della carta 293
- parola chiave GEOGCS 286
- parola chiave PROJCS 286
- parola chiave UNIT 286
- PointFromShape 163, 183
- poligoni 131, 138
- PolyFromShape 163, 184
- precisione
 - geocoding 15, 43
 - mantenimento per sistemi di riferimento spaziali 27
- procedure memorizzate
 - db2gse.gse_disable_autogc 70
 - db2gse.gse_disable_db 72
 - db2gse.gse_disable_sref 73
 - db2gse.gse_enable_autogc 74
 - db2gse.gse_enable_db 77
 - db2gse.gse_enable_idx 78
 - db2gse.gse_enable_sref 81
 - db2gse.gse_export_shape 84
 - db2gse.gse_import_sde 86
 - db2gse.gse_import_shape 89
 - db2gse.gse_register_gc 91
 - db2gse.gse_register_layer 93
 - db2gse.gse_run_gc 100
 - db2gse.gse_unregist_gc 103
 - db2gse.gse_unregist_layer 104
- programma di esempio
 - compilazione ed esecuzione 20
 - descrizione 57
- programma di utilità dell'istanza DB2 (db2iupdt) 21
- proiezioni
 - azimutali 293
 - carta
 - parametri 293
 - tipi 292
 - coniche 293
 - planari 293
- proiezioni azimutali 293
- proiezioni coniche 293
- proiezioni della carta 292
- proiezioni planari 293
- punti 131, 135
- Q**
- query
 - programma di esempio 63
- R**
- rappresentazione a tasselli 160
- rappresentazioni in formato ESRI
 - descrizione 304
 - funzioni spaziali associate 163
- rappresentazioni WKB (well-known binary)
 - descrizione 300
 - funzioni spaziali associate 162
- rappresentazioni WKT (well-known text)
 - descrizione 295
- rappresentazioni WKT (well-known text) (*Continua*)
 - funzioni spaziali associate 160
 - requisiti software 18
 - requisiti spazio su disco 18
- S**
- scenario delle attività 12
- semplice o non semplice 133
- sferoidi 288
- ShapeToSQL 163, 186
- sistema di coordinate
 - geocentriche 287
- sistemi di coordinate 285
 - DB2GSE.COORD_REF_SYS vista del catalogo 115
 - derivazione di sistemi di riferimento spaziale 26
 - descrizione 6, 25
- sistemi di riferimento spaziale
 - creare
 - db2gse.gse_enable_sref 81
 - discussione 25
 - finestra Creare riferimento spaziale 29
 - programma di esempio 58
 - descrizione 11
 - in fase di cancellazione
 - db2gse.gse_disable_sref 73
 - programma di esempio 58
 - specifici di parametri
 - false M 28, 32
 - false X 28, 31
 - false Y 28, 31
 - false Z 28, 31
 - fattori di scala 28, 31
 - fattori offset 27, 31
 - unità M 29, 32
 - unità XY 28, 31
 - unità Z 29, 32
 - vista del catalogo
 - DB2GSE.SPATIAL_REF_SYS 117
- SRID (spatial reference system identifier) 135, 297
- ST_Area 139, 142, 188
- ST_AsBinary 163, 190
- ST_AsText 161, 191
- ST_Boundary 133, 192
- ST_Buffer 157, 194
- ST_Centroid 139, 142, 196
- ST_Contains 153, 197
- ST_Convexhull 199
- ST_ConvexHull 160
- ST_CoordDim 136, 201
- ST_Crosses 150, 203
- ST_Difference 156, 205

- ST_Dimension 135, 206
- ST_Disjoint 146, 208
- ST_Distance 155, 210
- ST_Endpoint 137, 211
- ST_Envelope 134, 212
- ST_Equals 145, 214
- ST_ExteriorRing 139, 215
- ST_GeometryFromText 217
- ST_GeometryN 140, 221
- ST_GeometryType 132, 222
- ST_GeomFromText 161, 295
- ST_GeomFromWKB 162, 219, 300
- ST_InteriorRingN 139, 224
- ST_Intersection 155, 229
- ST_Intersects 147, 231
- ST_IsClosed 137, 140, 232
- ST_IsEmpty 133, 234
- ST_IsRing 137, 236
- ST_IsSimple 133, 237
- ST_IsValid 132, 238
- ST_Length 137, 140, 240
- ST_LineFromText 161, 242, 295
- ST_LineFromWKB 162, 243, 300
- ST_MLineFromText 161, 245, 295
- ST_MLineFromWKB 162, 246, 301
- ST_MPointFromText 161, 248, 295
- ST_MPointFromWKB 162, 249, 300
- ST_MPolyFromText 161, 250, 295
- ST_MPolyFromWKB 162, 251, 301
- ST_NumGeometries 139, 252
- ST_NumInteriorRing 139, 253
- ST_NumPoints 137, 254
- ST_OrderingEquals 146, 255
- ST_Overlaps 149, 256
- ST_Perimeter 139, 258
- ST_Point 136, 261
- ST_PointFromText 136, 259, 295
- ST_PointFromWKB 162, 260, 300
- ST_PointN 137, 262
- ST_PointOnSurface 139, 263
- ST_PolyFromText 161, 264, 295
- ST_PolyFromWKB 162, 265, 300
- ST_Polygon 160, 267
- ST_Relate 155, 268
- ST_SRID 135, 270
- ST_StartPoint 137, 271
- ST_SymmetricDiff 272
- ST_Touches 148, 274
- ST_Transform 135, 275
- ST_Union 157, 276
- ST_Within 152, 277
- ST_WKBToSQL 162, 278
- ST_WKTToSQL 160, 280
- ST_X 136, 281
- ST_Y 136, 282

- stringhe lineari 131, 136, 137
- strutture
 - DB2GSE.GEOMETRY_
 - COLUMNS viste del catalogo 116
 - descrizione 12
 - registrazione delle colonne della vista
 - db2gse.gse_register_layer 93
 - finestra Creare struttura spaziale 38
 - programma di esempio 63
 - registrazione delle colonne di tabella
 - db2gse.gse_register_layer 93
 - finestra Creare struttura spaziale 36
 - programma di esempio 60
 - utilizzo della procedura memorizzata
 - db2gse.gse_unregister_layer per annullare la registrazione 104

T

- tipi di dati spaziali 33
 - corrispondenza con le geometrie 131
 - descrizione 33
- Tipi di formato misura nelle aree
 - XY 310
- trigger
 - attivazione geocoding automatico
 - db2gse.gse_enable_autogc 74
 - disabilitazione geocode automatico
 - db2gse.gse_disable_autogc 70
 - utilizzo per il richiamo di geocoder 33
 - utilizzo per richiamare il geocoder 42

U

- unità angolari 288
- unità lineari 287
- unità M
 - specifica 29, 32
- unità XY
 - specifica 28, 31
- unità Z
 - specifica 29, 32

V

- viste del catalogo
 - DB2GSE.COORD_REF_SYS 115
 - DB2GSE.GEOMETRY_
 - COLUMNS 116

- viste del catalogo (*Continua*)
 - DB2GSE.SPATIAL_
 - GEOCODER 116
 - DB2GSE.SPATIAL_REF_SYS 117
 - voci di dati 6
 - vuoto o non vuoto 133

W

- Windows NT
 - il punto in cui vengono memorizzate le definizioni macro per le costanti 67
 - installazione di DB2 Spatial Extender 19
 - ubicazione di memorizzazione dei dati di riferimento 23
- WKBBGeometry 302

Z

- Z 136

Come ottenere ulteriori informazioni dalla IBM

Se si riscontrano problemi tecnici, rivedere ed eseguire quanto suggerito nel manuale *Troubleshooting Guide* prima di contattare il supporto clienti DB2. Questo manuale fornisce informazioni utili da comunicare al supporto clienti DB2 per una rapida risoluzione dei problemi.

Contattare il rappresentante IBM di zona oppure un qualsiasi distributore software IBM per ordinare o ottenere informazioni sui prodotti DB2 Universal Database.

Informazioni sul prodotto

<http://www.ibm.com/software/data/>

Le pagine WWW del DB2 forniscono le informazioni DB2 correnti relative alle novità, descrizioni dei prodotti, programmi educativi ed altro.

<http://www.ibm.com/software/data/db2/library/>

DB2 Product and Service Technical Library fornisce l'accesso alle FAQ (frequently asked questions), le correzioni, i manuali e le informazioni tecniche DB2 più aggiornate.

Nota: Queste informazioni sono soltanto in Inglese.

<http://www.elink.ibm.com/pbl/pbl/>

Il sito Web per ordinare pubblicazioni internazionali fornisce informazioni su come ordinare tali pubblicazioni.

<http://www.ibm.com/education/certify/>

Il sito web Professional Certification Program fornisce informazioni relative alla verifica di certificazione per una vasta gamma di prodotti IBM, incluso DB2.

<ftp://software.ibm.com>

Collegarsi come anonymous. Nell'indirizzo `/ps/products/db2`, è possibile trovare i demo, le correzioni, le informazioni e gli strumenti relativi al DB2 e a molti altri prodotti.

<comp.databases.ibm-db2>, <bit.listserv.db2-l>

Questi newsgroup Internet sono a disposizione degli utenti per discutere delle loro esperienze con i prodotti DB2.

Da Compuserve: GO IBMDB2

Immettere questo comando per accedere ai forum IBM DB2 Family.
Tutti i prodotti DB2 sono supportati mediante questi forum.

Per informazioni su come contattare l'IBM al di fuori degli Stati Uniti, consultare l'Appendice A dell'*IBM Software Support Handbook*. Per accedere a questo documento, visitare la seguente pagina Web:

<http://www.ibm.com/support/>, quindi selezionare il collegamento IBM Software Support Handbook accanto al pulsante della pagina.

Nota: In alcuni paesi, i rivenditori autorizzati IBM possono contattare la relativa struttura di supporto e non il Centro di supporto IBM.

Riservato ai commenti del lettore

IBM DB2 Spatial Extender
Guida di riferimento per l'utente
Versione 7

Pubblicazione N. SC13-2902-00

Commenti relativi alla pubblicazione in oggetto potranno contribuire a migliorarla. Sono graditi commenti pertinenti alle informazioni contenute in questo manuale ed al modo in cui esse sono presentate. Si invita il lettore ad usare lo spazio sottostante citando, ove possibile, i riferimenti alla pagina ed al paragrafo.

Si prega di non utilizzare questo foglio per richiedere informazioni tecniche su sistemi, programmi o pubblicazioni e/o per richiedere informazioni di carattere generale.

Per tali esigenze si consiglia di rivolgersi al punto di vendita autorizzato o alla filiale IBM della propria zona oppure di chiamare il "Supporto Clienti" IBM al numero verde 167-017001.

I suggerimenti ed i commenti inviati potranno essere usati liberamente dall'IBM e dalla Selfin e diventeranno proprietà esclusiva delle stesse.

Commenti:

Si ringrazia per la collaborazione.

Per inviare i commenti è possibile utilizzare uno dei seguenti modi.

- Spedire questo modulo all'indirizzo indicato sul retro.
- Inviare un fax al numero: +39-081-660236
- Spedire una nota via email a: translationassurance@selfin.it

Se è gradita una risposta dalla Selfin, si prega di fornire le informazioni che seguono:

Nome

Indirizzo

Società

Numero di telefono

Indirizzo e-mail

Indicandoci i Suoi dati, Lei avrà l'opportunità di ottenere dal responsabile del Servizio di Translation Assurance della Selfin S.p.A. le risposte ai quesiti o alle richieste di informazioni che vorrà sottoporci. I Suoi dati saranno trattati nel rispetto di quanto stabilito dalla legge 31 dicembre 1996, n.675 sulla "Tutela delle persone e di altri soggetti rispetto al trattamento di dati personali". I Suoi dati non saranno oggetto di comunicazione o di diffusione a terzi; essi saranno utilizzati "una tantum" e saranno conservati per il tempo strettamente necessario al loro utilizzo.



Selfin S.p.A.
Translation Assurance

Via F. Giordani, 7

80122 NAPOLI



Numero parte: CT7C0IT

Printed in Denmark by IBM Danmark A/S

SC13-2902-00



(1P) P/N: CT7C0IT



Spine information:



IBM DB2 Spatial Extender

DB2 Spatial Extender - Guida di riferimento
per l'utente

Versione 7