

IBM® DB2® Spatial Extender



使用與參考手冊

版本 7

IBM® DB2® Spatial Extender



使用與參考手冊

版本 7

請注意！

使用本資訊及其支援的產品之前，請先閱讀第327頁的『注意事項』的一般資訊。

第二版 (2001 年 6 月)

除非在新版本中另外指出，否則本版將適用於 IBM DB2 Spatial Extender 版本 7.2 及所有後續版次和修訂版。

© Copyright International Business Machines Corporation 1998, 2001. All rights reserved.

目錄

圖	vii	呼叫 Spatial Extender	28
表	ix	第3章 設置資源.	31
關於本書	xi	資源庫存	31
適合閱讀本書的人	xi	參考資料	31
慣例	xi	讓資料庫能執行空間作業的資源.	31
如何傳送意見	xi	為空間作業啓用資料庫.	32
<hr/>		建立空間參照系統	32
第1篇 使用 Spatial Extender.	1	關於座標和空間參照系統	33
第1章 關於 Spatial Extender	3	從「控制中心」建立空間參照系統	36
Spatial Extender 的目的.	3	第4章 定義一些空間直欄、將它們登記為	39
代表地理特徵的資料.	4	「層」,以及啓用地理編碼程式來維護它們.	39
以資料代表地理特徵的方法	4	關於空間資料類型	39
空間資料的性質	5	單一單元特徵的資料類型.	40
空間資料來自何處	6	多單元特徵的資料類型.	41
如何建立及使用 Spatial Extender GIS	7	全部特徵的資料類型	41
Spatial Extender 的介面與相關功能性	7	為表格定義空間直欄、將此直欄登記為	
為建立及使用 Spatial Extender GIS 而執行的		「層」,以及啓用地理編碼程式來維護它.	41
作業	8	限制.	43
實務範例:保險公司更新其 GIS	10	將概略表直欄登記為「層」	44
第2章 安裝 Spatial Extender.	15	第5章 在空間直欄大量輸入資料	45
DB2 Spatial Extender 架構	15	使用地理編碼程式	45
系統需求	16	關於地理編碼.	45
支援的作業系統	16	以批次模式執行地理編碼程式	48
必要的資料庫軟體	16	匯入和匯出資料.	49
磁碟空間需求.	17	關於匯入和匯出.	49
安裝適用於 Windows NT 及 Windows 2000		在資料庫層次將資料匯入新的或現存的表格	50
的 DB2 Spatial Extender	18	在表格層次將資料匯入現存的表格	52
安裝 DB2 Spatial Extender for AIX	19	將資料匯出到形狀檔	53
裝載 CD-ROM	19	第6章 建立空間索引	55
使用 SMIT 或 installp 指令	22	使用「控制中心」建立空間索引.	55
建立 DB2 Spatial Extender 案例環境	23	決定格線資料格大小	56
安裝驗證	23	第7章 擷取及分析空間資訊	57
範例程式的疑難排解秘訣	24	執行空間分析的方法	57
後置安裝注意事項	26	建置空間查詢.	57
下載 ArcExplorer	26	空間函數與 SQL	57
使用「DB2 Spatial Extender 地理編碼程式		空間述詞與 SQL	58
參照資料」及「資料與地圖」的 CD-ROM	26		

第8章 撰寫 Spatial Extender 的應用程式	61	座標和測量	134
使用範例程式	61	X 與 Y 座標	134
範例程式步驟	61	Z 座標和測量	134
<hr/>		ST_CoordDim 函數	135
第2篇 參考資料	67	內部、界限及外部	135
第9章 儲存程序	69	單純或不單純	135
db2gse.gse_disable_autogc	71	空白或非空白	136
db2gse.gse_disable_db	73	包封	136
db2gse.gse_disable_sref	74	維度	136
db2gse.gse_enable_autogc	75	空間參照系統識別字	137
db2gse.gse_enable_db	78	可引證的幾何與相關函數	137
db2gse.gse_enable_idx	79	Points	137
db2gse.gse_enable_sref	81	線串	138
db2gse.gse_export_shape	83	多邊形	140
db2gse.gse_import_sde	85	多點	141
db2gse.gse_import_shape	87	多線串	141
db2gse.gse_register_gc	90	多重多邊形	142
db2gse.gse_register_layer	92	顯示關係與比較、產生幾何及轉換值的格式之 函數	143
db2gse.gse_run_gc	98	顯示地理特徵之間關係或比較的函數	144
db2gse.gse_unregist_gc	100	從現存的幾何產生新幾何的函數	155
db2gse.gse_unregist_layer	101	轉換幾何值格式的函數	160
第10章 訊息	103	第14章 SQL 查詢的空間函數	165
「控制中心」傳回的訊息	103	巢狀	165
儲存程序傳回的訊息	103	將 ST_Geometry 轉換為次類型	166
空間函數傳回的訊息	112	將集合轉換成基本幾何	166
第11章 型錄概略表	119	AsShape	168
DB2GSE.COORD_REF_SYS	119	EnvelopesIntersect	169
DB2GSE.GEOMETRY_COLUMNS	120	GeometryFromShape	171
DB2GSE.SPATIAL_GEOCODER	120	Is3d	173
DB2GSE.SPATIAL_REF_SYS	121	IsMeasured	174
第12章 空間索引	123	LineFromShape	175
範例程式片斷	123	LocateAlong	177
B 樹狀結構索引	124	LocateBetween	179
建立空間索引的方法	124	M	181
如何產生空間索引	125	MLine FromShape	182
使用空間索引的準則	129	MPointFromShape	184
選取格線資料格大小	130	MPolyFromShape	186
選取層次數	130	PointFromShape	187
第13章 幾何與相關的空间函數	131	PolyFromShape	189
關於幾何	131	ShapeToSQL	191
內容與相關函數	133	ST_Area	193
類別	133	ST_AsBinary	195
		ST_AsText	196
		ST_Boundary	197

ST_Buffer	199
ST_Centroid	201
ST_Contains	202
ST_ConvexHull	204
ST_CoordDim	206
ST_Crosses	208
ST_Difference	210
ST_Dimension	211
ST_Disjoint	213
ST_Distance	215
ST_Endpoint	216
ST_Envelope	217
ST_Equals	219
ST_ExteriorRing	220
ST_GeometryN	222
ST_GeometryType	223
ST_GeomFromText	225
ST_GeomFromWKB	227
ST_InteriorRingN	229
ST_Intersection	234
ST_Intersects	236
ST_IsClosed	237
ST_IsEmpty	239
ST_IsRing	241
ST_IsSimple	242
ST_IsValid	243
ST_Length	245
ST_LineFromText	247
ST_LineFromWKB	248
ST_MLineFromText	250
ST_MLineFromWKB	251
ST_MPointFromText	253
ST_MPointFromWKB	254
ST_MPolyFromText	256
ST_MPolyFromWKB	257
ST_NumGeometries	258
ST_NumInteriorRing	259
ST_NumPoints	260
ST_OrderingEquals	261
ST_Overlaps	262
ST_Perimeter	264
ST_Point	265
ST_PointFromText	266
ST_PointFromWKB	267
ST_PointN	269
ST_PointOnSurface	270

ST_PolyFromText	271
ST_PolyFromWKB	272
ST_Polygon	274
ST_Relate	275
ST_SRID	277
ST_StartPoint	278
ST_SymmetricDiff	279
ST_Touches	281
ST_Transform	282
ST_Union	284
ST_Within	285
ST_WKBToSQL	286
ST_WKTToSQL	288
ST_X	289
ST_Y	290
Z	291

第15章 座標系統	293
座標系統的概觀	293
支援的長度單位	295
支援的角度單位	295
支援的旋轉橢圓體	296
支援的大地測量基準點	297
支援的中央子午線	299
支援的地圖投影	300
支援的圓錐投影	301
支援的方位或平面投影	301
支援的地圖投影參數	301

第16章 空間資料的檔案格式	303
OGC 知名文字表示法	303
OGC 知名二進位 (WKB) 表示法	308
數值類型定義	308
數值類型的 XDR (Big Endian) 編碼	309
數值類型的 NDR (Little Endian) 編碼	309
在 NDR 與 XDR 之間轉換	309
WKBGeometry 位元組串流說明	309
WKB 表示法的主張	311
ESRI 形狀表示法	311
XY 空間中的形狀類型	313
XY 空間中的測量形狀類型	316
XYZ 空間中的形狀類型	320

第3篇 後記	325
注意事項	327

商標 329
索引 **331**

連絡 IBM **337**
產品資訊 337



1. 代表地理特徵的表格橫列；表格橫列的地址資料代表地理特徵	4	26. LocateBetween	159
2. 有新增空間直欄的表格	5	27. ST_ConvexHull	160
3. 內含空間資料的表格是由來源資料取得	6	28. 使用區域尋找建築物覆蓋區	194
4. 含有新建空間資料的表格是從現存的空間資料所取得	7	29. 半徑 5 英哩範圍的緩衝區應用於一個點	200
5. 主從架構安裝	16	30. 使用 ST_Contains 確定全部建築物位於自己的用地內	203
6. 空間資料類型的階層	40	31. 使用 ST_Crosses 尋找流經危險廢棄物區域的水路	209
7. 10.0e0 格線層次的應用程式	126	32. 使用 ST_Disjoint 尋找不在 (相交) 危險廢棄物場所的建築物	214
8. 新增格線層次 30.0e0 和 60.0e0 的效果	128	33. 使用 ST_ExteriorRing 判斷一個島嶼海岸線長度	221
9. Spatial Extender 支援的幾何階層	132	34. 使用 ST_InteriorRingN 判斷島嶼內的湖岸長度	229
10. 線串物件	139	35. 使用 ST_Intersection 判斷危險廢棄物可能影響每一棟建築物的區域大小	235
11. 多邊形	140	36. 使用 ST_Length 判斷一個縣的水路總長	246
12. 多線串	142	37. 使用 ST_Overlaps 判斷至少有一部份是在危險廢棄物區域內的建築物	263
13. 多重多邊形	143	38. 使用 ST_SymmetricDiff 判斷不含敏感區域 (有人居住的建築物) 的危險廢棄物區域	280
14. ST_Equals	146	39. 使用 NDR 格式的表示	311
15. ST_Disjoint	147	40. 具有一個孔和 8 個頂點的多邊形	315
16. ST_Touches	149	41. 多邊形位元組串流內容	316
17. ST_Overlaps	150		
18. ST_Within	153		
19. ST_Contains	154		
20. 兩個都市間的最小距離	155		
21. ST_Intersection	156		
22. ST_Difference	157		
23. ST_Union	157		
24. ST_Buffer	158		
25. LocateAlong	159		

表

1. 磁碟空間需求	17	24. db2gse.gse_register_gc 儲存程序的輸入參數	90
2. 資料 & 地圖 CD-ROM 資訊	26	25. db2gse.gse_register_gc 儲存程序的輸出參數	91
3. 空間函數與作業	57	26. db2gse.gse_register_layer 儲存程序的輸入參數	93
4. 索引開發規則	59	27. db2gse.gse_register_layer 儲存程序的輸出參數	97
5. Spatial Extender 範例程式	62	28. db2gse.gse_run_gc 儲存程序的輸入參數	98
6. db2gse.gse_disable_autogc 儲存程序的輸入參數	71	29. db2gse.gse_run_gc 儲存程序的輸出參數	99
7. db2gse.gse_disable_autogc 儲存程序的輸出參數	72	30. db2gse.gse_unregist_gc 儲存程序的輸入參數	100
8. db2gse.gse_disable_db 儲存程序的輸出參數	73	31. db2gse.gse_unregist_gc 儲存程序的輸出參數	100
9. db2gse.gse_disable_sref 儲存程序的輸入參數	74	32. db2gse.gse_unregist_layer 儲存程序的輸入參數	102
10. db2gse.gse_disable_sref 儲存程序的輸出參數	74	33. db2gse.gse_unregist_layer 儲存程序的輸出參數	102
11. db2gse.gse_enable_autogc 儲存程序的輸入參數	75	34. 空間函數傳回的訊息 SQLSTATE 值與 SQLCODE 值	116
12. db2gse.gse_enable_autogc 儲存程序的輸出參數	77	35. DB2GSE.COORD_REF_SYS 型錄概略表中的直欄	119
13. db2gse.gse_enable_db 儲存程序的輸出參數	78	36. DB2GSE.GEOMETRY_COLUMNS 型錄概略表中的直欄	120
14. db2gse.gse_enable_idx 儲存程序的輸入參數	79	37. DB2GSE.SPATIAL_GEOCODER 型錄概略表中的直欄	120
15. db2gse.gse_enable_idx 儲存程序的輸出參數	80	38. DB2GSE.SPATIAL_REF_SYS 型錄概略表中的直欄	121
16. db2gse.gse_enable_sref 儲存程序的輸入參數	81	39. 範例幾何的 10.0e0 格線資料格登錄	126
17. db2gse.gse_enable_sref 儲存程序的輸出參數	82	40. 三層索引中的幾何交點	128
18. db2gse.gse_export_shape 儲存程序的輸入參數	83	41. ST_Within 的矩陣	145
19. db2gse.gse_export_shape 儲存程序的輸出參數	84	42. 相等矩陣	146
20. db2gse.gse_import_sde 儲存程序的輸入參數	85	43. ST_Disjoint 的矩陣	147
21. db2gse.gse_import_sde 儲存程序的輸出參數	86	44. ST_Intersects (1) 的矩陣	148
22. db2gse.gse_import_shape 儲存程序的輸入參數	88	45. ST_Intersects (2) 的矩陣	148
23. db2gse.gse_import_shape 儲存程序的輸出參數	89	46. ST_Intersects (3) 的矩陣	148
		47. ST_Intersects (4) 的矩陣	148
		48. ST_Touches (1) 的矩陣	149
		49. ST_Touches (2) 的矩陣	150
		50. ST_Touches (3) 的矩陣	150
		51. ST_Overlaps (1) 的矩陣	150

52.	ST_Overlaps (2) 的矩陣	151	66.	幾何類型和它們的文字表示法	305
53.	ST_Crosses (1) 的矩陣	151	67.	點位元組串流內容	313
54.	ST_Crosses (2) 的矩陣	152	68.	多點位元組串流內容	313
55.	ST_Within 的矩陣	153	69.	折線位元組串流內容	314
56.	ST_Contains 的矩陣	154	70.	多邊形位元組串流內容	316
57.	等於型樣矩陣	275	71.	PointM 位元組串流內容	316
58.	支援的長度單位	295	72.	MultiPointM 位元組串流內容	317
59.	支援的角度單位	295	73.	PolyLineM 位元組串流內容	318
60.	支援的旋轉橢圓體	296	74.	PolygonM 位元組串流內容	320
61.	支援的大地測量基準點	297	75.	PointZ 位元組串流內容	320
62.	支援的中央子午線	299	76.	MultiPointZ 位元組串流內容	321
63.	支援的地圖投影	300	77.	PolyLineZ 位元組串流內容	322
64.	支援的圓錐投影	301	78.	PolygonZ 位元組串流內容	324
65.	支援的地圖投影參數	301			

關於本書

本書分成兩部份。第一個部份包含關於 DB2 Spatial Extender 的概念資訊，以及說明在 Windows NT 和 AIX 系統如何安裝、架構、管理和規劃 Spatial Extender。第二個部份由關於與 Spatial Extender 一起使用的儲存程序、幾何、函數、訊息和型錄概略表的參考資訊組成。

文字左邊的直線表示技術變更。

適合閱讀本書的人

本書適合負責設置空間環境的管理者以及使用空間資料開發應用程式的程式設計師閱讀。

慣例

本書使用這些強調標示慣例：

粗體字類型

指出指令及圖形式使用者介面 (GUI) 控制項 (例如，欄位名稱、資料夾名稱、功能表選項)。

等寬字體類型

指出撰寫程式碼或所鍵入文字範例。

斜體字 指出應以值取代的變數。斜體字類型也表示書籍標題及強調文字。

大寫字母

指出 SQL 關鍵字及物件名稱 (例如，目錄、概略表及伺服器)。

如何傳送意見

您的回饋可幫助 IBM 提供高品質資訊。您對本書或其它 DB2 文件若有任何意見，請來信指教。您可使用下列方法來提供意見：

- 從 Web 傳送您的意見。您可以存取 IBM Data Management 線上讀者意見表，網址是 <http://www.ibm.com/software/data/rcf>。
- 透過電子郵件將意見傳送至 comments@vnet.ibm.com。請包含產品名稱、產品版本號碼以及書名和書號 (若有的話)。若對特定文字有意見，請包含該文字位置 (例如，章節標題、表格號碼、頁碼或說明主題標題)。

第1篇 使用 Spatial Extender

第1章 關於 Spatial Extender

本章以解釋其目的、討論其處理的資料及說明其使用方法來介紹 Spatial Extender。以介紹本書其餘的簡易指南作為本章結尾。

Spatial Extender 的目的

使用 Spatial Extender 來建立地理資訊系統 (GIS)：它綜合了物件、資料、及可讓您建立及分析有關地理特徵之空間資訊的應用程式。地理特徵包括構成地球表面的物件及佔用在其上之物件。它們組成了自然環境 (如河川、森林、丘陵和沙漠) 及人文環境 (都市、住宅、辦公大樓、地標等)。

空間資訊包括下列事實：

- 和環境相關之地理特徵的位置 (例如，都市裡醫院和診所的位置，或接近當地震帶的都市住宅區)
- 地理特徵間彼此相關的方式 (例如，含括在特定的範圍裡某些河川系統的資訊，或該範圍裡某些橫跨河川系統支流的橋樑)
- 引用至一或多個地理特徵的測量 (例如，辦公大樓之間的距離及其地分界線，或鳥類保護區週圍的長度)

不論是空間資訊本身或和傳統關聯性資料庫管理系統 (RDBMS) 輸出資訊組合在一起，都可幫助您設計專案及作出生意上和政策上的決策。例如，假設縣福利行政區的主管需要驗證哪些福利的申請者和領受者是實際住在該行政區服務的區域裡。Spatial Extender 可從服務區域的位置及申請者和領受者的住址中取得此資訊。

或假設一家連鎖餐廳的老闆想要在附近都市作生意。要決定在何處開新餐廳的話，該老闆需要回答諸如此類的問題：這些都市裡何處是基本上會常到我餐廳的顧客所聚集之處？主要公路在何處？何處的犯罪率最低？競爭餐廳所在地點在何處？Spatial Extender 可以用視覺顯示的方式產生空間資訊來回答這些問題，而基礎的 RDBMS 可以建立標籤及文字來解譯這些顯示畫面。

本書中出現數個其它使用 Spatial Extender 的範例，尤其在第57頁的『第7章 擷取及分析空間資訊』、第61頁的『第8章 撰寫 Spatial Extender 的應用程式』和第165頁的『第14章 SQL 查詢的空間函數』。

代表地理特徵的資料

本節提供您所建立、儲存及操作之資料的概觀以取得空間資訊。涵蓋的主題有：

- 以資料代表地理特徵的方法
- 空間資料的性質
- 產生空間資料的方法

以資料代表地理特徵的方法

在 Spatial Extender 裡，地理特徵可由表格或概略表裡的橫列代表，或以部份此橫列代表。例如，就以在第3頁的『Spatial Extender 的目的』中所提到的兩個地理特徵而論，辦公大樓和住宅。在圖1裡，BRANCHES 表格的每一個橫列代表一個銀行分行。以整體的變異來看，在圖1的 CUSTOMERS 表格代表一位銀行顧客。然而，每一列的一部份--尤其是包含客戶地址的資料格-- 可被視為代表客戶住宅。

BRANCHES

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

CUSTOMERS

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

圖 1. 代表地理特徵的表格橫列；表格橫列的地址資料代表地理特徵。BRANCHES 表格裡的資料橫列代表一家銀行分行。CUSTOMERS 表格裡的地址資料格代表客戶住處。兩個表格裡的姓名和地址都是虛構的。

在圖1裡的表格含有定義及說明銀行分行和客戶的資料。這種資料稱為屬性資料。

屬性資料的子集--表示分行及客戶地址的值--可轉換為產生空間資訊的值。例如，如顯示於圖1，一家分公司的地址是 92467 Airzone Blvd., San Jose CA 95141。一位客戶的地址是 9 Concour Circle, San Jose CA 95141。Spatial Extender 可把這些地址轉換為指出分行及客戶所在與環境相關位置的值。第5頁的圖2以指定包含這些值的新直行來顯示 BRANCHES 和 CUSTOMERS 表格。

BRANCHES

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

CUSTOMERS

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

圖 2. 有新增空間直欄的表格. 在每一個表格裡，LOCATION 直欄會包含與地址相對應的座標。

當地址及類似識別字被用來當作空間資訊的起始點時，稱之為來源資料。因為從來源資料衍生的值會產生空間資訊，所以這些衍生值被稱為空間資料。下一節會說明空間資料並介紹它的關聯資料類型。

空間資料的性質

許多空間資料是由座標組成。座標是一個表示和參照點相對位置的數字。例如，緯度是表示和赤道相對位置的座標。經度是表示和格林威治子午線相對位置的座標。因此，黃石國家公園的位置是以其緯度 (赤道以北 44.45度) 以及經度 (格林威治子午線以西 110.40 度) 來定義。

緯度、經度、它們的參照點及其它相關參數通稱為座標系統。除了緯度和經度之外，以值為基礎的座標系統也會存在。這些座標系統有自己的測量位置、參照點、以及其它的區分參數。

最簡單的空間資料項目是由兩個座標所組成，定義單一地理特徵位置。(資料項目佔用關聯式表格資料格的值。) 更大規模的空間資料項目是由數個座標所組成，定義如道路或河川可能形成的線條路徑。第三種是由定義區域周界線的座標所組成；例如，一片土地或泛濫平原的邊緣。這些以及其它類由 Spatial Extender 所支援的空間資料項目在第131頁的『第13章 幾何與相關的空間函數』中有更完整的說明。

每一個空間資料項目即是一個空間資料類型的案例。以兩個座標標記位置的資料類型為 ST_Point；以座標定義線條路徑的資料類型為 ST_LineString；以座標定義周長的資料類型為 ST_Polygon。這些類型和其它空間資料的資料類型是屬於單一階層的結構化類型。若要取得階層的概觀，請參閱第39頁的『關於空間資料類型』。

空間資料來自何處

您可以取得空間資料的方法：

- 從屬性資料取得
- 從其它空間資料取得
- 匯入

使用屬性資料作為來源資料

Spatial Extender 可從屬性資料取得空間資料，例如地址 (如在第4頁的『以資料代表地理特徵的方法』中所提到的)。此程序稱為**地理編碼**。若要看到包含的順序，視第5頁的圖2為『之前』圖片以及視圖3為『之後』圖片。第5頁的圖2顯示 BRANCHES 表格和 CUSTOMERS 表格都只有指定給空間資料的 NULL 值。假設 Spatial Extender 在這些表格內地理編碼地址以取得和地址對應的座標，然後把座標放入直欄內。圖3會說明結果。

BRANCHES

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

CUSTOMERS

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

圖 3. 內含空間資料的表格是由來源資料取得。在 CUSTOMERS 表格裡的 LOCATION 直欄包含由地理編碼程式從 ADDRESS、CITY、STATE、和 ZIP 直欄內的地址所取得的座標。同樣地，在 BRANCHES 表格裡的 LOCATION 直欄包含由地理編碼程式從這個表格的 ADDRESS、CITY、STATE、和 ZIP 直欄內之地址所取得的座標。此範例為虛構的；所顯示的為模擬座標而非實際的座標。

Spatial Extender 使用一個名為**地理編碼**程式的函數，把屬性資料轉換為空間資料，然後將此空間資料放入表格直欄。若要取得有關地理編碼程式的更多資訊，請參閱第45頁的『關於地理編碼』。

使用其它空間資料作為來源資料

空間資料不但可從屬性資料建立也可從其它空間資料建立。例如，假設定義於 BRANCHES 表格內的銀行分行想要知道有多少客戶是位在每一個分行的五英哩範圍之內。在銀行能從資料庫取得此資訊之前，它必須提供有位於每一個分行周圍五英哩半徑內區域定義的資料庫。Spatial Extender 函數 ST_Buffer 能建立這樣的

定義。ST_Buffer 使用每一個分行的座標作為輸入，所以能建立區分所要區域周界線的座標。圖4會顯示含有由 ST_Buffer 所提供資訊的 BRANCHES 表格。

BRANCHES

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

圖 4. 含有新建空間資料的表格是從現存的空間資料所取得。在 SALES_AREA 直欄內的座標是從 LOCATION 直欄內座標的 ST_Buffer 函數所取得。就像 LOCATION 直欄內的座標一樣，在 SALES_AREA 直欄內的那些座標是模擬的；它們不是實際的座標。

除了 ST_Buffer 之外，Spatial Extender 也提供數個其它可從現存空間資料取得新的空間資料的函數。若要取得 ST_Buffer 以及這些其它函數的說明，請參閱第155頁的『從現存的幾何產生新幾何的函數』。

匯入空間資料

取得空間資料的第三種方法是從 Spatial Extender 所支援格式的檔案匯入。若要取得這些格式的說明，請參閱第303頁的『第16章 空間資料的檔案格式』。這些檔案通常包含用於地圖：人口調查、沖積平原、地震斷層等。藉由將這些資料搭配您產生的空間資料，可增加參考的地理資訊。例如，若公共工程部門需要判斷居住的社區容易發生哪些意外事件，可以使用 ST_Buffer 定義一個包圍社區的區域。然後，公共工程部門可匯入有關沖積平原與地震斷層的資料，查看與此區域重疊的是哪些問題區域。

如何建立及使用 Spatial Extender GIS

您可以藉由設置 Spatial Extender 並在 Spatial Extender 及其基礎 DB2 RDBMS 組合環境內開發 GIS 專案，來建立一個 Spatial Extender GIS。使用 GIS 的方法是實施這些專案；亦即，藉由產生及分析空間和傳統資訊，這些專案就是設計來提供此資訊。整個工作包括執行數組作業。本節說明執行這些作業的介面、提供作業概觀並提出一個實務範例來說明這些作業。

Spatial Extender 的介面與相關功能性

本節提出一些介面，您可藉由這些介面建立 Spatial Extender GIS (亦即，為它設置資源、取得空間資料等) 以及使用它 (亦即，產生及分析有關地理特徵的資訊) 的介面。

您可以使用下列方法來建立 Spatial Extender GIS：

- 使用「DB2 控制中心」的 Spatial Extender 視窗及功能表選項。相關指示，請參閱：
 - 第31頁的『第3章 設置資源』
 - 第39頁的『第4章 定義一些空間直欄、將它們登記為「層」，以及啓用地理編碼程式來維護它們』
 - 第45頁的『第5章 在空間直欄大量輸入資料』
 - 第55頁的『第6章 建立空間索引』
- 執行一種將呼叫 Spatial Extender 儲存程式的應用程式。關於開發這類程式的準則，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。
- 使用「控制中心」與應用程式。例如，您可以使用「控制中心」來呼叫預設地理編碼程式。此外，若您想使用另一個地理編碼程式，則必須呼叫應用程式中的 db2gse.gse_register_gc 儲存程序，先將它登記給 Spatial Extender。（關於非預設地理編碼程式的資訊，請參閱第45頁的『關於地理編碼』。關於 db2gse.gse_register_gc 儲存程序的資訊，請參閱第90頁的『db2gse.gse_register_gc』。）
- 使用「控制中心」、應用程式或兩者，搭配其它介面使用。例如，要建立一個表格來保留某空間函數（例如，地理編碼程式）產生的資料，可以使用「指令行處理器」或「控制中心」介面。

您可以透過下列方法來使用 Spatial Extender GIS：

- 使用地理瀏覽器來轉換圖形資訊；例如「環境系統研究學會 (ESRI)」提供的 ArcExplorer Version 3.0
- 從「DB2 控制中心」或「指令行處理器」明確提出 SQL 查詢
- 從應用程式提出 SQL 查詢

為建立及使用 Spatial Extender GIS 而執行的作業

本節提供您建立及使用 Spatial Extender GIS 的作業概觀。建立 GIS 的作業包括設置 Spatial Extender 及開發 GIS 專案。為使用 GIS 而執行的作業包括實施專案。此概觀首先提到設置 Spatial Extender，然後是開發及實施 GIS 專案。本節結論指出此概觀中所描述的作業在實際作法中可能不同。

設定 Spatial Extender

設置 Spatial Extender：

1. 規劃與準備（決定要開發哪些 GIS 專案、決定要對 Spatial Extender 啓用哪些資料庫、選取管理 Spatial Extender 及開發專案的人員等）。
2. 安裝 Spatial Extender。
3. 放置支援 GIS 專案的適當資源；例如：

Spatial Extender 提供的資源

這些資源包括系統型錄、空間資料類型、空間函數 (包括預設的地理編碼程式) 等。設置這些資源的作業稱為為空間作業啟用資料庫。

使用者、供應商或兩者開發的地理編碼程式。

預設地理編碼程式可將美國地址轉換成空間資料。您的組織及其它組織可提供地理編碼程式，將外國地址及其它種類的屬性資料轉換成空間資料。

關於安裝 Spatial Extender 的指示，請參閱第15頁的『第2章 安裝 Spatial Extender』。關於使用「控制中心」將資源放置到適當位置的說明，請參閱第31頁的『第3章 設置資源』。關於使用應用程式將資源放置到適當位置的準則，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。如需有關設置 Spatial Extender 的整體工作之實務範例，請參閱第11頁的『整合空間及傳統資料的系統』。

開發與實施 GIS 專案

開發與實施 GIS 專案：

1. 規劃與準備 (設定專案目標、決定所需的表格和資料、決定所要使用的座標系統等)。
2. 決定所要使用的空間參照系統。一般而言，座標值包括正整數、負數及小數。不過，Spatial Extender 必須以正整數形式來儲存所有座標值。空間參照系統是一組參數，定義特定座標系統中的負數和小數如何轉換成正整數，使 Spatial Extender 能夠儲存它們。決定空間直欄所要使用的座標系統之後，您必須指定空間參照系統，以便對該直欄產生必要的轉換。若現存的空間參照系統符合您的需求，您可以使用它；否則，您可以建立一個空間參照系統。
3. 定義一或多個直欄來包含空間資料、將它們登記給 Spatial Extender，並啓用地理編碼程式來自動維護它們。

登記空間直欄的工作之一是將它記錄到 Spatial Extender 型錄中。從您登記空間直欄開始，它就稱為一個層，因為空間直欄產生的資訊會新增一個層 (stratum 或 layer) 到 GIS 為您建立的虛擬地理景觀中。登記空間直欄之後，您就可以對它執行一些空間作業；例如，大量輸入資料到空間直欄中，然後對它定義空間索引。

4. 大量輸入資料到空間直欄中：
 - 針對需要地理編碼程式的專案，請為地理編碼程式設定一些參數。然後，在單一作業中執行它，使它為所有可用的來源資料做地理編碼，並將結果座標載入一個層中。
 - 針對需要匯入空間資料的專案，請輸入資料。

5. 方便存取空間直欄。尤其包括定義索引讓 DB2 快速地存取空間資料，以及定義概略表讓使用者有效地擷取相關的資料。在定義這樣的一個概略表之後，您必須將它的空間直欄登記為「層」。
6. 建立及分析空間資訊和相關的業務資訊。這包括查詢空間直欄和相關的屬性直欄。在這類查詢中，您可以併入傳回各種不同資訊的 Spatial Extender 函數；例如，兩地理特徵之間的最小距離，或定義包圍一個地理特徵的區域之座標。關於傳回此類座標的函數 ST_Buffer 之資訊，請參閱第6頁的『使用其它空間資料作為來源資料』和第199頁的『ST_Buffer』。關於使用空間函數的查詢之範例，請參閱第57頁的『第7章 擷取及分析空間資訊』和第165頁的『第14章 SQL 查詢的空間函數』。

關於使用「控制中心」來執行涉及開發 GIS 專案的作業之相關指示，請參閱：

- 第31頁的『第3章 設置資源』
- 第39頁的『第4章 定義一些空間直欄、將它們登記為「層」，以及啓用地理編碼程式來維護它們』
- 第45頁的『第5章 在空間直欄大量輸入資料』
- 第55頁的『第6章 建立空間索引』

關於使用「控制中心」來實施 GIS 專案的準則，請參閱第57頁的『第7章 擷取及分析空間資訊』。

關於使用應用程式來開發及實施 GIS 專案的準則，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

如需說明整體工作的實務範例，請參閱第11頁的『建立分公司及調整保險費的專案』。

作業組的差別

根據您的需求以及使用的介面，建立及使用 Spatial Extender GIS 所執行的作業組有不同的內容和順序。例如，定義一些直欄來包含空間資料、將它們登記為「層」，並啓用地理編碼程式自動維護它們等作業。透過「控制中心」，您可以從單一視窗中同時執行這些作業。不過，若您是透過程式來呼叫儲存程序，則可以分別執行這些作業，而且可以自行決定何時執行。

實務範例：保險公司更新其 GIS

本節顯示一個實務範例，說明上節中描述的作業組。

Safe Harbor Real Estate Insurance Company 的資訊系統環境，包括 DB2 Universal Database 系統及個別的 GIS 資料庫管理系統。從某種程度來說，查詢可從這兩個系統中擷取資料組合。例如，DB2 表格儲存關於收入的資訊，GIS 表格儲存的是公

司各分公司的位置。因此，您可以找到有一定收入的分公司位置。但無法整合這兩個系統中的資料（例如，使用者無法將 DB2 直欄與 GIS 直欄結合在一起），而且 DB2 服務（例如查詢最佳化）無法使用於 GIS。為克服這些缺點，Safe Harbor 需要取得 Spatial Extender 並建立一個新的 GIS 開發部門。以下各節說明該部門如何設置 Spatial Extender 及完成第一個專案。

整合空間及傳統資料的系統

若要設置 Spatial Extender，Safe Harbor 的 GIS 開發部門要以下列方式進行：

1. 部門準備將 Spatial Extender 併入它的 DB2 環境內。例如：
 - a. 部門的管理小組要指定一個空間管理小組來安裝及實施 Spatial Extender，另外指定一個空間分析小組來建立及分析空間資訊。
 - b. 由於 Safe Harbor 的商業決策主要取決於客戶需求，所以管理小組決定在含有客戶相關資訊的資料庫中安裝 Spatial Extender。此資訊的絕大部份是儲存於一個稱為 CUSTOMERS 的表格中。
為了方便稱呼選取的資料庫，GIS 開發部門的成員稱它為 GIS 資料庫。不過，這些成員知道它不是保留給 GIS 專案專用；像從前一樣，非空間應用程式還是可以繼續使用它。
2. 空間管理小組要安裝 Spatial Extender。
3. 空間管理小組設置 GIS 專案需要的資源：
 - 小組使用「控制中心」提供讓 GIS 資料庫用於空間作業的資源。這些資源包括 Spatial Extender 型錄、空間資料類型、空間函數等。
 - 由於 Safe Harbor 開始將業務拓展到加拿大，所以空間管理小組開始要求加拿大供應商提供地理編碼程式，將加拿大的地址轉換成空間資料。

建立分公司及調整保險費的專案

為了完成 Spatial Extender 下的第一個 GIS 專案，GIS 開發部門以下列方式進行：

1. 部門準備開發專案；例如：
 - 管理小組設定專案目標：
 - 決定要建立新分公司的地點。
 - 以靠近危險區域（指交通意外發生率高的地區、犯罪率高的地區、洪水區、地震帶等）的客戶為基礎來調整保險費。
 - GIS 專案將與美國境內的客戶與公司有關聯。因此，空間管理小組決定：
 - 使用座標系統，精確定義美國境內與 Safe Harbor 有業務往來的位置。
 - 使用預設地理編碼程式，因為它是設計來地理編碼美國地址。
 - 空間管理小組決定哪些是符合專案目標的必要資料，以及哪些表格將包含此類資料。

2. 空間管理小組可使用「控制中心」來建立兩個空間參照系統。一個用來決定如何將定義公司位置的座標轉換成 Spatial Extender 可儲存的資料項目。另一個用來決定如何將定義客戶住所的座標轉換成 Spatial Extender 可儲存的資料項目。
3. 空間管理小組可使用「控制中心」來定義一些包含空間資料的直欄、將它們登記為「層」，並啓用地理編碼程式自動維護它們：
 - 此小組會將 LOCATION 直欄加入 CUSTOMERS 表格中。此表格已包含客戶的地址。預設的地理編碼程式會將它們轉換成空間資料，並將此資料載入 LOCATION 直欄中。
 - 此小組會建立 OFFICES 表格來包含目前儲存於個別 GIS 中的資料。此資料包括 Safe Harbor 的分公司地址、透過地理編碼程式而衍生自這些地址的空間資料，以及定義每一個分公司附近五英哩內一個區域的空間資料。地理編碼程式所產生的資料會載入 LOCATION 直欄中。定義這些區域的資料會載入 SALES_AREA 直欄中。
 - 此小組會將兩個 LOCATION 直欄及 SALES_AREA 直欄登記為「層」。
 - 此小組會啓用預設的地理編碼程式來自動維護這兩個 LOCATION 直欄。
4. 空間管理小組會大量輸入資料到 CUSTOMER 表格的 LOCATION 直欄、整個 OFFICES 表格，以及新的 HAZARD_ZONES 表格中：
 - 此小組使用「控制中心」來大量輸入資料到 CUSTOMER 表格的 LOCATION 直欄中：
 - a. 此小組指示地理編碼程式將代表某地址的空間資料插入 LOCATION 直欄中，但有下列條件限制：此地址必須與美國 Census Bureau 記錄的副本完全相同。(Spatial Extender 有檢附一個 Census Bureau 提供的地址檔。地理編碼程式在將來源資料中的地址轉換成空間資料之前，必須嘗試讓此地址符合檔案中的副本。使用者指定正確符合的百分率，滿足條件的空間資料才可以存放到表格中。此百分率稱為精確度。)
 - b. 此小組會以批次模式執行地理編碼程式，以便在一個作業中地理編碼表格內的所有地址。小組覺得遺憾的是，地理編碼程式大約每 10 個地址就會拒絕一個!
 - c. 此小組猜測所拒絕的是沒有完全符合 Census Bureau 記錄的新地址。為了解決這個問題，小組決定將精確度縮減為 85。
 - d. 此小組重新以批次模式執行地理編碼程式。地址被拒絕的比率降至一個可接受的層次。
 - 利用個別 GIS 提供的公用程式，此小組會將分公司資料載入一個檔案中。然後，使用「控制中心」將此檔案中的資料匯入新的 OFFICES 表格中。
 - 透過「控制中心」，此小組會建立 HAZARD_ZONES 表格、將它的空間直欄登記為「層」，並將資料匯入給它。此資料來自地圖供應商提供的檔案。

5. 透過「控制中心」，空間管理小組可以更方便存取新的「層」：
 - 此小組會對它們建立索引。
 - 此小組會建立一個概略表，將 CUSTOMERS 和 HAZARD ZONES 表格中的直欄結合在一起。然後，將概略表的空間直欄登記為「層」。
6. 空間分析小組執行查詢來取得一些協助它符合原始目標的資訊，這些原始目標包括：決定建立新分公司的地點，以及根據接近危險區域的客戶來調整保險費。

第2章 安裝 Spatial Extender

本章提供安裝適用於 AIX、Windows NT 及 Windows 2000 的 DB2 Spatial Extender 的指示。同時說明下列主題：

- DB2 Spatial Extender 架構
- 系統需求
- 安裝適用於 Windows NT 及 Windows 2000 的 DB2 Spatial Extender
- 安裝 DB2 Spatial Extender for AIX
- 安裝驗證
- 後置安裝注意事項
- 呼叫 Spatial Extender

DB2 Spatial Extender 架構

Spatial Extender 系統由 DB2 Universal Database、Spatial Extender 及地理瀏覽器 (例如 ArcExplorer Java Version 3.0) 所組成。一般而言，對空間作業啓用的資料庫是位於伺服器中。您可以使用從屬站應用程式，透過 Spatial Extender 儲存程序及空間查詢來存取空間資料。您也可以在獨立式環境中架構 DB2 Spatial Extender，這是從屬站與伺服器均常駐在相同機器上的一種架構。在主從架構及獨立式架構中，您可以使用地理瀏覽器來檢視空間資料。

IBM 目前並未分送可以產生視覺化查詢結果的地理瀏覽器。如需地理瀏覽器及如何取得地理瀏覽器的其餘相關資訊，請參閱第26頁的『下載 ArcExplorer』。

第16頁的圖5說明 Spatial Extender 的架構。

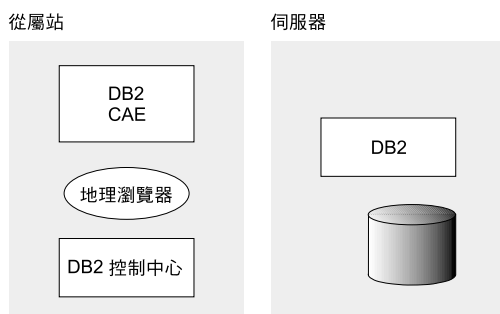


圖 5. 主從架構安裝

系統需求

本節說明 DB2 Spatial Extender 的軟硬體需求。

支援的作業系統

Spatial Extender 可以安裝在 AIX 版本 4.2（或更新的版本）、具有 Service Pack 5 的 Windows NT 版本 4.0（或更新的版本）及 Windows 2000 上。

註：如果您計劃使用 ArcSDE 來提交及檢視空間資料，則您會需要 AIX 版本 4.3.3 或更新的版本。

必要的資料庫軟體

安裝 DB2 Spatial Extender 之前，您必須在從屬站及伺服器上安裝及架構下列 DB2 軟體：

從屬站軟體

關於 DB2 Spatial Extender 從屬站產品：

- DB2 Administration Client 版本 7.1

如果您不打算使用「DB2 控制中心」、存取空間資料的地理瀏覽器或 DB2 Spatial Extender 範例程式，則您可以安裝及使用 DB2 Administration Client 版本 6.0。

- Fixpack 1

當您從 CD-ROM 安裝 DB2 Spatial Extender 從屬站時，即已在系統中自動安裝了具有 Fixpack 1 的 DB2 Administration Client。

重要事項: 如果已在從屬站上安裝了 DB2 Universal Database Enterprise Edition 版本 7.1 或 DB2 Universal Database Enterprise-Extended Edition 版本 7.1，則在您安裝 DB2 Spatial Extender 之前，必須先安裝 Fixpack 1。

伺服器軟體

關於 DB2 Spatial Extender 伺服器產品，在您安裝 DB2 Spatial Extender 之前，必須先在系統中安裝下列其中一項伺服器產品：

- DB2 Universal Database Enterprise Edition 版本 7.1 及 Fixpack 1
- DB2 Universal Database Enterprise-Extended Edition 版本 7.1 及 Fixpack 1

如果您計劃使用「DB2 控制中心」，請建立及架構 DB2 Administration Server (DAS)。如需建立及架構 DAS 的其餘相關資訊，請參閱 *IBM DB2 Universal Database Administration Guide: Implementation*。

註: 雖然您可以在 DB2 Universal Database Enterprise-Extended Edition 使用 DB2 Spatial Extender，但無法像在大量並行處理 (MPP) 環境中一樣在多重節點上分割空間索引。

磁碟空間需求

表1列示 Spatial Extender 磁碟空間需求。

表 1. 磁碟空間需求

Spatial Extender 產品	磁碟空間
DB2 Spatial Extender 的伺服器產品：	
<ul style="list-style-type: none">• Spatial Extender 伺服器檔案庫程式碼、範例地理編碼程式參照資料及文件• 在個別 CD-ROM 中的選用及可用項目：地理編碼程式參照資料（美國） 如需使用地理編碼程式參照資料的其餘相關資訊，請參閱第27頁的『DB2 Spatial Extender 地理編碼程式參照資料』。	594 MB 總磁碟空間： <ul style="list-style-type: none">• 31 MB（Spatial Extender 伺服器檔案庫程式碼、範例地理編碼程式資料及文件）• 563 MB（美國地理編碼程式參照資料）
DB2 Spatial Extender 的從屬站產品（包括範例程式資料）	1 MB

表 1. 磁碟空間需求 (繼續)

Spatial Extender 產品	磁碟空間
	<p>註：本表格中所計算的磁碟空間需求是假設您已在 Windows NT 或 Windows 2000 的一般安裝類型中安裝了 DB2 Universal Database 及 DB2 Spatial Extender，或在 AIX 中具有預先選取的元件。如果您是要安裝 DB2 Spatial Extender，或以不同的安裝類型安裝 DB2 Universal Database，則您的磁碟空間計算可能會有所不同。</p>

安裝適用於 Windows NT 及 Windows 2000 的 DB2 Spatial Extender

安裝適用於 Windows NT 及 Windows 2000 的 Spatial Extender：

1. 將 Spatial Extender CD-ROM 插入光碟機。即會開啓「DB2 發射台」，這是一個您可以在其中安裝 DB2 Spatial Extender 的介面。
2. 按一下**安裝**。安裝程式起始設定後，即會開啓「選取產品」視窗。

附註：

- a. 在安裝程序期間，隨時可按一下**取消**，以終止及結束安裝。
 - b. 如果您收到一則警告訊息，表示 DB2 目前正在執行中且被一些程序鎖定，則只有在資料庫未在使用中且沒有任何重要的使用者連線時，按一下**是**。DB2 將會關閉這些處理且不會嘗試儲存資料。如果您是在作用中系統上執行安裝時，請使用其它的管理方法來關閉鎖定的 DB2 處理。
 - c. 如果您是在獨立式環境中安裝 DB2 Spatial Extender，請在 Windows NT 及 Windows 2000 中，從安裝程式選取 **DB2 Administration Client**。安裝程式中的預設設定並**沒有**預先選取 **DB2 Administration Client**。
3. 從清單中選取您想要安裝的產品：
 - 如果您是要以獨立式架構安裝 DB2 Spatial Extender，請選取 **DB2 Spatial Extender Server** 及 **DB2 Administration Client**。
 - 如果您是要在伺服器平台上安裝 DB2 Spatial Extender，請選取 **DB2 Spatial Extender Server**。
 - 如果您是要在從屬站環境中安裝 DB2 Spatial Extender，請選取 **DB2 Administration Client**。

DB2 Administration Client 選項會安裝 DB2 Universal Database 子元件、Spatial Extender 範例程式及資料與「DB2 Universal Database 控制中心」。**DB2 Spatial Extender Server** 選項包含 DB2 Spatial Extender 伺服器程式庫程式碼、地理編碼程式參照資料的範例及文件。

註： DB2 Spatial Extender 伺服器產品只含有可用的地理編碼程式資料子集。在 DB2 Spatial Extender 隨附的個別 CD-ROM 中，含有完整的美國地理編碼程式參照資料集。

4. 按一下**下一步**。即會開啓「選取安裝類型」視窗。
5. 選取安裝類型。如果您選取**自訂**，則會開啓「選取元件」視窗。您可以選取您要安裝的元件。若要執行此動作，您必須瞭解 DB2 元件與設定值。
6. 按一下**下一步**以開啓「選擇目的地位置」視窗。
7. 選擇您要安裝 DB2 Spatial Extender 的資料夾。若要變更預設設定中的資料夾，請按一下**瀏覽**。

註： 如果 DB2 Universal Database 已經安裝在系統中，則您無法選擇在此視窗中選擇新位置或建立新資料夾。「選擇目的地位置」視窗會顯示下列：

- 安裝 DB2 Spatial Extender 的資料夾位置
 - 安裝 DB2 Spatial Extender 所需的磁碟空間
8. 按一下**下一步**以安裝 DB2 Spatial Extender。即會開啓「安裝進度」視窗，並顯示安裝程序的進度。

安裝 DB2 Spatial Extender for AIX

本節說明您在執行安裝 DB2 Spatial Extender for AIX 時的必要步驟。您可以使用 DB2 安裝程式（交談式或自動式）、系統管理介面工具 (SMIT) 或 **installp** 指令。本節中的主題包括：

- 裝載 CD-ROM
- 使用 DB2 安裝程式
- 使用 SMIT 及 **installp** 指令
- 建立 DB2 Spatial Extender 案例環境

裝載 CD-ROM

在執行各種 DB2 Spatial Extender 的 AIX 安裝步驟時，您必須裝載 CD-ROM。您可以在其中找到本書中所參照的 */cdrom*，請務必在起始該步驟前，先執行本節中所述的裝載程序。下列裝載指示適用於使用 DB2 安裝程式、SMIT 及 **installp** 指令所執行的安裝程序。

若要裝載 CD-ROM：

1. 以具有 Root 權限的使用者身份登入。
2. 將 CD-ROM 插入光碟機中。
3. 輸入下列指令以建立裝載 CD-ROM 的目錄：

```
mkdir -p /cdrom
```

其中 */cdrom* 是 CD-ROM 裝載目錄。

4. 輸入下列指令以配置 CD-ROM 檔案系統：

```
smitty storage
```
5. 選取**檔案系統**。
6. 選取**新增/變更/顯示/刪除檔案系統/**。
7. 選取 **CD-ROM 檔案系統**。
8. 選取**新增 CDROM 檔案系統**。
9. 選取**裝置名稱**。

CD-ROM 檔案系統的裝置名稱必須是唯一的。如果現存系統名稱與您想要選取的名稱重複，則請刪除現存的 CD-ROM 檔案系統或使用另一個目錄名稱。

10. 在蹦現視窗中，鍵入下列裝載點：

```
/cdrom
```

11. 鍵入下列指令以裝載 CD-ROM 檔案系統：

```
smit mountfs
```

12. 鍵入檔案系統名稱（例如，*/dev/cd0*）。
13. 鍵入目錄名稱，例如 (*cdrom*)。
14. 鍵入檔案系統類型，例如 (*cdrfs*)。
15. 將裝載「唯讀」系統設為**是**。
16. 登出。

使用 DB2 安裝程式

本節說明下列主題：

- 交談式安裝
- 自動式安裝作業
- 產生追蹤日誌或檔案

使用 DB2 安裝程式執行 AIX 安裝期間，您可以選擇要以交談式安裝或在自動式環境中安裝。在交談式安裝中，您會與一系列的螢幕畫面相互作用以設定及架構 DB2 Spatial Extender。在自動式安裝作業中，您可以使用在呼叫 DB2 安裝程式前

建立的回應檔來提供安裝及架構資訊。如果您需要在一台以上的機器中安裝 DB2 Spatial Extender，則您可以自行設定檔案並使用它在多重工作站上安裝產品。

以交談式安裝 DB2 Spatial Extender:

1. 以 root 使用者的身份登入目標從屬站或伺服器機器。
2. 將 CD-ROM 插入光碟機。
3. 鍵入 `cd /cdrom`。
4. 鍵入 `./db2setup`。開啓 DB2 安裝程式。
5. 選取您要安裝的產品：
 - 如果您要以獨立式架構來安裝 DB2 Spatial Extender，請選取 **Spatial Extender Server** 及 **Spatial Extender Client**。
 - 如果您是要在伺服器平台上安裝 DB2 Spatial Extender，請選取 **DB2 Spatial Extender Server**。
 - 如果您是要在從屬站環境中安裝 DB2 Spatial Extender，請選取 **DB2 Spatial Extender Client**。
 - 如果您想要附加的 DB2 從屬站支援，請選取 **DB2 Administration Client**。請使用 Tab 鍵來導覽您要選取的元件，並按下 Enter 鍵。每一個選取的產品都有一個自行設定視窗可供使用。
6. 針對您選擇的元件，選取國家語言。
7. 以 Tab 鍵跳至**確定**按鈕，然後按下 Enter 鍵以安裝 DB2 Spatial Extender。

以自動式安裝 DB2 Spatial Extender:

1. 建立自動式安裝作業的回應檔。
2. 啓動自動式安裝作業。

本節中會針對這些步驟有詳細的說明。

步驟 1. 建立自動式安裝作業的回應檔：

註：如果您接受範例回應檔中的預設值，則可以略過步驟 1 並繼續執行步驟 2。

1. 開啓您要安裝的產品範例回應檔。範例回應檔是位於 `/cdrom/db2/install/samples`，其中 `/cdrom` 是 DB2 Spatial Extender 的可安裝版本位置。有兩個範例回應檔，一個是用於 Spatial Extender 伺服器 (`db2gse.rsp`)，另一個是用於 Spatial Extender 從屬站 (`db2gsec.rsp`)。範例回應檔包含：
 - 安裝專用的關鍵字
 - 環境變數的登錄值/設定值
 - 資料庫管理程式架構參數

2. 啓動某一項目以變更回應檔中的值。若要啓動項目：
 - a. 除去鍵盤/環境變數左邊的星號 (*)。
 - b. 消除值右邊的現行設定。
 - c. 鍵入新的設定。
 - d. 如果您做了變更，請以新的檔名儲存檔案，以保留原始的範例回應檔。如果您是要從 CD-ROM 中直接安裝，則您必須在本端檔案系統中儲存更名的回應檔。

步驟 2. 使用回應檔啓動自動式安裝作業。

1. 以具有 Root 權限的使用者身份登入。
2. 輸入 **db2setup** 指令：

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

其中 */cdrom* 是 DB2 Spatial Extender 可安裝的映像檔的位置，*responsefile_directory* 是儲存回應檔的目錄，而 *responsefile_file* 是回應檔的名稱。

3. 安裝完成時，請檢查日誌檔中的訊息。日誌檔的預設位置是 */tmp/db2setup.log*。

產生 DB2 安裝程式的追蹤日誌檔： 如果您在使用 DB2 安裝程式時發生問題，您可以產生追蹤日誌檔 (*db2setup.trc*)。您可以將追蹤日誌檔及 *db2setup.log* 檔傳送給「IBM 軟體支援中心」以便能進一步的診斷。檔案產生時，兩個檔案都是儲存在 */tmp* 目錄中。

若要產生追蹤日誌檔，請執行 **db2setup** 指令並使用 **-d** 旗號，如下所示：*./db2setup -d*。這會觸發 DB2 安裝程式以追蹤模式執行。繼續操作介面，讓問題重新發生。當您完成時，即已建立追蹤日誌檔 */tmp/db2setup.trc*。

使用 SMIT 或 **installp** 指令

若要使用「系統管理介面工具 (SMIT)」或 **installp** 指令來安裝 DB2 Spatial Extender：

1. 以 root 使用者的身份登入目標從屬站或伺服器機器。
2. 將 CD-ROM 插入光碟機。
3. 執行 SMIT 或 **installp** 指令以安裝 DB2 Spatial Extender。

若要執行 SMIT：

- a. 輸入 **smit install_latest** 指令。即會開啓 SMIT 工具功能表。
- b. 在軟體欄位的輸入裝置/目錄中鍵入 */cdrom/db2*。
- c. 按一下**執行**或按下 **Enter** 鍵以驗證安裝目錄是否存在。

- d. 在**要安裝的軟體**欄位中，識別是否要安裝從屬站或伺服器元件。
若是從屬站，請鍵入下列：db2_07_01.gc1n。
若是伺服器，請鍵入下列：db2_07_01.gsrv。
- e. 按一下**執行**或 **Enter** 鍵（系統會提示您確認安裝參數）。
- f. 按下 **Enter** 鍵以確認。
產品檔案會從 CD-ROM 安裝到您的硬碟上，這可能需要幾分鐘的時間。
- g. 登出。

建立 DB2 Spatial Extender 案例環境

db2icrt 指令是用來建立新的 DB2 案例。您在安裝 DB2 Spatial Extender 後建立的所有新 DB2 案例，在案例環境中都包含 DB2 Spatial Extender。

在安裝 Spatial Extender 前所建立的 DB2 案例在其案例環境中不包含 DB2 Spatial Extender。若要使用 Spatial Extender 更新現存的 DB2 案例，請使用 **db2iupdt** 指令。

請以具有 Root 權限的使用者身份鍵入下列指令：

```
db2iupdt instance_name
```

instance_name 參數是案例的名稱。

在 AIX 中，此公用程式是位於下列目錄中：`/usr/lpp/db2_07_01/instance`。若需要說明，請在指令行鍵入 `db2iupdt -h` 來開啓一個說明功能表。

註：在您驗證安裝之前，請先使用 Spatial Extender 更新案例環境。

安裝驗證

安裝 DB2 Spatial Extender 之後，請建立資料庫並執行安裝檢查程式，以驗證是否已正確地安裝及架構 DB2 Spatial Extender。

註：對於 AIX 安裝，請在您執行安裝檢查程式之前，先檢查是否已建立 DB2 Spatial Extender 案例環境。

您可以使用 DB2 Spatial Extender 範例程式 (runGseDemo) 來驗證安裝。您可以使用 DB2 工具或使用「DB2 控制中心」中的使用者介面，從指令行來變更資料庫架構參數。下列指示適用於 AIX、Windows NT 及 Windows 2000。

若要驗證安裝：

1. 以案例擁有者的身份登入（僅適用於 AIX）。

- 請以最小值 2048 增加 UDF 記憶體使用大小的資料庫管理程式架構。例如，鍵入 `db2 update dbm cfg using UDF_MEM_SZ 2048`。如果 2048 不適用，請以 256 來遞增 `UDF_MEM_SZ` 參數。

註：當應用程式中的 UDF 參照數增加時，即會增加 UDF 記憶體使用大小的記憶體需求。特別是在具有空間資料類型的 UDF 是當成輸入及/或輸出參數使用時，此情況為真。

- 建立資料庫。例如，鍵入 `db2 create database mydb`，其中 *mydb* 是資料庫名稱。
- 增加資料庫的 DB2 日誌檔大小。
若要增加日誌檔大小：
 - 連接至您建立的資料庫。例如，鍵入 `db2 connect to mydb`，其中 *mydb* 是資料庫名稱。
 - 增加日誌檔大小。例如，鍵入 `db2 update db logfilesize using LOGFILE 1000`。
 - 切斷資料庫的連線。例如，鍵入 `db2 connect reset`。

註：每一次您啓用資料庫的空間作業，就必須增加 DB2 日誌檔大小。

- 尋找安裝檢查。例如，鍵入 `runGseDemo`。
若是 AIX，鍵入 `cd $HOME/sqllib/samples/spatial`，其中 *\$HOME* 是案例擁有者的起始目錄。
若是 Windows NT 及 Windows 2000，請鍵入 `cd c:\sqllib\samples\spatial`，其中 *c:\sqllib* 是您安裝 DB2 Spatial Extender 的目錄。
- 執行安裝檢查程式。例如，鍵入：

```
runGseDemo mydb userID password
```


參數 *mydb* 是資料庫名稱。

範例程式的疑難排解秘訣

DB2 範例程式的設計原意在於呈現安裝時的問題。安裝驗證時，您可能會收到一些錯誤訊息，這些訊息可以協助您診斷特定的系統問題。大部份的錯誤訊息都是由少數的一般使用者錯誤所造成的。若要避免這些錯誤，請務必在每一次執行安裝檢查程式時執行下列動作：

- 請確定您是否已在適當的環境中安裝了 DB2 Spatial Extender 從屬站及伺服器產品。對於獨立式架構，請確定是否已安裝了從屬站及伺服器產品兩者。
- 使用與任何空間作業無關的新資料庫。
- 增加 UDF 記憶體使用大小的資料庫管理程式架構。

- 增加日誌檔大小。

管理從屬站

如果在您安裝 DB2 Spatial Extender 時，沒有選取 **DB2 Administration Client** 選項（指 NT 安裝）或 **Spatial Extender Client** 選項（指 AIX 安裝），則會收到下列錯誤訊息：「無法辨識指定的名稱是否為內部或外部指令、可操作的程式或批次檔。」

由於在您的系統中無法使用範例程式，所以才會發生此錯誤。範例程式是與 DB2 Administration Client 及 Spatial Extender Client 包裝在一起的。如果沒有在系統中安裝 DB2 Administration Client 或 Spatial Extender Client，就無法在系統中使用範例程式。

若要修正此問題：

1. 重新安裝 DB2 Spatial Extender。請從 Windows NT 及 Windows 2000 的安裝程式中選取 **DB2 Administration Client**，或從 AIX 的 DB2 安裝程式中選取 **Spatial Extender Client**。
2. 重複第23頁的『安裝驗證』中的步驟以重新執行範例程式。

已經啓用資料庫的空間作用

如果您正在其中執行範例程式的資料庫已經啓用空間作業，則您會收到下列錯誤訊息：

```
啓用資料庫 logtst...  
從 ENABLE_DB 傳回：  
回覆碼 = -14  
傳回訊息文字 =  
GSE0014E 資料庫已為空間作業而啓用。
```

若要修正此問題，請捨棄資料庫並重複第23頁的『安裝驗證』中的步驟。

註：請檢查資料庫並驗證安裝是否是新的且沒有任何相關聯的空間作業；如果有的話，則範例程式將會失敗。

資料庫管理程式架構

如果您無法在資料庫管理程式架構中增加 UDF 記憶體使用大小，則會收到下列錯誤訊息：

```
發生非預期的 SQL 錯誤（“SQL0973N 在 "UDF_MEM" 資料堆沒有足夠的  
儲存體可用來處理陳述式。”）。 SQLSTATE=57011
```

如需如何增加資料庫管理程式架構大小的指示，請參閱第23頁的『安裝驗證』的步驟2。

日誌檔大小

如果您無法增加日誌檔大小，則會收到下列錯誤訊息：

```
啓用資料庫 logst...
```

```
從 EN 傳回
```

```
ABLE DB:
```

```
回覆碼 = -8
```

```
傳回訊息文字 =
```

```
GSE0008E 發生非預期的 SQL 錯誤 (“SQL3306N 插入橫列時，  
發生 SQL 錯誤 “-964””)。
```

如需如何增加日誌檔大小的指示，請參閱第23頁的『安裝驗證』中的步驟4。

後置安裝注意事項

安裝 Spatial Extender 之後，請考慮下列事項：

- 下載 ArcExplorer Java 版本 3.0
- 使用「DB2 Spatial Extender 地理編碼程式參照資料」及「資料與地圖」的 CD-ROM

下載 ArcExplorer

Environmental Systems Research Institute (ESRI) 提供可以產生 DB2 Spatial Extender 查詢資料視覺化結果的瀏覽器。此瀏覽器就是 ArcExplorer Java 版本 3.0。您可以從 ESRI 網站 (<http://www.esri.com>) 下載 ArcExplorer Java 版本 3.0。ArcExplorer 需要標準版或企業版的 Java® 2 Runtime Environment (JRE) 版本 1.2.2。

如需安裝及使用 ArcExplorer Java 版本 3.0 的其餘相關資訊，請參閱 *Using ArcExplorer* 一書，您也可以在 ESRI 網站中取得本書。

重要事項: DB2 Universal Database 版本 7.1 是與 IBM Java Development Kit (JDK) 版本 1.1.8 一起出貨。當您安裝適用於 ArcExplorer 的 JRE 版本 1.2.2 時，請將它安裝到與 DB2 不同的目錄。請記得要設定 CLASSPATH 環境變數。

使用「DB2 Spatial Extender 地理編碼程式參照資料」及「資料與地圖」的 CD-ROM

DB2 Spatial Extender 是與五個資料與地圖 CD-ROM 及一個地理編碼程式資料 CD-ROM 一起出貨的。

DB2 Spatial Extender 資料與地圖

在標籤為「DB2 Spatial Extender 資料與地圖 1 - 5」的五個 CD-ROM 中附有資料與地圖資訊。第27頁的表2提供每一個 CD-ROM 中的資料摘要。

表 2. 資料 & 地圖 CD-ROM 資訊

資料 & 地圖 CD-ROM	地圖資料摘要類型
CD-ROM 1	加拿大、歐洲、墨西哥、美國及全球
CD-ROM 2	美國（詳細）
CD-ROM 3	美國（西部地區）
CD-ROM 4	美國（東部地區）
CD-ROM 5	美國（南部地區）及範例影像資料

如需 ESRI 所提供的資料之詳細說明，請參閱「DB2 Spatial Extender 資料與地圖」CD-ROM 中的 ESRI 解說檔 `esridata.hlp`。

- 關於 Windows NT 及 Windows2000，請檢視 `x: esridata.hlp` 中的解說檔，其中 `x:` 是光碟機。
- 關於 AIX，請檢視或列印 CD-ROM 的 `/cdrom/esridata.hlp` 中的解說檔，其中 `/cdrom` 是您的裝載點。

DB2 Spatial Extender 地理編碼程式參照資料

「DB2 Spatial Extender 地理編碼程式參照資料」CD-ROM 中的地理編碼程式參照資料是特別建立的，以使用 DB2 Spatial Extender 預設地理編碼程式。它是由美國基本地圖街道網路資料組成，預設地理編碼程式用來確定在空間啓用資料庫中的位址緯度與經度。此基本地圖資料通稱為「參照資料」。預設地理編碼程式會採用資料庫中的位址資料（非空間性的），將它與參照資料比較及配對，然後轉換成 DB2 Spatial Extender 可以儲存的座標。此處理程序稱為地理編碼。

如需地理編碼程式的其餘相關資訊，請參閱第45頁的『使用地理編碼程式』。

存取地理編碼程式參照資料: 您可以從 CD-ROM 中直接存取地理編碼程式資料，或您可以將資料複製到硬碟中。若要從 CD-ROM 將地理編碼程式資料檔複製到 DB2 Spatial Extender 伺服器環境，請執行本節中所說明的步驟。

若是 AIX：

1. 裝載 CD-ROM。如需裝載 CD-ROM 的相關指示，請參閱第19頁的『裝載 CD-ROM』。
2. 以具有 Root 權限的使用者身份登入目標伺服器機器。
3. 鍵入下列：

```
cp /cdrom/db2/* /usr/lpp/db2_07_01/gse/refdata/
```
4. 登出。

若是 Windows NT 及 Windows 2000，您可以使用「指令視窗」或「Windows 檔案總管」。

若要使用「指令視窗」來存取地理編碼程式資料：

1. 按一下開始 -> 程式集 -> IBM DB2 -> 指令視窗。
2. 鍵入下列：

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

以對應於光碟機的字母取代 *d:*。

若要使用「Windows 檔案總管」來存取地理編碼程式資料：

從 *d:\db2* 將所有檔案複製到 *c:\sqllib\gse\refdata*，其中 *d:* 是光碟機且 *c:\sqllib* 是安裝 DB2 的目錄。

將 EDGELocator.loc 檔案提供給預設的地理編碼程式： CD-ROM 中提供的參照資料包括 EDGELocator.loc 檔案。EDGELocator.loc 檔案是預設地理編碼程式用來尋找特定的參照資料。例如，如果您要在加利福尼亞、肯塔基及奧勒岡中執行位址的地理編碼，則預設地理編碼程式會使用 CD-ROM 中的定位器檔案來確定位址位置。

在以預設的地理編碼程式啓用增量地理編碼（亦稱爲自動地理編碼），或在批次模式中執行預設地理編碼程式時，您必須使用 **vendorSpecific** 輸入參數。當您指定 **vendorSpecific** 輸入參數時，定位器檔案的目錄路徑與檔名必須傳送到地理編碼程式。

例如，下面的範例指令 **gseadm** 是用來呼叫使用預設地理編碼程式在批次模式中撰寫的地理編碼：

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

如需執行地理編碼程式及使用 **vendorSpecific** 參數的其餘相關資訊，請參閱第39頁的『第4章 定義一些空間直欄、將它們登記爲「層」，以及啓用地理編碼程式來維護它們』及第45頁的『第5章 在空間直欄大量輸入資料』。

呼叫 Spatial Extender

安裝 Spatial Extender 之後，您可以使用「DB2 控制中心」來設定 GIS 環境，然後開始使用空間資訊。

若要從「DB2 控制中心」呼叫 Spatial Extender：

1. 在「控制中心」視窗中，按一下您要執行 Spatial Extender 的伺服器。

2. 按一下 **Databases** 資料夾。即會在內容窗格中顯示資料庫。
3. 在您要使用的資料庫上按一下右鍵，然後在蹦現功能表中按一下您要執行的空間作業。

第3章 設置資源

安裝 Spatial Extender 之後，您可以對資料庫提供在建立空間直欄和操作空間資料時需要的資源。本章彙總這些資源以及說明使這些資源成爲可用資源的兩項作業：爲空間作業啓用資料庫以及建立空間參照系統。

資源庫存

您用來建立空間直欄和操作空間資料的資源包括：

- 參照資料：Spatial Extender 檢查的地址，目的是爲了驗證您要地理編碼的地址
- 讓資料庫能執行空間作業的資源：儲存程序、空間函數以及其它資源
- 使用者和供應商提供的非預設地理編碼程式。
- 空間參照系統

本節討論讓資料庫能執行空間作業的資料和資源。關於非預設地理編碼程式的資訊，請參閱第45頁的『關於地理編碼』。關於空間參照系統的資訊，請參閱第33頁的『關於座標和空間參照系統』。

參考資料

參考資料由 United States Census Bureau 收集美國境內的一些最新地址所組成。預設地理編碼程式將資料庫中的一個地址轉換成座標之前，它必須使該地址的部份或全部與參考資料中的一個地址相符。

安裝 Spatial Extender 時可使用參考資料。關於本資料需要的磁碟空間數量，請參閱第17頁的『磁碟空間需求』。若要在 AIX 上確認已適當載入此資料，請在 \$DB2INSTANCE/sqllib/gse/refdata/ 目錄尋找它。若要在 Windows NT 上確認已適當載入此資料，請在 %DB2PATH%\gse\refdata\ 目錄尋找它。

讓資料庫能執行空間作業的資源

安裝 Spatial Extender 之後執行的第一項工作是爲空間作業啓用資料庫。這包括起始一個動作，讓 Spatial Extender 能夠使用下列資源載入資料庫：

- 儲存程序。從「控制中心」要求一個動作時，Spatial Extender 會呼叫這些儲存程序的其中一個來執行該動作。
- 空間資料類型。您必須將一個空間資料類型指定給要儲存空間資料的表格或概略表直欄。關於詳細資訊，請參閱第39頁的『關於空間資料類型』。

- **Spatial Extender** 的型錄表格和概略表。某些作業根據 **Spatial Extender** 型錄而定。例如，具有空間資料類型的直欄必須以層次的身份登記在型錄之後，才能移入該直欄。關於層次的資訊，請參閱第9頁的『開發與實施 GIS 專案』。
- 一個空間索引類型。它可讓您定義層次的索引。
- 空間函數。您使用這些函數以數種方法來運用空間資料；例如，決定地理特徵的關係以及產生其它空間資料。這些函數的其中一個函數是預設的地理編碼程式。它將美國境內的地址轉換成座標，然後將這些座標插入空間直欄。關於空間函數的其它資訊，請參閱第131頁的『第13章 幾何與相關的空間函數』和第165頁的『第14章 SQL 查詢的空間函數』。關於預設地理編碼程式的其它資訊，請參閱第45頁的『關於地理編碼』。
- 一個名稱是 **DB2GSE** 的綱目，它含有剛才列示的物件。

關於如何使用「控制中心」起始載入這些資源的說明，請參閱『為空間作業啓用資料庫』。關於在應用程式中使用常式執行相同作業的準則，請參閱第61頁的『第8章 撰寫 **Spatial Extender** 的應用程式』。

為空間作業啓用資料庫

若要瞭解為空間作業啓用資料庫需要什麼樣的授權，請參閱第78頁的『授權』。

從「控制中心」為空間作業啓用資料庫：

1. 在「控制中心」視窗上，展開物件樹直到在您要執行 **Spatial Extender** 的伺服器下面找到 **Databases** 資料夾為止。
2. 按一下 **Databases** 資料夾。資料庫顯示在視窗的右邊內容窗格中。
3. 在您要的资料庫按一下右鍵，然後在蹦現功能表按一下 **Spatial Extender --> 啓用**。 **Spatial Extender** 提供資料庫一些資源，讓您建立和使用空間直欄和資料。

提示: **Spatial Extender** 必須安裝在某資料庫常駐的伺服器之後，您才能為空間作業啓用該資料庫。

建立空間參照系統

本節說明空間參照系統與座標系統的關係，以及解釋如何從「控制中心」建立空間參照系統。

關於座標和空間參照系統

本節繼續討論從第5頁的『空間資料的性質』開始討論的座標系統。然後根據第9頁的『開發與實施 GIS 專案』提供的空間參照系統定義展開此討論。它也提供一些準則讓您決定什麼值要指定給空間參照系統的參數。

座標系統、座標和測量

您可以將座標系統視為一個涵蓋某特定地理區域的想像格線。範例包括：涵蓋地球的格線、涵蓋一個國家的格線或涵蓋一個州內的某個區域的格線。此區域中的每一個地理特徵位於東西向格線和南北向格線的交點。一個名稱爲 *X* 座標的值指出此位置落在東西向格線的位置。另一個名稱爲 *Y* 座標的值指出此位置落於南北向格線的位置。這兩個值參照格線中心的位置 (又稱爲原點)。

位於原點的 *X* 和 *Y* 座標是 0。從原點向東，*X* 座標是正座標；從原點向西，它們是負座標。同樣地，從原點向北，*Y* 座標是正座標；從原點向南，它們是負座標。爲解釋這個分佈，試想下面的一般範例：座標系統 A 包括涵蓋大都會區的格線。*X* 座標 7 表示一個從本格線原點向東 7 個計量單位的位置。*X* 座標 -9.5 表示一個從原點向西 9.5 個計量單位的位置。

一個空間直欄內的每一個資料項目包括 (1) 一個 *X* 座標和一個 *Y* 座標，它們定義地理特徵的位置，或 (2) 多個 *X* 和 *Y* 座標，它們定義一個特徵的一部份的位置或定義一個特徵涵蓋的區域。也可以包括其它兩種值--一個 *Z* 座標和一個測量--。不像 *X* 和 *Y* 座標，在 Spatial Extender 不使用 *Z* 座標和測量單位定義位置或區域。它們只傳送 GIS 應用程式需要的資訊。*Z* 座標通常指出地理特徵的高度或深度。在原點上面的 *Z* 座標是正座標；在原點下面的 *Z* 座標是負座標。測量單位是數字；它可以傳送任何類型資訊。例如，假設您在 GIS 呈現油井。若需要應用程式處理一些表示地震資料的命中點 ID，那麼您可以儲存這些值作爲測量單位。

空間參照系統、偏移和比例係數

如『座標系統、座標和測量』所指出的情形，座標可以是以小數表示的負座標。對於測量單位來說也是這種情形。不過，爲了減少儲存體負荷，Spatial Extender 將座標和測量單位儲存成非負數整數 (亦即，儲存成正整數或 0)。因此，實際負小數座標和測量單位必須轉換成非負數整數，Spatial Extender 才能儲存它們。而且您需要告訴 Spatial Extender 如何轉換。您可以設定一些參數執行轉換。用來轉換一個地理區域內的座標和測量單位的參數設定值通稱爲空間參照系統。

您可透過下列動作建立空間參照系統：

- 首先，決定描述之特徵的最低負座標和測量單位。(距離 0 愈遠的負值就愈低。*X* 座標 -10 低於 *X* 座標 -5；測量單位 -100 低於測量單位 -50。)

- 然後，指定**偏移因數** (或簡稱**偏移**)：從負座標和測量單位扣除時產生非負數的值。
- 再指定**比例係數**：乘以小數座標和測量單位時所產生的整數值，這些整數的精準度至少與座標或測量單位相同。例如，試想一個具有精準度 4 的座標：92.77。您可以將它乘以比例係數 100 以獲得具有精準度 4 的整數：9277。請注意：在建立空間參照系統時，偏移因數會比比比例係數先套用。

決定最低負座標和測量單位

如果座標及測量是正整數，則 DB2 Spatial Extender 會儲存它們，但如果是負數或小數，則不會。因此，有必要將負座標及測量轉換為正座標及測量，將小數座標及測量轉換成整數。若要讓此轉換生效，您要定義一組參數，以在套用負數或小數座標測量時，即產生正整數。此組參數稱為**空間參照系統**。用來轉換負值的參數稱**偏移因數**，那些用來轉換小數值的參數稱為**比例係數**。

當您呼叫以小數座標或測量輸入的空間函數及空間參照系統的 ID 時，函數會將小數座標或測量乘以系統中的比例係數。結果就是 DB2 Spatial Extender 儲存的整數。比例係數必須夠大，以確保此整數的精準度與小數座標的精準度相同。

例如，假設 ST_Point 函數的輸入是由 X 座標 10.01、Y 座標 20.03 及空間參照系統的 ID 所組成。在呼叫 ST_Point 時，它會將值 10.01 及值 20.03 乘以 X 及 Y 座標的空間參照系統比例係數。如果此比例係數是 10，則 DB2 Spatial Extender 儲存的結果整數就會分別是 100 及 200。因為這些整數的精準度 (3) 小於座標的精準度 (4)，DB2 Spatial Extender 無法將這些整數轉換回原始座標，或從其中導出與這些座標所屬的座標系統一致的值。但如果比例係數是 100，則 DB2 Spatial Extender 儲存的結果整數就會是 1001 及 2003-- 該值可以轉換回原始座標或從其中導出相容的座標。

設定空間參照系統的參數之前，您必須決定一個地理區域中的最低負 X 座標、Y 座標、Z 座標和測量單位，該區域含有您想要瞭解其資訊的特徵。您可以透過回答下列問題瞭解這些值：

- 就您描述的特徵而言，有沒有特徵位於使用的座標系統的原點西邊？若有的話，什麼 X 座標指出最西邊特徵的位置或西邊？(解答會是正在處理的負 X 座標的最低座標。) 例如，若描述油井，而且部份油井位於原點西邊，那麼什麼 X 座標指出最西邊的油井位置？
- 有任何特徵在原點南邊嗎？若有的話，什麼 Y 座標指出最南邊特徵的位置或南邊？(解答會是正在處理的負 Y 座標的最低座標。) 例如，若描述油井，而且部份油井位於原點南邊，什麼 Y 座標指出最南邊的油井位置？
- 若要使用 Z 座標定義深度，哪一個特徵是最深的，以及那一個 Z 座標代表這個特徵的最低點？(解答會是正在處理的負 Z 座標的最低座標。)

- 若想要將測量單位包括在空間資料內，會有負的測量單位嗎？若有的話，什麼是負的測量單位的最低測量單位？

確定最低負座標和測量單位之後，對每一個座標新增它的值的百分之 5 到 10。例如，若最低負座標 X 座標是 -100，那麼您可以對它新增 -5。本書將結果數字稱為增值。

註：DB2 Spatial Extender 預設空間參照系統的 ID 是 0（零）。DB2 Spatial Extender 提供與預設地理編碼程式一起使用的空間參照系統。此系統的 ID 是 1。

指定偏移因數

下一步，指定 Spatial Extender 應使用什麼偏移因數將負座標和測量單位轉換成非負的座標和測量單位：

- 決定您要增值的 X 值成為什麼值之後，請指定從這個值扣除之後剩下 0 的偏移。然後 Spatial Extender 會從全部負 X 座標扣除這個數字以到達一個正值。Spatial Extender 也會從其它 X 座標扣除這個數字。
例如，若增值的 X 值是 -105，那麼您需要從這個值扣除 -105 以得到 0。然後 Spatial Extender 會從與您描述的特徵相關的全部 X 座標扣除 -105。因為這些座標沒有大於 -100，所以產生自此扣除的值會是正值。
- 同樣地，指定從增值的 Y 值、增值的 Z 值和增值的測量單位扣除之後剩下 0 的偏移。

從 X 座標扣除的偏移稱為假 X。從 Y 座標、Z 座標和測量單位扣除的偏移分別稱為假 Y、假 Z 和假 M。關於從「控制中心」指定這些參數的說明，請參閱第 36 頁的『從「控制中心」建立空間參照系統』。

指定比例係數

下一步，指定 Spatial Extender 應使用什麼比例係數將小數座標和測量單位轉換成整數：

- 乘以一個小數 X 座標或一個小數 Y 座標時，指定一個產生 32 位元整數的比例係數。建議您使這個比例係數成為係數 10：10 的一次方 (10)，係數 10 的二次方 (100)，係數 10 的三次方 (1000) 或更大係數 (如果需要)。決定比例係數應該是係數 10 的什麼：
 1. 決定哪些 X 和 Y 座標是或可能是小數。例如，假設您要處理一些不同的 X 和 Y 座標，您測定其中三個座標是小數：1.23、5.1235 和 6.789。
 2. 選取具有最長小數位數的小數座標。然後決定這個座標可以乘以什麼係數 10 以產生相等精準度的整數。說明如右：就現行範例中的三個小數座標而言，5.1235 有最長小數位數。它乘以 10 的 4 次方 (10000) 會產生整數 51235。

3. 判斷上述乘法產生的整數是否太長而無法儲存成 32 位元資料項目。51235 不會太長。但假設除了 1.23、5.11235 和 6.789 以外，您的 X 和 Y 座標範圍包括第四個小數值 10006.789876。因為這個座標的小數位數比其它三個座標長，所以您要將這個座標--不是 5.1235--乘以係數 10。若要將它轉換成爲整數，您可以將它乘以 10 的 6 次方 (1000000)。但是結果值 10006789876 太長而無法儲存成 32 位元資料項目。若 Spatial Extender 嘗試儲存它，那麼將出現無法預期的結果。

爲了避免這個問題，請在乘以原始座標時選取一個產生一個小數的係數 10，Spatial Extender 可截斷這個小數成爲可儲存的整數 (失去最少精準度的整數)。在這種情況下，您可選取 10 的 4 次方 (10000)。將 10000 乘以 10006.789876 產生 100067898.76。Spatial Extender 會截斷這個數字成爲 100067898，稍微減少它的精確度。

- 若描述的特徵有小數 Z 座標，請遵循前述程序確定這些座標的比例係數。若這些特徵與小數測量單位相關，請遵循這個相同程序以確定這些測量單位的比例係數。

X 和 Y 座標的比例係數稱爲 XY 單位。Z 座標和測量單位的比例係數分別稱爲 Z 單位和 M 單位。關於從「控制中心」指定這些參數的說明，請參閱『從「控制中心」建立空間參照系統』。

從「控制中心」建立空間參照系統

本節概述從「控制中心」建立空間參照系統的一些步驟。此概觀之後詳細說明如何完成每一個步驟。

執行這些步驟不需要授權。

從「控制中心」建立空間參照系統的步驟概觀：

1. 開啓「建立空間參照系統」視窗。
2. 指出要使用哪一個座標系統。
3. 指定要建立的空間參照系統的識別字。
4. 決定何種範圍的座標和測量單位適用於您想要瞭解的地理特徵。
5. 指定一些值，這些值可用來將負座標或小數座標和測量單位轉換成 Spatial Extender 可儲存的資料項目。
6. 告訴 Spatial Extender 建立您要的空間參照系統。

從「控制中心」建立空間參照系統的詳細步驟：

1. 開啓「建立空間參照系統」視窗。

- a. 在「控制中心」視窗上，展開物件樹直到在您要執行 Spatial Extender 的伺服器下面找到 **Databases** 資料夾為止。
 - b. 按一下 **Databases** 資料夾。資料庫顯示在視窗的右邊內容窗格中。
 - c. 在您為空間資料啓用的資料庫按一下右鍵，然後在蹦現功能表按一下 **Spatial Extender --> 空間參照**。「空間參照」視窗開啓。
 - d. 在「空間參照」視窗上，按一下**建立**。「建立空間參照」系統視窗開啓。
2. 在「建立空間參照」視窗上，使用**座標系統**欄位指出要使用的座標系統。
 3. 指定要建立的空間參照系統的識別字。

- 在**名稱**欄位中，鍵入系統的 1 到 64 個字元名稱。

限制: 請勿指定另一個空間參照系統名稱。資料庫中不可以有兩個空間參照系統同名。

- 在 **ID** 欄位中，鍵入一個數值識別字。它必須是整數。

限制: 請勿指定另一個空間參照系統的 **ID**。資料庫中不可以有兩個空間參照系統有相同 **ID**。

4. 使用「控制中心」以外的媒體--例如，紙張或白板--決定適用於您要描述的地理特徵的最低負座標和測量單位。關於如何應用的準則，請參閱第34頁的『決定最低負座標和測量單位』。
5. 在「建立空間參照」視窗上，指定一些值，這些值將負座標或小數座標和測量單位轉換成 Spatial Extender 支援的資料項目-- 亦即，轉換成 32 位元非負數整數。
 - a. 指定一些值將負的或小數 **X** 座標轉換成非負數整數：
 - 在**偏移**直欄中，在最接近 **X** 的欄位指定假 **X**：
 - 若您在步驟4中識別的 **X** 座標範圍內的任何值是負值，請鍵入一個假 **X**，從最低負座標扣除它之後剩下一個正數。關於準則，請參閱第35頁的『指定偏移因數』。
 - 若全部 **X** 座標是非負座標，請在假 **X** 鍵入 0。
 - 在**比例係數**欄位，在 **X** 最右邊的欄位指定一個 **XY** 單位。乘以任何小數 **X** 座標或小數 **Y** 座標時，這個 **XY** 單位必須是一個產生整數的單位，該整數可儲存成損失最少精準度的 32 位元資料項目。關於準則，請參閱第35頁的『指定比例係數』。

在 **X** 最右邊的欄位指定 **XY** 單位之後，它也會出現在 **Y** 最右邊的欄位。
 - b. 指定一個假 **Y**，它允許 Spatial Extender 將負 **Y** 座標轉換成正值。您在**偏移**直欄中，在最接近 **Y** 的欄位指定假 **Y**：

- 若您在步驟第37頁的4中識別的 Y 座標範圍內的任何值是負值，請指定一個假 Y，從最低負座標扣除它之後剩下一個正數。關於準則，請參閱第35頁的『指定偏移因數』。
 - 若全部 Y 座標是正座標，請在假 Y 鍵入 0。
- c. 若想要將 Z 座標包括在空間資料內，請指定一些值將負的或小數 Z 座標轉換成非負數整數：
- 在**偏移**直欄中，在最接近 Z 的欄位，鍵入一個假 Z：
 - 若您在步驟第37頁的4中識別的 Z 座標範圍內的任何值是負值，請鍵入一個假 Z，從最低負座標扣除它之後剩下一個正數。關於準則，請參閱第35頁的『指定偏移因數』。
 - 若全部 Z 座標是非負座標，請在假 Z 鍵入 0。
 - 在**比例係數**直欄中，在 Z 最右邊的欄位鍵入一個 Z 單位。乘以任何小數 Z 座標時，這個 Z 單位必須是一個產生整數的單位，該整數可以儲存成損失最少精準度的 32 位元資料項目。關於準則，請參閱第35頁的『指定比例係數』。
- d. 若要將測量單位包括在空間資料內，請指定一些值將負的或小數測量單位轉換成正整數：
- 在**偏移**直欄中，在最接近**線性**標籤的欄位中鍵入一個假 M：
 - 若您在步驟第37頁的4中識別的測量單位範圍內的任何值是負值，請鍵入一個假 M，從最低負測量單位扣除它之後剩下一個正數。關於準則，請參閱第35頁的『指定偏移因數』。
 - 若全部測量單位是正單位，請在假 M 鍵入 0。
 - 在**比例係數**直欄中，在**線性**標籤最右邊的欄位指定一個 M 單位。乘以任何小數測量單位時，這個 M 單位必須是一個產生整數的單位，該整數可以儲存成損失最少精準度的 32 位元資料項目。關於準則，請參閱第35頁的『指定比例係數』。
6. 按一下**確定**建立想要的空間參照系統。

第4章 定義一些空間直欄、將它們登記為「層」，以及啓用地理編碼程式來維護它們

在對 Spatial Extender GIS 設置一些資源之後，您就可以開始建立含有空間資料的物件。例如，若您需要一些新表格來包含空間資料，可以先定義它們，然後對您要加入資料的直欄指定空間資料類型。若要新增空間直欄到現存的表格中，也可以這麼做。

當提供含有某空間直欄的新表格或現存表格時，您必須將此直欄登記為「層」。此外，若您打算使用地理編碼程式來大量輸入資料到直欄中，您可以在登記直欄為「層」時啓用地理編碼程式來自動維護該直欄。按下列方式啓用地理編碼程式：Spatial Extender 會定義已編碼的觸發函式，每當空間直欄的對應屬性直欄接收新資料或更新的資料時，觸發函式即呼叫地理編碼程式。呼叫後，地理編碼程式會將新資料或更新的資料轉換成空間資料，然後將此資料放置到空間直欄中。

為表格定義空間直欄之後，需要的話，您可以透過這個表格直欄來建立一個概略表直欄。在將表格直欄登記為「層」之後，也要將概略表直欄登記為「層」。

本章討論您可以指定給空間直欄的資料類型之用法及本質。其次，本章說明如何使用「控制中心」來為表格定義空間直欄、將空間直欄登記為「層」，以及啓用地理編碼程式來維護它。最後，本章說明如何使用「控制中心」將概略表直欄登記為「層」。

關於空間資料類型

本節介紹空間直欄必要的資料類型，並提供選擇空間直欄資料類型的準則。

當您對空間作業啓用某資料庫時，Spatial Extender 會提供資料庫一個結構式資料類型的階層。第40頁的圖6顯示此階層。在本圖中，可引證的類型為白色背景；不可引證的類型為灰色背景。

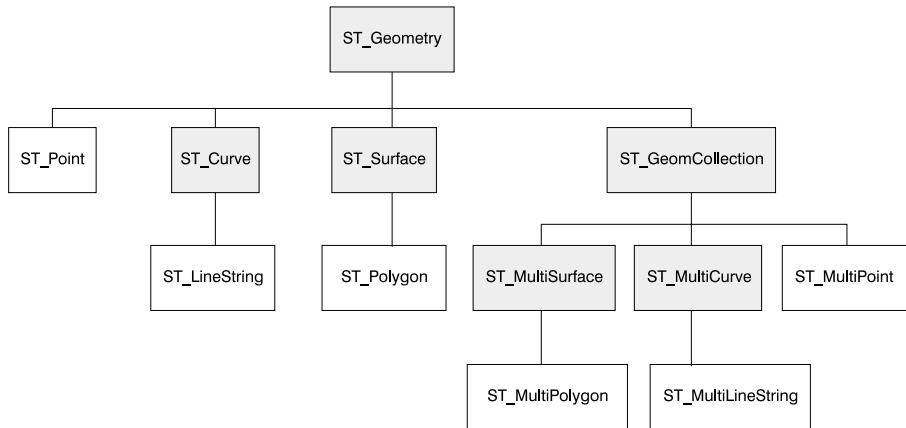


圖 6. 空間資料類型的階層。以白色框指定的資料類型為可引證的。以灰暗框指定的資料類型為不可引證的。

圖6中的階層包括：

- 可視為形成單一單元的地理特徵之資料類型；例如，個別的住宅及隔離的湖泊。
- 由多重單元或元件組成的地理特徵之資料類型；例如，公路系統及山系。
- 所有種類的地理特徵之資料類型。

單一單元特徵的資料類型

使用 `ST_Point`、`ST_LineString` 及 `ST_Polygon` 來儲存一些座標，這些座標定義可視為形成單一單元的特徵所佔用的空間：

- 當您要指出被個別地理特徵所佔用空間中的點時，請使用 `ST_Point`。此特徵可能很小，例如一口水井；可能很大，例如一個城市；或不大不小，例如一整棟建築物或公園。上述情況中，空間中的點可位於東西座標線（例如平行線）與南北座標線（例如子午線）的交集點上。`ST_Point` 資料項目包括定義這類交集的 X 座標值與 Y 座標值。X 座標指示位於東西線上的交集點；Y 座標指示位於南北線上的交集點。
- 對於定義線性特徵（例如街道、溝渠及輸油管）所佔用的空間之座標，請使用 `ST_LineString`。
- 當您要指示多邊形特徵（例如福利區、森林或野生動植物產地）涵蓋的空間延伸範圍時，請使用 `ST_Polygon`。`ST_Polygon` 資料項目由定義此類特徵周長的座標組成。

在某些情況下，`ST_Polygon` 和 `ST_Point` 可用於相同的特徵。例如，假設您需要有關好幾棟公寓的空間資訊。若要顯示空間中每一棟所在的點，您要使用

ST_Point 來儲存定義每一個這種點的 X 及 Y 座標。另一方面，若要顯示每一棟涵蓋的區域，您要使用 ST_Polygon 來儲存定義此類區域周長的座標。

多單元特徵的資料類型

使用 ST_MultiPoint、ST_MultiLineString 及 ST_MultiPolygon 來儲存一些座標，這些座標定義多單元組成的特徵所佔用的空間：

- 當您要顯示由個別單元組成的特徵，而且要指出每一個元件所佔用空間中的點，請使用 ST_MultiPoint。ST_MultiPoint 資料項目包括定義此類特徵的各元件位置之 X 與 Y 座標對。例如，考慮表格中哪些橫列要代表群島，哪些直欄包括 ST_MultiPoint 直欄。此直欄中每一個資料項目包括定義每一群島中獨立島嶼位置的 X 與 Y 座標對。
- 當您要顯示由長度單位組成的特徵，而且要每一個單元所佔用空間的相關資訊，請使用 ST_MultiLineString。ST_MultiLineString 資料項目由定義此類空間的座標組成。例如，考慮表格中哪些橫列要代表河流系統，哪些直欄包括 ST_MultiLineString 直欄。此直欄中每一個資料項目包括定義各系統中的河流路線的座標組。
- 當您要顯示由多邊形單元組成的特徵，而且要每一個單元所佔用空間的相關資訊時，請使用 ST_MultiPolygon。例如，考慮表格中哪些橫列要代表美國中西部各郡，哪些直欄包括 ST_MultiPolygon 直欄。此直欄包含農地的相關資訊。尤其，此直欄中每一個資料項目包括定義某一郡的農地周長的座標組。

全部特徵的資料類型

若不確定要使用其它哪些資料類型，您可使用 ST_Geometry。由於 ST_Geometry 為其它資料類型所屬階層的 Root，所以 ST_Geometry 直欄可儲存其它資料類型所指定儲存於直欄中的任何或全部值。

請注意：如果您打算使用預設地理編碼程式來大量輸入空間直欄資料，則直欄必須是 ST_Point 或 ST_Geometry 類型。

為表格定義空間直欄、將此直欄登記為「層」，以及啓用地理編碼程式來維護它

本節提供如何為表格定義空間直欄、將此直欄登記為「層」，以及啓用地理編碼程式來維護它的逐步概觀。此概觀之後詳細說明如何完成每一個步驟。

若要知道需要哪一種授權才能將表格直欄登記為「層」，請參閱第92頁的『授權』。若要知道需要哪一種授權才能啓用地理編碼程式來維護此直欄，請參閱第75頁的『授權』。

為表格定義空間直欄、將此直欄登記為「層」，以及啓用地理編碼程式來維護它的步驟概觀：

1. 若此空間直欄要成為新表格的一部份，請建立這個表格。
2. 開啓「建立空間層」視窗。
3. 將空間直欄加入表格中，指示您要將此直欄登記為「層」；或指示您要將現存直欄登記為「層」。
4. 指示哪一個空間參照系統要用於該「層」。
5. 若此「層」要包含匯入的資料，或另一個空間直欄產生的資料，請通知 Spatial Extender 建立該「層」。
6. 如果此「層」要包含衍生自屬性資料的資料：
 - a. 請指定哪一個直欄包含此屬性資料。
 - b. 指示您要啓用地理編碼程式來維護該「層」。
 - c. 通知 Spatial Extender 建立該「層」。

為表格定義空間直欄、將此直欄登記為「層」，以及啓用地理編碼程式來維護它的詳細步驟：

1. 若此空間直欄要成為新表格的一部份，請建立這個表格：
 - 使用您選擇的介面 (例如，「控制中心」或「指令行處理器」) 來建立此表格。
 - 若您打算使用地理編碼程式，請加入一至十個直欄供地理編碼程式操作。地理編碼程式無法將十個以上的資料直欄當作輸入直欄。
 - 請加入您要登記為「層」的空間直欄，或在步驟3中定義此直欄。

若要使用現存的表格，請跳至下一個步驟。

2. 開啓「建立空間層」視窗。
 - a. 從「控制中心」視窗中，展開物件樹狀結構，直到您找到 **Tables** 資料夾，它代表您要用於空間作業的資料庫中表格。
 - b. 按一下 **Tables** 資料夾。表格顯示在視窗的右邊內容窗格中。
 - c. 在您要的表格上按一下右鍵，然後在蹦現功能表中按一下 **Spatial Extender** --> **空間層**。出現「空間層」視窗。
 - d. 從「空間層」視窗中，按一下**建立**。即開啓「建立空間層」視窗。
3. 從「建立空間層」視窗中，將空間直欄加入表格中，指示您要將此直欄登記為「層」；或指示您要將現存直欄登記為「層」。
 - 若要將空間直欄加入表格中並定義此直欄為「層」：
 - a. 在層直欄欄位中，鍵入直欄名稱。

- b. 在**直欄類型**欄位中，選取或鍵入您要直欄使用的資料類型。關於可容許的資料類型的討論，請參閱第39頁的『關於空間資料類型』。
- 若要將現存直欄定義為「層」，請在**層直欄**欄位中選取它。

限制：請勿選取已定義為「層」的直欄。

4. 在**空間參照名稱**欄位中，指定要用於該「層」的空間參照系統名稱。
5. 若要該「層」包含匯入的資料，或另一個空間直欄產生的資料，請按一下**確定**來登記它。
6. 若要該「層」包含衍生自屬性資料的資料：
 - a. 請指定哪一個直欄要包含此屬性資料：
 - 1) 在**可用的直欄**方框中選取直欄。最多可以選取十個直欄。
 - 2) 按一下 **>** 按鈕、**>>** 按鈕或兩者，在**所選取直欄**方框中列示選取的直欄。
 - b. 若要啓用地理編碼程式來維護該「層」：
 - 1) 請選取**啓用自動地理編碼程式**勾選框。
 - 2) 在**名稱**欄位中，選取要使用的地理編碼程式名稱。
 - 3) 在**精準度層次**欄位中，以百分率指定輸入記錄必須符合參照資料中對應的記錄到何種程度才能被處理。此百分率稱為**精準度**。例如，假設地理編碼程式讀取含有地址 **557 Bailey, San Jose 94120** 的輸入記錄。若精準度為 **100**，但此地址與其參照資料中副本之間的相符程度不是完全正確，則地理編碼程式會拒絕它。若精準度為 **75**，但此記錄與其參照資料副本間的相符程度至少有 **75%** 正確，則地理編碼程式會處理它。
 - 4) 若地理編碼程式由供應商提供，請使用**內容**方框，指定您要使用的任何供應商提供的地理編碼參數。
 - c. 按一下**確定**，將選取的直欄登記為「層」，並在必要時，啓用地理編碼程式來維護該直欄。

限制

您無法變更加入到地理編碼程式的直欄。不過，可變更其它的地理編碼內容。例如，您可以啓用或停用遞增式地理編碼。

註：「刪除層」視窗不是用來刪除層，而是取消登記層；亦即從 **DB2 Spatial Extender** 系統型錄中除去其所有相關資訊。取消登記層時，該層所使用的表格或概略表直欄仍會繼續存在。

將概略表直欄登記為「層」

若要知道需要哪一種授權才能將概略表直欄登記為「層」，請參閱第92頁的『授權』。

將概略表直欄登記為「層」：

1. 開啓「建立空間層」視窗。
 - a. 從「控制中心」視窗中，展開物件樹狀結構，直到您找到 **Views** 資料夾，它代表您要用於空間作業的資料庫中概略表。
 - b. 按一下 **Views** 資料夾。視窗右邊的內容窗格中即顯示一些概略表。
 - c. 在您要的概略表上按一下右鍵，然後在蹦現功能表中按一下 **Spatial Extender --> 空間層**。出現「空間層」視窗。
 - d. 從「空間層」視窗中，按一下**建立**。即開啓「建立空間層」視窗。
2. 使用**層直欄**方框，指定您要登記為「層」的直欄。
3. 在**基礎空間層欄位**中，指定根據選取的概略表直欄所決定的表格直欄名稱。此表格直欄必須已登記為「層」。
4. 按一下**確定**，將指定的概略表直欄登記為「層」。

註：在捨棄具有已登記為「層」之直欄的表格之前，您必須先執行下列一或多個動作：

- 取消登記該層。
- 如果該層有空間索引，請刪除此索引。
- 如果透過表格直欄已定義概略表直欄，並且這個概略表直欄已登記為層時，也要取消登記這一層。

第5章 在空間直欄大量輸入資料

將空間直欄登記成爲層之後，您就可開始提供空間資料。如第6頁的『空間資料來自何處』所述，有三種方式可提供此資料：使用一個名爲地理編碼程式的函數，自屬性資料中衍生出來；使用其它函數自其它空間資料衍生出來；從檔案中匯入。本章：

- 談論地理編碼，並解釋如何使用「控制中心」以批次模式來地理編碼屬性資料
- 討論匯入和匯出資料，並解釋如何使用「控制中心」將資料匯入您的 GIS 以及自 GIS 匯出

關於可自現存的空間資料中衍生出新空間資料的函數，請參閱第155頁的『從現存的幾何產生新幾何的函數』。

使用地理編碼程式

本節說明地理編碼的處理，並解釋如何在「控制中心」內以批次模式來執行地理編碼程式。

關於地理編碼

如之前所述，DB2 Spatial Extender 會使用稱爲地理編碼程式的函數，來執行兩個活動：將位址資料轉換成空間資料，並將此空間資料放入表格直欄中。本節區分地理編碼程式及其來源之間的基本差異。亦說明地理編碼程式的兩種操作模式，並介紹您在使用地理編碼程式時需要考慮的因素。

透過 Spatial Extender，您可以：

- 使用 Spatial Extender 提供的預設地理編碼程式。
- 外掛協力廠商供應商開發的地理編碼程式。
- 外掛您自己的地理編碼程式。

預設地理編碼程式可地理編碼美國地址，並且轉換成 ST_Point 資料。若您需要儲存其它空間資料類型，您可外掛一個地理編碼程式來建立這種資料。若您需要的空間資料係指美國以外的地區，或沒有地址的位置-- 例如以土壤含量各異的農地-- 您也可以外掛一個符合此需求的地理編碼程式。

使用外掛地理編碼程式之前，必須先行登記。使用者和供應商可使用 db2gse.gse_register_gc 儲存程序來登記。無法從「控制中心」來登記。關於

db2gse.gse_register_gc 的資訊，請參閱第90頁的『db2gse.gse_register_gc』。關於使用 Spatial Extender 儲存程序的一般資訊，請參閱第69頁的『第9章 儲存程序』。

地理編碼程式有兩種操作模式：

- 在批次模式，嘗試一次將空間直欄中的全部現存來源資料轉換成空間資料，將此資料移入直欄內。您可從「執行地理編碼程式」視窗中起始該作業。另外，您也可在應用程式中起始該作業，只要撰寫程式碼來呼叫 db2gse.gse_run_gc 儲存程序即可。
- 在增量模式下，地理編碼程式會轉換表格中插入或更新的資料，將結果的空間值放入直欄中來更新直欄。您可自「建立空間層」視窗中要求插入和更新觸發函式來起動此作業。另外，您也可在應用程式中提出要求，只要撰寫程式碼來呼叫 db2gse.gse_enable_autogc 儲存程序即可。

增量地理編碼亦稱為自動地理編碼。

計畫使用地理編碼程式時，您需要考慮下列因素：

1. 使用「控制中心」時，您通常先使用「建立空間層」視窗，然後再使用「執行地理編碼程式」視窗。這表示在您起始批次地理編碼之前，您可以先讓 Spatial Extender 設定觸發函式來處理增量地理編碼。因此，增量地理編碼可能優先於批次地理編碼。以批次模式來處理全部的來源資料時，地理編碼程式會地理編碼增量模式中操作的相同資料。這個多餘的程序不會造成資料重複（空間資料第二次產生時，第二次的資料將覆蓋第一次的資料）。不過，效能會降低。避免這種情形的方法是延遲設置觸發函式，直到完成批次地理編碼之後再設置。
2. 若觸發函式在您準備以批次模式來地理編碼時已備妥，則建議您停用觸發函式，直到批次地理編碼完成為止。您可在「執行地理編碼程式」視窗，或藉由撰寫程式碼來呼叫 db2gse.gse_disable_autogc 儲存程序，在應用程式中停用觸發函式。若您使用「執行地理編碼程式」視窗，則在地理編碼完成時，Spatial Extender 將自動重新啟用觸發函式。若您使用 db2gse.gse_disable_autogc 儲存程序，您可呼叫 db2gse.gse_enable_autogc 儲存程序使觸發函式再生效。
3. 若您要以批次模式執行地理編碼程式來移入具有索引的空間直欄，請先停用或捨棄索引。否則，地理編碼程式執行時，索引將嚴重造成效能低落。若您使用「控制中心」，您可從「執行地理編碼程式」視窗中停用索引。當地理編碼完成時，Spatial Extender 會自動重新啟用索引。若您使用應用程式，您可使用 SQL DROP 陳述式來捨棄索引。但請記下索引的參數，以便於批次地理編碼完成之後重建索引。
4. 當地理編碼程式讀取一筆來源資料記錄時，將在參照資料內搜尋相符的副本。此相符過程必須有特定的正確程度（稱為精準度），以利地理編碼程式處理記錄。例如，精準度 85 表示一筆來源記錄和其參照資料的副本，至少必須具備 85% 的精確率，以利處理來源記錄。

您可指定精準度。但可能需要做些調整。例如，假設精準度是 100。若有許多來源記錄包含的地址比參照資料還要新，則這些記錄和參照資料之間將不可能達到完全相符。因此，地理編碼程式將拒絕這些記錄。大致上，若地理編碼程式產生的空間資料似乎不足或極不精確，您仍然可變更精準度，重新執行地理編碼程式來解決這個問題。

有數種方式可以控制在發出確定前地理編碼程式處理的記錄數或範圍：

方法 1

在每一個確定前，地理編碼程式可以在特定的記錄數中地理編碼位址資料。此方法可以讓您精確地管理工作單元的大小。但是，此方法比此處所說明的其它方法使用更大量的額外負荷。

若要起始方法 1，請指定每一個確定前要處理的記錄數。如果您使用的是「控制中心」，請使用「執行地理編碼程式」視窗中的**確定範圍**旋轉鈕來設定此數字。如果您是要撰寫應用程式，請對 `db2gse.gse_run_gc` 儲存程序的 `commitScope` 參數指定此數字。

方法 2

地理編碼程式可以在發出確定前，在表格的所有記錄中地理編碼位址資料。使用此方法，地理編碼程式可以用一種類似批次作業的方法處理記錄，每一筆記錄中所使用的額外負荷比方法 1 少許多。但是，您對工作單元大小的控制權比方法 1 小。因此，在地理編碼程式作業時，您無法控制要保留多少鎖定或要完成多少日誌登錄。尤其，如果地理編碼程式發生需要 `Rollback` 的錯誤，您就必須設定地理編碼程式以在所有記錄上重新執行。如果表格非常大，且錯誤及 `rollback` 是在已處理了大部份的記錄後才發生，則最後的資源成本會非常高。

若要從「控制中心」起始方法 2，請將「執行地理編碼程式」視窗中的**確定範圍**旋轉鈕設為零。如果您是要撰寫應用程式，請將 `db2gse.gse_run_gc` 儲存程序的 `commitScope` 參數設為零。

方法 3

在每一個確定前，地理編碼程式可以在表格記錄的子集中地理編碼位址資料。然後，它可以在第二個子集（如果必要，可以在第三個、第四個等等）中地理編碼位址資料。地理編碼程式會以一種類似批次作業的方法來處理每一個子集，每一個記錄所使用的額外的負荷比方法 1 少許多。但如方法 2 相同，方法 3 不容許您對工作單元大小有直接的控制權。尤其，它需要額外的負荷以多次設定及執行地理編碼程式 - 每一個記錄子集一次。

若要從「控制中心」定義記錄子集：

- 在「執行地理編碼程式」視窗中，使用 **Where** 子句方框來編寫 SELECT WHERE 子句的程式，指定主旨中的記錄範圍。
- 將「執行地理編碼程式」視窗中的**確定範圍**旋轉鈕設為零。

若要在應用程式中定義記錄子集，請編寫 db2gse.gse_run_gc 儲存程序，如下所示：

- 請使用 whereClause 參數來指定子集中的記錄範圍。
- 將 commitScope 參數設為零。

以批次模式執行地理編碼程式

本節概略說明從「控制中心」內以批次模式執行地理編碼程式的步驟。此概觀之後詳細說明如何完成每一個步驟。

關於以批次模式執行地理編碼程式所需的授權，請參閱第98頁的『授權』。

以批次模式執行地理編碼程式的步驟概觀

1. 開啓「執行地理編碼程式」視窗。
2. 指出您要使用的地理編碼程式。
3. 停用可能損害地理編碼程式效能的物件。
4. 指定 DB2 發出確認之前有多少記錄須地理編碼。
5. 指出地理編碼程式要如何操作。
6. 指示 Spatial Extender 執行地理編碼程式。

以批次模式執行地理編碼程式的詳細步驟

1. 開啓「執行地理編碼程式」視窗。
 - a. 在「控制中心」視窗中，展開物件樹狀結構，直到在您已啓用空間功能的資料庫中找到 **Tables** 資料夾為止。
 - b. 按一下 **Tables** 資料夾。表格顯示在視窗的右邊內容窗格中。
 - c. 在內容窗格中以滑鼠右鍵按一下您要的表格，並按一下蹦現功能表中的**空間層**。出現「空間層」視窗。
 - d. 從「空間層」視窗中：
 - 1) 在您要移入資料的直欄內，選取已定義的層。
 - 2) 按一下**執行地理編碼程式**按鈕，出現「執行地理編碼程式」視窗。
2. 若您要使用預設地理編碼程式，請勿變動**名稱**方框，其中顯示預設值的名稱。否則，請使用方框來選取您要的地理編碼程式。
3. 停用可能損害地理編碼程式效能的物件：

- 若您要輸入資料的直欄含有索引，請勾選**地理編碼處理期間暫停空間索引**勾選框。
 - 若觸發函式已設定為對此直欄啟動增量地理編碼，請勾選**地理編碼處理期間暫停空間觸發函式**勾選框。
當您按一下「執行地理編碼程式」視窗的**確定**時，索引和觸發函式將自動重新啟用。
4. 使用**確定範圍**旋轉鈕，指定 DB2 發出確認之前有多少記錄須地理編碼。例如，若您要讓 DB2 一次確認 100 筆已經地理編碼的記錄，請指定數字 100。
要訣：若您要讓 DB2 只在處理完全部記錄之後才發出確認，請指定 0。
 5. 使用**地理編碼程式參數**群組框的欄位，指出您要讓地理編碼程式如何操作：
 - 使用**精準度**旋轉鈕，以百分率指定來源記錄及其參照資料副本之間的相符程度。關於精準度的詳細資訊，請參閱第45頁的『關於地理編碼』。
 - 若您使用廠商提供的地理編碼程式，且想要利用支援的內容，請使用**內容**方框來設定這些內容。
 - 若您只要地理編碼表格中的一個列子集，請使用 **WHERE 子句**方框，編寫 **SELECT WHERE** 子句，為您所要的列指定準則。此子句可參照到表格中的任何直欄。
只鍵入準則。省略關鍵字 **WHERE**。例如，若表格有一個 **STATE** 直欄，且您只想地理編碼的列含有此直欄中的 **MA** 值，請鍵入：
`STATE='MA'`
 6. 按一下 **確定**來執行地理編碼程式。

匯入和匯出資料

本節說明匯入和匯出資料的程序，並解釋如何使用「控制中心」來執行下列動作：

- 從資料交換檔中匯入資料到新的或現存的表格
- 從資料交換檔中匯入資料到現存的表格
- 從表格中匯出資料到資料交換檔

關於匯入和匯出

本節列示匯入和匯出空間資料的原因。亦討論資料交換檔，這種檔案可做為匯出來源和匯入目標之間的介面。

您可使用 **Spatial Extender** 在資料交換檔中匯入和匯出空間資料。請考慮這些實務範例：

- 您的 GIS 包含的空間資料，代表著您的辦公室、客戶及其它商務要項。您想要以代表組織文化環境--城市、街道、重要定點等--的空間資料來補充此資料。您要的資料可向地圖供應商取得。您可使用 Spatial Extender，從供應商提供的資料交換檔中匯入您要的資料。
- 您要從 Oracle 系統中將空間資料移轉至您的 Spatial Extender GIS。您接著使用 Oracle 公用程式將資料載入資料交換檔中。然後，使用 Spatial Extender，將此檔案中的資料匯入您爲了執行空間作業而啓用的資料庫。
- 您要使用 GIS 瀏覽器來向客戶顯示空間資訊。瀏覽器只需要使用檔案，不需要連接資料庫。您可使用 Spatial Extender 將資料匯出到資料交換檔，然後使用瀏覽器公用程式將資料載入瀏覽器中。

「控制中心」支援 Spatial Extender 使用兩種資料交換檔：形狀檔和 ESRI_SDE 傳送檔。形狀檔通常用來匯入檔案系統所產生的資料，以及將資料匯出到要載入檔案系統中的檔案。ESRI_SDE 傳送檔通常用來匯入 ESRI 資料庫所產生的資料。

在資料庫層次將資料匯入新的或現存的表格

本節概述使用「控制中心」的 **Databases** 資料夾，從形狀或 ESRI_SDE 傳送檔將資料匯入新的或現存表格的步驟。此概觀之後詳細說明如何完成每一個步驟。

當您匯入一組 ESRI 形狀表示法時，您至少會收到兩個檔案。所有檔案的名稱都相同，但副檔名不同。例如，您一定會收到的兩個檔案的副檔名是 .shp 及 .shx。

若要接收一組形狀表示法的檔案，請在「匯入空間資料」視窗的**檔名**欄位中鍵入檔案共有的名稱。請勿鍵入副檔名。這樣，您就可以確定所有您需要的檔案（.shp 檔、.shx 檔及所有其它可能包含的檔案）都會匯入。

例如，假設一組 ESRI 形狀表示法儲存在檔案 Lakes.shp 及 Lakes.shx 中。在匯入這些表示法時，您只能在「檔名」欄位中鍵入 Lakes。

SDE 傳送檔具有名稱，但沒有副檔名。因此，在匯入 SDE 傳送檔時，您要在「檔名」欄位中鍵入其名稱，但不要鍵入任何副檔名。同樣地，在「匯入空間資料」視窗的**例外檔**欄位中，請勿鍵入您要指定的檔案的副檔名。只要鍵入檔案的名稱。

關於匯入形狀資料的必要授權，請參閱第87頁的『授權』。關於匯入 ESRI_SDE 資料的必要授權，請參閱第85頁的『授權』。

將資料匯入新的或現存表格中的步驟概觀：

1. 開啓「匯入空間資料」視窗。
2. 指定檔案的路徑、名稱及格式，此檔案包含要匯入的資料。

3. 指定每一次確認之前要匯入多少記錄。
4. 若您要將空間資料匯入到一個即將建立的表格中，請提供此表格的名稱和直欄 (匯入資料) 的名稱。若將空間資料匯入現存的表格中，請指出要匯入資料的直欄。
5. 指定要和資料結合的空間參照系統。
6. 指定一個檔案來收集無法匯入的記錄。
7. 指示 **Spatial Extender** 匯入資料，若您在此視窗中定義表格，則還要建立表格，將要匯入資料的直欄登記成爲一個層。

將資料匯入新的或現存表格中的步驟明細：

1. 開啓「匯入空間資料」視窗。
 - a. 在「控制中心」視窗中，展開物件樹狀結構，直到在您執行 **Spatial Extender** 的伺服器下找到 **Databases** 資料夾爲止。
 - b. 按一下 **Databases** 資料夾。資料庫顯示在視窗的右邊內容窗格中。
 - c. 在您要匯入資料的資料庫上按一下右鍵，按一下崩現功能表中的 **Spatial Extender** --> **匯入空間資料**。出現「匯入空間資料」視窗。
2. 指定檔案的路徑、名稱及格式，此檔案包含要匯入的資料：
 - a. 使用**檔名欄位**來指定路徑和名稱。
 - b. 使用**檔案格式**方框來指定格式。格式包括：

形狀 此爲預設值。

ESRI_SDE

若您指定此格式，則**空間參照名稱欄位**預設爲此格式相關的空間參照系統名稱。

3. 使用**確定範圍**欄位，指定每一次確認之前要匯入的記錄數。例如，若要求 **DB2** 一次確認 100 筆記錄，請指定數字 100。

要訣: 若您要讓 **DB2** 只在處理完全部記錄之後才發出確認，請指定 0。

4. 指定要匯入資料的表格和直欄。
 - a. 使用**層綱目**方框，指定要匯入資料的表格綱目。
 - b. 指定表格和直欄：
 - 若表格尚不存在：
 - 1) 在**層表格**欄位中，鍵入表格的名稱。
 - 2) 在**層直欄**欄位中，鍵入直欄的名稱，此直欄要用來存放匯入的資料。**Spatial Extender** 會自動將此直欄登記爲一個層。
 - 若表格已存在：

- 1) 在**層表格欄位**中指定表格。表格中必須已包含您要用來匯入資料的直欄。此外，此直欄必須已登記成爲一個層。
- 2) 在**層直欄欄位**中，指定要存放匯入資料的直欄名稱。
5. 在**空間參照名稱欄位**中，鍵入或選取要和此資料結合的空間參照系統。(若資料是來自 **ESRI_SDE** 傳送檔，則相關空間參照系統的名稱會自動顯示在欄位中。)
6. 在**異常狀況檔欄位**中，指定一個新檔案的路徑和名稱，用來收集無法匯入的記錄。稍後，您可修正這些記錄，再從該檔案中匯入。
Spatial Extender 將建立此檔案；請勿指定已存在的檔案。
7. 按一下**確定**來匯入資料。另外，若您爲一個不存在的表格提供名稱，則此表格將會建立，且用來匯入資料的直欄將登記爲一個層。此外，也會建立您所指定的異常狀況檔。

在表格層次將資料匯入現存的表格

本節概述使用「控制中心」的 **Tables** 資料夾，從形狀檔或 **ESRI_SDE** 傳送檔將資料匯入現存表格的步驟。此概觀之後詳細說明如何完成每一個步驟。

關於匯入形狀資料的必要授權，請參閱第87頁的『授權』。關於匯入 **ESRI_SDE** 資料的必要授權，請參閱第85頁的『授權』。

將資料匯入現存表格中的步驟概觀：

1. 開啓「匯入空間資料」視窗。
2. 指定檔案的路徑和名稱，此檔案包含要匯入的資料。
3. 指定每一次確認之前要匯入多少記錄。
4. 指定直欄，直欄要用來包含您所匯入的空間資料。
5. 指定要和此資料結合的空間參照系統。
6. 指定一個檔案來收集無法匯入的記錄。
7. 指示 **Spatial Extender** 匯入資料，若您指定一個尚未建立的直欄，則還要建立此直欄並登記成爲一個層。

將資料匯入現存表格中的步驟明細：

1. 開啓「匯入空間資料」視窗。
 - a. 在「控制中心」視窗中，展開物件樹狀結構，直到爲您要匯入資料的資料庫找到 **Tables** 資料夾爲止。
 - b. 按一下 **Tables** 資料夾。表格顯示在視窗的右邊內容窗格中。
 - c. 在您要匯入資料的表格上按一下右鍵，按一下蹦現功能表中的 **Spatial Extender** --> **匯入空間資料**。出現「匯入空間資料」視窗。

2. 在**檔名**方框，指定檔案的路徑和名稱，此檔案包含要匯入的資料。
3. 使用**確定範圍**方框，指定每一次確認之前要匯入的記錄數。例如，若要求 DB2 一次確認 100 筆記錄，請指定數字 100。

要訣：若您要讓 DB2 只在處理完全部記錄之後才發出確認，請指定 0。

4. 指定直欄，直欄要用來包含您所匯入的空間資料。
 - 若直欄尚不存在於表格內，請使用**層直欄**方框來鍵入該直欄的名稱。
 - 若直欄已存在，請使用**層直欄**方框來選取或鍵入直欄的名稱。
5. 使用**空間參照名稱**方框，指定要和匯入的資料結合的空間參照名稱。
 - 若您要新增直欄到表格中，請鍵入或選取空間參照系統的名稱。
 - 若匯入的資料是針對一個現存直欄，請勿變動**空間參照名稱**方框。此方框顯示預設空間參照系統的名稱。
6. 在**異常狀況檔**欄位中，指定一個新檔案的路徑和名稱，用來收集無法匯入的記錄。稍後，您可修正這些記錄，再從該檔案中匯入。
Spatial Extender 將建立此檔案；請勿指定已存在的檔案。
7. 按一下**確定**來匯入資料。另外，若您指定一個尚不存在的直欄，此直欄將被建立並登記成爲一個層。此外，也會建立您所指定的異常狀況檔。

將資料匯出到形狀檔

本節概述將資料匯出到形狀檔的步驟。此概觀之後詳細說明如何完成每一個步驟。

關於執行這些步驟的必要授權，請參閱第83頁的『授權』。

從形狀檔中匯出資料的步驟概觀

1. 開啓「匯出空間資料」視窗。
2. 指定含有要匯出的空間資料的直欄。
3. 若您要匯出一個資料列子集，請向 Spatial Extender 識別此子集。
4. 指定檔案的路徑和名稱，此檔案用來接收匯出的資料。
5. 指示 Spatial Extender 去匯出資料。

將資料匯出到形狀檔的步驟明細：

1. 開啓「匯出空間資料」視窗。
 - a. 在「控制中心」視窗中，展開物件樹狀結構，直到您在含有空間資料的資料庫中找到 **Tables** 或 **Views** 資料夾爲止：
 - b. 按一下 **Tables** 或 **Views** 資料夾。表格或概略表顯示在視窗的右邊內容窗格中。

- c. 在表格或概略表上 (含有要匯出的資料) 按一下右鍵，按一下蹦現功能表中的 **Spatial Extender** --> **匯出空間資料**。出現「匯出空間資料」視窗。
2. 在**層直欄**欄位中，指定含有要匯出的空間資料的直欄名稱。
3. 若您要匯出一個表格列子集，請使用 **WHERE** 子句方框鍵入一個 **WHERE** 子句，為您所要的列指定準則。在此子句中，您可以只參照到表格或概略表中您要匯出資料的直欄。

只鍵入準則。省略關鍵字 **WHERE**。例如，若表格或概略表有一個 **STATE** 直欄，且您只想地理編碼的列含有此直欄中的 **MA** 值，請鍵入：

```
STATE='MA'
```

4. 在**檔名**欄位中，指定檔案的路徑和名稱，此檔案用來接收匯出的資料。
5. 按一下**確定**來匯出資料。

第6章 建立空間索引

本章說明如何使用「控制中心」建立空間資料的索引。

將您的資料移入空間直欄之後，您就可以開始建立空間索引。一般索引結構(如 B 樹狀結構)對表格資料執行線性一維排序。為空間作業啓用的表格資料不儲存成單一登錄，而是二維。例如，多邊形之類的空間幾何是由一個空間直欄或層次中的一些座標值組成。因為 B 樹狀結構索引無法處理空間資料類型，所以 Spatial Extender 建立一個稱為格線索引的專屬索引技術。格線索引是以 B 樹狀結構索引為基礎，後者已增強，能夠處理二維資料和索引空間直欄。格線索引支援三個層次而且對許多物件、大小和資料分佈提供良好效能。關於空間索引的其它資訊，請參閱第123頁的『第12章 空間索引』。

若要瞭解建立空間索引需要什麼授權，請參閱第79頁的『授權』。

使用「控制中心」建立空間索引

使用「控制中心」建立空間索引：

1. 在物件樹狀結構中，選取 **Tables** 資料夾。內容窗格顯示全部現存的表格。
2. 在內容窗格上，對要建立索引的表格按一下右鍵，然後按一下闕現功能表中的 **Spatial Extender --> 空間索引**。「空間索引」視窗開啓。
3. 在「空間索引」視窗上，按一下**建立**。「建立空間索引」視窗開啓。
4. 在**名稱**欄位中，鍵入要建立的新空間索引名稱。

註：Spatial Extender 無法讓您選擇索引的綱目名稱。Spatial Extender 會自動新增綱目並建立完整名稱。

5. 在**層直欄**欄位中，選取建立索引的層次。一層只能有一個空間索引。層次是向 Spatial Extender 定義或登記的空間直欄。
6. 在**格線大小**欄位中，鍵入要指定給每一個欄位的格線大小值。

透過增加資料格大小，輸入格線層次 (**最細**、**中間**，以及**最粗**)。因此第二層必須有資料格大大於第一層，而第三層則大大於第二層。

若要設定**最細**欄位、**中間**欄位或**最粗**欄位的格線大小，請按工作站鍵盤的上下移鍵。按上移鍵可以 0.1 度遞增大小；按下移鍵可以 0.1 度遞減大小。

您不需指定全部三個格線大小。

決定格線資料格大小

正確的格線大小是經由錯誤和試驗的程序而決定的。建議根據索引的物件之大約大小設定格線大小。設定得太小或太大的格線會降低效能。太小的格線會影響索引搜尋時的索引鍵值/物件比率。在本情況中，會建立太多索引鍵值而且傳回大量候選索引鍵值。就那些設定太大的格線而言，起始索引搜尋會傳回少數候選的索引鍵值，但是在最終表格掃描時可能會降低效能。

關於選取格線資料格大小和格線層次數目的其它資訊，請參閱第130頁的『選取格線資料格大小』。

第7章 擷取及分析空間資訊

建構空間索引之後，即可使用空間表格了。本章討論與擷取及分析空間資料相關的一些問題。本章包含各種擷取方法的概觀，並提供了利用空間函數查詢表格的範例。

執行空間分析的方法

您可以透過下列任何程式設計環境，利用 SQL 及空間函數來執行空間分析：

- 地理瀏覽器 (例如，ESRI 的 ArcExplorer)。
關於使用 ArcExplorer 的詳細資訊，請參閱使用 *ArcExplorer*，您可以從 ESRI 網站取得，網址是 <http://www.esri.com>。
- 交談式 SQL 陳述式。
您可以從「DB2 指令中心」、「DB2 指令視窗」或指令行處理器來輸入 交談式 SQL 陳述式。
- 使用者開發的應用程式 (例如，ODBC、JDBC 及內含的 SQL)。

建置空間查詢

本節說明如何利用空間函數及述詞來建置空間查詢。

空間函數與 SQL

Spatial Extender 包括對空間資料執行各種作業的函數。本節範例說明如何使用空間函數來建置屬於您自己的空間查詢。

表3提供空間函數及其所能執行的作業類型之列表。

表 3. 空間函數與作業

函數類型	作業範例
計算	計算兩點之間的距離
比較	尋找位於洪水區內的所有客戶
資料交換	將資料轉換成支援的格式
轉換	對某一點新增五英里半徑範圍

關於空間函數的詳細資訊，請參閱第131頁的『第13章 幾何與相關的空間函數』和第165頁的『第14章 SQL 查詢的空間函數』。

範例 1：比較

下列查詢搜尋各百貨公司與客戶之間的平均距離。本範例使用的空間函數為 ST_Distance 和 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
```

範例 2：資料交換

下列查詢搜尋居住於舊金山海灣區的客戶位置。本範例使用的空間函數為 ST_AsText (資料交換) 和 ST_Within。ST_AsText 可將 c.location 直欄內的空間資料轉換成 OGC TEXT 格式。

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

範例 3：計算

下列查詢搜尋超過 10.5 英里長的所有街道。本範例使用的空間函數為 ST_Length。

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

範例 4：轉換

此查詢搜尋居住於洪水區或洪水區邊界 2 英里內的客戶。本範例使用的空間函數為 ST_Buffer (轉換) 和 ST_Within。變數 :floodzone 是 C/C++ 中所施行、內含的 SQL 程式的主變數。

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

空間述詞與 SQL

空間函數中稱為空間述詞的一個特殊群組可增進查詢效能。空間述詞 (例如用來比較兩個多邊形是否重疊的 ST_Overlaps) 在時間和記憶體需求兩方面的執行成本高。因此，利用最佳化技術來使執行成本降至最低很重要。當您根據本節稍後說明的規則來使用空間述詞時，DB2 查詢最佳化工具會利用空間索引來增進查詢效能。關於空間述詞的詳細資訊，請參閱第144頁的『述詞函數』。用來開發空間索引的空間述詞如下：

- ST_Contains
- ST_Crosses
- ST_Disjoint
- ST_Distance
- ST_Envelope

- ST_Equals
- ST_Intersects
- ST_Overlaps
- ST_Touches
- ST_Within

有關全部空間函數及述詞的完整列示，請參閱第165頁的『第14章 SQL 查詢的空間函數』。

索引開發規則

若要利用空間述詞使空間查詢最佳化，則適用下列規則：

- 述詞必須用於 **WHERE** 子句中。
- 述詞必須位於比較的左邊。例如：


```
WHERE db2gse.ST_Within(c.location,:BayArea)=1
```
- 相等比較必須使用整數常數 1。


```
WHERE db2gse.ST_Within(c.location,:BayArea)=1
```
- 述詞中必須有一個空間直欄當作搜尋目標，而該直欄上必須建立一個空間索引。

索引開發範例

表4顯示建立空間查詢來開發空間索引的正確與不正確方法。

表 4. 索引開發規則

空間查詢	違反規則
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=1</pre>	本範例沒有違反任何條件。
<pre>SELECT * FROM customers c WHERE db2gse.ST_Distance(c.location,:SanJose)<10</pre>	本範例沒有違反任何條件。
<pre>SELECT * FROM customers c WHERE db2gse.ST_Length(c.location)>10</pre>	ST_Length 不是空間述詞。
<pre>SELECT * FROM customers c WHERE 1=db2gse.ST_Within(c.location,:BayArea)</pre>	述詞必須位於比較的左邊。
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(c.location,:BayArea)=2</pre>	相等比較必須使用整數常數 1。

表 4. 索引開發規則 (繼續)

空間查詢	違反規則
<pre>SELECT * FROM customers c WHERE db2gse.ST_Within(:SanJose,:BayArea)=1</pre>	<p>述詞中必須有一個空間直欄當作搜尋目標，而該直欄上必須建立一個空間索引。(SanJose 及 BayArea 不是空間直欄，因此沒有與它們相關的空間索引。)</p>

第8章 撰寫 Spatial Extender 的應用程式

本章說明如何使用 Spatial Extender 範例程式來撰寫應用程式，以處理和自行設定空間資訊。內含下列主題：

- 使用範例程式
- 範例程式步驟

使用範例程式

Spatial Extender 範例程式讓應用程式設計更容易。透過範例程式，您可以：

- 使例行空間程序自動化
- 將範例程式碼剪貼到您自己的應用程式
- 對於具有空間功能的資料庫，瞭解建立和維護的必要步驟

可使用範例程式來撰寫 Spatial Extender 的複雜作業，例如撰寫一個應用程式，其中使用資料庫介面來呼叫 Spatial Extender 儲存程序。從範例程式中，您可複製和自行設定您的應用程式。若您不熟悉 Spatial Extender 的程式設計步驟，您可執行範例程式來詳細顯示每一個步驟。首先，無論如何您都必須建立範例程式。請以範例 make 檔來執行此動作。關於建立及執行範例程式的指示，請參閱第23頁的『安裝驗證』。

範例程式步驟

第62頁的表5顯示範例程式步驟、相關的儲存程序、以及每一個步驟的說明。呼叫儲存程序的 C 函數顯示在第62頁的表5的「動作」直欄中，且以括弧括住。關於儲存程序的詳細資訊，請參閱第69頁的『第9章 儲存程序』。範例程式係以第10頁的『實務範例：保險公司更新其 GIS』中的實務範例為根據。

表 5. Spatial Extender 範例程式

範例程式步驟	動作	說明
啓用/停用空間資料庫	<ol style="list-style-type: none"> 1. 啓用空間資料庫 (gseEnableDB) 2. 停用空間資料庫 (gseDisableDB) 3. 啓用空間資料庫 (gseEnableDB) 	<ol style="list-style-type: none"> 1. 這是使用 Spatial Extender 的第一個必要步驟。已為空間作業而啓用的資料庫具有一組空間類型、一組空間函數、一組空間述詞、一個新的索引類型、以及一組管理表和概略表。 2. 當您對錯誤的資料庫啓用空間功能時，通常可執行此步驟。停用空間資料庫時，表示您除去一組空間類型、一組空間函數、一組空間述詞、一個新的索引類型、以及一組管理表和概略表。 註： 如果有物件是依附於啓用資料庫程序所建立的物件，則停用資料庫將失敗。例如，以 ST_Point 類型的空間直欄來建立表格，將造成停用資料庫失敗。這是因為表格依附於停用資料庫程序所要捨棄的類型 ST_Point。 3. 同於 1。
登記空間參照系統	<ol style="list-style-type: none"> 1. 為 CUSTOMERS 表格的 LOCATION 直欄登記空間參照系統 (gseEnableSref) 2. 為 OFFICES 表格的 LOCATION 直欄登記空間參照系統 (gseEnableSref) 3. 為 OFFICES 表格的 LOCATION 直欄取消登記空間參照系統 (gseEnableSref) 4. 為 OFFICES 表格的 ZONE 直欄重新登記空間參照系統 (gseEnableSref) 	<ol style="list-style-type: none"> 1. 此步驟定義一個新的空間參照系統 (SRS)，用來解譯 CUSTOMERS 表格的空間資料。空間參照系統包括的幾何資料，是可儲存於具備空間功能之資料庫的直欄。在 SRS 登記到一個特定層之後，該層可用的座標就可儲存於相關的 CUSTOMERS 表格直欄中。 2. 此步驟定義一個新的空間參照系統 (SRS)，用來解譯 OFFICES 層的空間資料。每一個表格層有一個 SRS 定義。OFFICES 表格層可能需要一個不同於 CUSTOMERS 表格層的相關 SRS。 3. 若您指定錯誤的 SRS 參數給層或空間直欄，則執行此步驟。當您取消登記 OFFICES 表格層的 SRS 時，表示您除去定義及其相關參數。 4. 此步驟定義一個新的空間參照系統 (SRS)，用來解譯 OFFICES 層的空間資料。

表 5. *Spatial Extender* 範例程式 (繼續)

範例程式步驟	動作	說明
建立空間表格	<ol style="list-style-type: none"> 1. 新增 LOCATION 直欄來變更 CUSTOMERS 表格 (gseSetupTables) 2. 建立 OFFICES 表格 (gseSetupTables) 	<ol style="list-style-type: none"> 1. CUSTOMERS 表格代表已在資料庫中保存數年的業務資料。ALTER TABLE 陳述式新增一個類型 ST_Point 的新直欄 (LOCATION)。此直欄將於後續步驟中對地址直欄地理編碼時移入資料。 2. OFFICES 表格代表保險公司每一家分公司的銷售區 (及其它資料)。在後續的步驟中，整個表格內將移入非 DB2 資料庫的屬性資料。此步驟將 SHAPE 檔案中的屬性資料匯入 OFFICES 表格內。
登記空間層	<ol style="list-style-type: none"> 1. 將 CUSTOMERS 表格中的 LOCATION 直欄登記為一個層 (gseRegisterLayer) 2. 取消登記 CUSTOMERS 表格中的 LOCATION 直欄 (gseUnregisterLayer) 3. 將 OFFICES 表格中的 ZONE 直欄登記為一個層 (gseRegisterLayer) 	<p>這些步驟向 Spatial Extender 將 LOCATION 和 ZONE 登記為層。在 Spatial Extender 公用程式 (例如地理編碼程式) 能夠移入或存取一個空間直欄之前，您需要將它登記為一個層。Spatial Extender 公用程式可存取某層之後，您仍然可以取消登記這一層。在取消登記為一個層之後，其相關直欄仍會繼續存在。</p>
移入空間層	<ol style="list-style-type: none"> 1. 對 CUSTOMERS 表格的 LOCATION 直欄，進行地址資料的地理編碼 (gseRunGC) 2. 使用添加模式來載入 OFFICES 表格 (gseImportShape) 3. 使用建立模式來載入 HAZARD_ZONE 表格 (gseImportShape) 	<ol style="list-style-type: none"> 1. 此步驟呼叫地理編碼公用程式來執行批次地理編碼。批次地理編碼通常於表格有極大部份需要地理編碼或重新地理編碼時執行。 2. 此步驟在 OFFICES 表格內載入 SHAPE 檔案中存在的空間資料。因為 OFFICES 表格存在，且已登記 OFFICES/ZONE 層，載入公用程式將添加新記錄到現存的表格中。 3. 此步驟在 HAZARD_ZONE 層內載入 SHAPE 檔案中存在的空間資料。因為表格和層不存在，所以載入資料之前，載入公用程式將建立表格和登記層。
登記地理編碼程式	<ul style="list-style-type: none"> • 如果地理編碼程式非預設，登記該地理編碼程式 (gseRegisterGc) • 對可能已登記的地理編碼程式取消登記 (gseUnregisterGc) • 如果地理編碼程式非預設，登記該地理編碼程式 (gseRegisterGc) 	

表 5. Spatial Extender 範例程式 (繼續)

範例程式步驟	動作	說明
啟用空間索引	<ol style="list-style-type: none"> 為 CUSTOMERS 表格的 LOCATION 直欄啟用空間索引 (gseEnableIdx) 為 OFFICES 表格的 ZONE 直欄啟用空間索引 (gseEnableIdx) 為 OFFICES 表格的 LOCATION 直欄啟用空間索引 (gseEnableIdx) 為 HAZARD_ZON 表格的 BOUNDRY 直欄啟用空間索引 (gseEnableIdx) 	這些步驟啟用 CUSTOMERS、OFFICES 及 HAZARD_ZONE 表格的空間索引。
啟用自動地理編碼	<ol style="list-style-type: none"> 為 CUSTOMERS 表格的 LOCATION 和 ADDRESS 直欄啟用自動地理編碼功能 (gseEnableAutoGC) 	此步驟開啓地理編碼程式的自動呼叫功能。使用自動地理編碼功能，可在後續的插入和更新作業中，讓 CUSTOMERS 表格的 LOCATION 和 ADDRESS 直欄保持同步。
插入/更新/刪除 CUSTOMERS 表格	<ol style="list-style-type: none"> 插入不同街道的記錄 (gseInsDelUpd) 使用新地址來更新部份記錄 (gseInsDelUpd) 刪除表格中的所有記錄 (gseInsDelUpd) 	這些步驟在 CUSTOMERS 表格的 LOCATION 直欄上示範插入、更新和刪除作業。一旦啟用自動地理編碼功能，當 LOCATION 直欄中插入或更新 ADDRESS 直欄的資訊時，資訊就會自動地理編碼。此程序在前一個步驟中已啟用。
停用自動地理編碼	<ol style="list-style-type: none"> 停用 CUSTOMERS 層的自動地理編碼功能 (gseDisableAutoGC) 停用 CUSTOMERS 層的空間索引 (gseDisableIdxCustomersLayer) 	這些步驟停用地理編碼程式和空間索引的自動呼叫功能，為下一個步驟做準備 (下一個步驟對整個 CUSTOMERS 表格重新地理編碼)。若您載入大量地理資料，則於載入資料之前，建議您先停用空間索引，在完成載入之後再重新啟用。
對 CUSTOMERS 表格重新進行地理編碼	<ol style="list-style-type: none"> 以較低的精準度對 CUSTOMERS 層重新進行地理編碼 - 以 90% 來代替 100% (gseRunGC) 重新啟用 CUSTOMERS 層的空間索引 (gseEnableIdx) 以較低的精準度重新啟用自動地理編碼功能 - 以 90% 來代替 100% (gseEnableAutoGC) 	這些步驟再次以批次模式執行地理編碼程式、以新的精準度重新啟用自動地理編碼功能、重新啟用空間索引和自動地理編碼功能。當空間管理者發現地理編碼程序的失敗率極高時，建議執行此動作。若精準度設定為 100%，可能無法對地址地理編碼，因為在參照資料中找不到符合的地址。降低精準度，地理編碼程式才有機會找到符合的資料。重新以批次模式地理編碼表格之後，自動地理編碼和空間索引會重新啟用，協助加強維護空間索引和空間直欄，供後續的插入和更新作業使用。

表 5. *Spatial Extender* 範例程式 (繼續)

範例程式步驟	動作	說明
建立一個概略表，將其空間直欄登記成為概略表層。	<ol style="list-style-type: none"> 經由合併 CUSTOMERS 表格和 HAZARD_ZONE 表格，建立一個概略表 HIGHRISK_CUSTOMERS (gseCreateView) 將概略表的空間直欄登記成為概略表層 (gseRegisterLayer) 	這些步驟建立一個概略表，將其空間直欄登記成為概略表層。
執行空間分析	<ol style="list-style-type: none"> 找出客戶與每一間辦公室的平均距離 (ST_Within, ST_Distance) 找出每一間辦公室的平均客戶收入和保險費 (ST_Within) 找出現存的辦公室未含括的客戶 (ST_Within) 找出每一個辦公區重疊的危險區域數目 (ST_Overlaps) 假設辦公室位於辦公區的中心點，找出最接近特定客戶位置的辦公室 (ST_Distance, ST_Centroid) 找出接近特定危險區邊界的客戶位置 (ST_Buffer, ST_Overlaps) 找出特定辦公室所包括的高風險客戶 <p>(所有步驟皆利用 gseRunSpatialQueries)</p>	這些步驟使用 DB2 SQL 語言的空間述詞和函數來執行空間分析。DB2 查詢最佳化工具開發空間直欄的空間索引，盡量改進查詢效能。
將空間層匯出到檔案	匯出 highRiskCustomers 層 (gseExportShape)	此步驟舉例說明如何將查詢結果匯出到 SHAPE 檔案。將查詢結果匯出到另一個檔案格式，可讓協力廠商工具 (例如，ESRI ArcExplorer Java Version 3.0) 使用這些資訊。

第2篇 參照資料

第9章 儲存程序

本章提供的儲存程序，可讓您使用 Spatial Extender 來建置地理資訊系統。當您從「控制中心」啓用和使用 Spatial Extender 時，您就隱含地呼叫了這些儲存程序。例如，當您在 Spatial Extender 視窗中按一下**確定**時，DB2 會為您呼叫該視窗相關的儲存程序。另外，您可以在應用程式中明確地呼叫儲存程序。建議您在這類程式中包括標頭檔 db2gse.h。此檔案包含您指定給儲存程序參數的常數之巨集定義。在 AIX，它是儲存在 \$DB2INSTANCE/sqlib/include/ 目錄中。在 Windows NT 上，此檔案儲存於 %DB2PATH%\include\ 目錄中。

注意：

儲存程序輸入參數的全部字串常數都須區分大小寫。若要尋找需要常數的參數，請參閱本章中的表格。

您必須以大寫字體指定所有欲指定給參數的綱目名稱、表格名稱、概略表名稱、直欄名稱或層名稱。

在呼叫儲存程序之前，您必須隱含地或明確地連接到有安裝 Spatial Extender 的資料庫。您使用的第一個儲存程序為 db2gse.gse_enable_db。它啓用該資料庫以進行空閒作業。唯有已啓用該資料庫之後，您才可以使用其它儲存程序。

儲存程序的施行是保存在 Spatial Extender 伺服器上的 db2gse 程式庫中。

您可以使用下列列示，按名稱或按執行的作業來查閱儲存程序。第一個列示顯示名稱：

- 第71頁的『db2gse.gse_disable_autogc』
- 第73頁的『db2gse.gse_disable_db』
- 第74頁的『db2gse.gse_disable_sref』
- 第75頁的『db2gse.gse_enable_autogc』
- 第78頁的『db2gse.gse_enable_db』
- 第79頁的『db2gse.gse_enable_idx』
- 第81頁的『db2gse.gse_enable_sref』
- 第83頁的『db2gse.gse_export_shape』
- 第85頁的『db2gse.gse_import_sde』

- 第87頁的『db2gse.gse_import_shape』
- 第90頁的『db2gse.gse_register_gc』
- 第92頁的『db2gse.gse_register_layer』
- 第98頁的『db2gse.gse_run_gc』
- 第100頁的『db2gse.gse_unregist_gc』
- 第101頁的『db2gse.gse_unregist_layer』

下一個列示顯示儲存程序執行的作業。

- 建立空間直欄的索引 (請參閱第79頁的『db2gse.gse_enable_idx』)。
- 建立空間參照系統 (請參閱第81頁的『db2gse.gse_enable_sref』)。
- 停用地理編碼程式，使它不能自動使空間直欄與其相對應的屬性直欄同步化 (請參閱第71頁的『db2gse.gse_disable_autogc』)。
- 在資料庫中停用空間作業的支援 (請參閱第73頁的『db2gse.gse_disable_db』)。
- 捨棄空間參照系統 (請參閱第74頁的『db2gse.gse_disable_sref』)。
- 啓用資料庫支援空間作業 (請參閱第78頁的『db2gse.gse_enable_db』)。
- 啓用地理編碼程式，使空間直欄自動與其相對應的屬性直欄同步化 (請參閱第75頁的『db2gse.gse_enable_autogc』)。
- 從形狀檔匯出層及其相關的表格 (請參閱第83頁的『db2gse.gse_export_shape』)。
- 從 ESRI_SDE 傳送檔匯入層及其相關的表格 (請參閱第85頁的『db2gse.gse_import_sde』)。
- 從形狀檔匯入層及其相關的表格 (請參閱第87頁的『db2gse.gse_import_shape』)。
- 登記一個非預設的地理編碼程式 (請參閱第90頁的『db2gse.gse_register_gc』)。
- 將一個空間直欄登記為一個層 (請參閱第92頁的『db2gse.gse_register_layer』)。
- 以批次模式執行地理編碼程式 (請參閱第100頁的『db2gse.gse_unregist_gc』)。
- 取消登記一個非預設的地理編碼程式 (請參閱第101頁的『db2gse.gse_unregist_layer』)。
- 取消登記一個層 (請參閱第101頁的『db2gse.gse_unregist_layer』)。

關於您執行這些作業的順序，請參閱第3頁的『第1章 關於 Spatial Extender』和第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

db2gse.gse_disable_autogc

對於同步化空間直欄及其相關屬性直欄的觸發函式，可使用此儲存程序來捨棄或暫停此觸發函式。例如，當您以批次模式對屬性直欄中的值進行地理編碼時，建議您停用觸發函式。詳細資訊，請參閱第45頁的『關於地理編碼』。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 `C` 函數 `gseDisableAutoGc`。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，必須尤其在權限、專用權或專用權組中具有授權。

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含的表格定義了要捨棄或暫停的觸發函式。
- 此表格中的 CONTROL 專用權。
- 此表格中的 ALTER、SELECT 及 UPDATE 專用權。

輸入參數

表 6. `db2gse.gse_disable_autogc` 儲存程序的輸入參數。

名稱	資料類型	說明
<code>operMode</code>	SMALLINT	指出是否要捨棄或暫停觸發函式。 捨棄觸發函式對於 SQL 陳述式沒有任何影響。 可以重建暫時停用的觸發函式，而不必重新指定之前設定的參數。 此參數不可為 NULL。 註解： 若要捨棄觸發函式，請使用 <code>GSE_AUTOGC_DROP</code> 巨集。若只要暫停，請使用 <code>GSE_AUTOGC_INVALIDATE</code> 巨集。若要找出和這些巨集相關的值，請查詢 <code>db2gse.h</code> 檔案。在 AIX 上，此檔案儲存於 <code>\$DB2INSTANCE/sqlib/include/</code> 目錄中。在 Windows NT 上，此檔案儲存於 <code>%DB2PATH%\include\</code> 目錄中。

表 6. *db2gse.gse_disable_autogc* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格或概略表所屬於的綱目名稱。 此參數可為 NULL。 註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 db2gse.gse_disable_autogc 儲存程序的使用者 ID。
layerTable	VARCHAR(128)	表格的名稱，表格中定義您要捨棄或暫停的觸發函式。 此參數不可為 NULL。
layerColumn	VARCHAR(128)	具空間功能的直欄的名稱，由您要捨棄或暫停的觸發函式來維護該直欄。 此參數不可為 NULL。

輸出參數

表 7. *db2gse.gse_disable_autogc* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_disable_db

使用此儲存程序可除去資源，這些資源容許 Spatial Extender 儲存空間資料和支援對此資料執行的作業。

此儲存程序的目的是爲了協助您在啓用資料庫的空間作業功能之後，以及在資料庫中新增任何空間表格直欄或資料之前，解決這段期間發生的問題。例如，在您啓用資料庫的空間作業之後，決定對另一個資料庫使用 Spatial Extender。只要您未定義任何空間直欄或匯入任何空間資料，就可呼叫此儲存程序，從第一個資料庫中除去全部空間資源。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 gseDisableDB。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，對於要除去 Spatial Extender 資源的資料庫，必須具備 SYSADM 或 DBADM 權限。

輸出參數

表 8. db2gse.gse_disable_db 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_disable_sref

使用此儲存程序來捨棄一個空間參照系統。處理此儲存程序時，DB2GSE.SPATIAL_REF_SYS 型錄概略表中會除去關於空間參照系統的資訊。關於此概略表的資訊，請參閱第121頁的『DB2GSE.SPATIAL_REF_SYS』。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 gseDisableSref。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

皆非必要的。

輸入參數

表 9. db2gse.gse_disable_sref 儲存程序的輸入參數。

名稱	資料類型	說明
srefId	INTEGER	要捨棄之空間參照系統的數值識別字。 此參數不可為 NULL。

輸出參數

表 10. db2gse.gse_disable_sref 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

限制

在捨棄空間參照系統之前，您必須先取消登記使用此系統的任何層。如果不要求剩餘的此類層，則捨棄空間參照系統的要求將被拒絕。

db2gse.gse_enable_autogc

此儲存程序的用途：

- 建立觸發函式，用來保持空間直欄和其相關屬性直欄的同步化。每次在屬性直欄中插入值時或更新其中的值時，將有一個觸發函式呼叫一個已登記的地理編碼程式，對已插入或更新的值進行地理編碼，將結果資料放入空間直欄內。
- 使已經暫停的觸發函式再生效。
- 確定對已插入或更新的值進行地理編碼時所用的函數。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 `gseEnableAutoGC`。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，必須尤其在權限、專用權或專用權組中具有授權。

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含的表格定義了此儲存程序所建立的觸發函式。
- 此表格中的 CONTROL 專用權。
- 此表格中的 ALTER、SELECT 及 UPDATE 專用權。

輸入參數

表 11. `db2gse.gse_enable_autogc` 儲存程序的輸入參數。

名稱	資料類型	說明
<code>operMode</code>	SMALLINT	值，指定第一次要建立起始地理編碼的觸發函式，或於暫停之後再生效。 此參數不可為 NULL。 註解： 若要建立觸發函式，請使用 <code>GSE_AUTOGC_CREATE</code> 巨集。若要使觸發函式再生效，請使用 <code>GSE_AUTOGC_RECREATE</code> 巨集。若要找出和這些巨集相關的值，請查詢 <code>db2gse.h</code> 檔案。在 AIX 上，此檔案儲存於 <code>\$DB2INSTANCE/sqlib/include/</code> 目錄中。在 Windows NT 上，此檔案儲存於 <code>%DB2PATH%\include\</code> 目錄中。

表 11. db2gse.gse_enable_autogc 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格所屬於的綱目名稱。 此參數可為 NULL。 註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 db2gse.gse_enable_autogc 儲存程序的使用者 ID。
layerTable	VARCHAR(128)	表格的名稱，指此儲存程序所建立或再生效之觸發函式所要處理的表格。 此參數不可為 NULL。
layerColumn	VARCHAR(128)	空間直欄的名稱，指此儲存程序所建立或再生效之觸發函式所維護的空間直欄。 此參數不可為 NULL。
gcId	INTEGER	地理編碼程式的識別字，指此儲存程序所建立或再生效之插入或更新觸發函式所呼叫的地理編碼程式。 當 operMode 參數設定為 GSE_AUTOGC_CREATE 時，此參數不可為 NULL。當 operMode 設定為 GSE_AUTOGC_RECREATE 時，則可為 NULL。
precisionLevel	INTEGER	為了讓地理編碼程式能夠順利處理來源資料，來源資料和相對應參照資料之間的相配程度。 當 operMode 參數設定為 GSE_AUTOGC_CREATE 時，此參數不可為 NULL。當 operMode 設定為 GSE_AUTOGC_RECREATE 時，則可為 NULL。 註解： 精準度的範圍從 1 至 100 百分比。
vendorSpecific	VARCHAR(256)	供應商所提供的技術資訊；例如，供應商用來設定參數的檔案路徑和名稱。 當 operMode 參數設定為 GSE_AUTOGC_CREATE 時，此參數不可為 NULL。當 operMode 設定為 GSE_AUTOGC_RECREATE 時，則可為 NULL。

輸出參數

表 12. *db2gse.gse_enable_autogc* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

限制

- `layerColumn` 參數必須參照一個已登記成爲表格層的直欄。
- 若 `operMode` 參數設定爲 `GSE_AUTOGC_CREATE`，您必須將已登記之地理編碼程式的識別字指定給 `gcId` 參數。

db2gse.gse_enable_db

使用此儲存程序可提供資源給資料庫，這些資源是儲存空間資料和支援作業時所必要的。這些資源包括空間資料類型、空間索引類型、型錄表格和概略表、提供的函數、以及其它儲存程序。此儲存程序的外部程式庫及函數名稱是 db2gse.gse_enable_db。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 gseEnableDB。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫儲存程序的使用者 ID，對於要啓用的資料庫，必須具備 SYSADM 或 DBADM 權限。

輸出參數

表 13. db2gse.gse_enable_db 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_enable_idx

使用此儲存程序可建立空間直欄的索引。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 `gseEnableIdx`。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含的表格要用到已啓用的索引。
- 此表格的 CONTROL 或 INDEX 專用權。

輸入參數

表 14. *db2gse.gse_enable_idx* 儲存程序的輸入參數。

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格所屬於的綱目名稱。 此參數可為 NULL。 註解： 您必須提供此參數的值。參數可以是 NULL 值。
layerTable	VARCHAR(128)	表格的名稱，表格中定義您所建立的索引。 此參數不可為 NULL。
layerColumn	VARCHAR(128)	具空間功能之直欄的名稱，指要使用您所建立的索引來搜尋的直欄。 此參數不可為 NULL。
indexName	VARCHAR(128)	要建立的索引之名稱。 此參數不可為 NULL。 註解： 請勿指定綱目名稱。Spatial Extender 自動指定索引給 layerSchema 參數所參考到的綱目。
gridSize1	DOUBLE	數字，指出最細索引格線的顆粒性。 此參數不可為 NULL。

表 14. *db2gse.gse_enable_idx* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
gridSize2	DOUBLE	<p>數字，表示 (1) 此索引沒有第二格線，或 (2) 第二格線的顆粒性。</p> <p>此參數可為 NULL。</p> <p>註解： 若無第二格線，請指定 0。若您想要第二格線，則顆粒性必須小於 <code>gridSize1</code> 所表示的格線。</p>
gridSize3	DOUBLE	<p>數字，表示 (1) 此索引沒有第三格線，或 (2) 第三格線的顆粒性。</p> <p>此參數可為 NULL。</p> <p>註解： 若無第三格線，請指定 0。若您想要第三格線，則顆粒性必須小於 <code>gridSize2</code> 所表示的格線。</p>

輸出參數

表 15. *db2gse.gse_enable_idx* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_enable_sref

使用此儲存程序可指定特定座標系統中的負數和小數如何轉換成正整數，以便 Spatial Extender 能夠儲存。您的規格通稱為一個空間參照系統。處理此儲存程序時，DB2GSE.SPATIAL_REF_SYS 型錄概略表中會新增關於空間參照系統的資訊。關於此概略表的資訊，請參閱第121頁的『DB2GSE.SPATIAL_REF_SYS』。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 gseEnableSref。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

皆非必要的。

輸入參數

表 16. db2gse.gse_enable_sref 儲存程序的輸入參數。

名稱	資料類型	說明
srId	INTEGER	空間參照系統的數值識別字。 此參數不可為 NULL。 註解： 此識別字在具空間功能的資料庫中必須是唯一的。
srName	VARCHAR(64)	空間參照系統的簡短說明。 此參數不可為 NULL。 註解： 此說明在具空間功能的資料庫中必須是唯一的。
falsex	DOUBLE	一個數字，從負數 X 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。 此參數不可為 NULL。
falsey	DOUBLE	一個數字，從負數 Y 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。 此參數不可為 NULL。
xyunits	DOUBLE	一個數字，乘以一個小數 X 座標或小數 Y 座標時，得到一個整數，可儲存成 32 位元的資料項目。 此參數不可為 NULL。

表 16. *db2gse.gse_enable_sref* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
falsez	DOUBLE	一個數字，從負數 Z 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。 此參數不可為 NULL。
zunits	DOUBLE	一個數字，乘以一個小數 Z 座標時，得到一個整數，可儲存成 32 位元的資料項目。 此參數不可為 NULL。
falsem	DOUBLE	一個數字，從負數單位中扣除時，產生一個非負數 (亦即，一個正數或零)。 此參數不可為 NULL。
munits	DOUBLE	一個數字，乘以一個小數單位時，得到一個整數，可儲存成 32 位元的資料項目。 此參數不可為 NULL。
scId	INTEGER	衍生出空間參照系統之座標系統的數值識別字。若要找出座標系統的數值識別字，請查詢 DB2GSE.COORD_REF_SYS 型錄概略表第119頁的『DB2GSE.COORD_REF_SYS』。 此參數不可為 NULL。

輸出參數

表 17. *db2gse.gse_enable_sref* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_export_shape

使用此儲存程序可將一個層及其相關的表格匯到一個形狀檔，或建立新的形狀檔，再將一個層及其相關的表格匯到此新檔案。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 `gseExportShape`。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，對於要匯出的表格，必須具備 `SELECT` 專用權。

輸入參數

表 18. *db2gse.gse_export_shape* 儲存程序的輸入參數。

名稱	資料類型	說明
<code>layerSchema</code>	<code>VARCHAR(30)</code>	<code>layerTable</code> 參數中指定之表格所屬於的綱目名稱。 此參數可為 <code>NULL</code> 。 註解： 若您未提供 <code>layerSchema</code> 參數的值，則預設為呼叫 <code>db2gse.gse_export_shape</code> 儲存程序的使用者 ID。
<code>layerTable</code>	<code>VARCHAR(128)</code>	您要匯出之表格的名稱。 此參數不可為 <code>NULL</code> 。
<code>layerColumn</code>	<code>VARCHAR(30)</code>	直欄的名稱，已登記成您要匯出的層。 此參數不可為 <code>NULL</code> 。
<code>fileName</code>	<code>VARCHAR(128)</code>	形狀檔的名稱，表示指定的層要匯到此檔案。 此參數不可為 <code>NULL</code> 。
<code>whereClause</code>	<code>VARCHAR(1024)</code>	<code>where</code> 子句的主體。對於要匯出的列集，定義限制。此子句可參照您所匯出之表格內的任何屬性直欄。在此子句中不需要關鍵字 <code>WHERE</code> 。 此參數可為 <code>NULL</code> 。

輸出參數

表 19. *db2gse.gse_export_shape* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

限制

您一次只能匯出一層。

db2gse.gse_import_sde

使用此儲存程序可將一個 SDE 傳送檔匯入到一個已啓用空間作業功能的資料庫中。此儲存程序有兩種操作方法：

- 若 SDE 傳送檔的目標是一個含有已登記之層直欄的現存表格，則 Spatial Extender 將在表格中載入檔案的資料。
- 否則，Spatial Extender 將建立一個含有空間直欄的表格，將此直欄登記成一個層，在該層和表格的其它直欄中載入檔案的資料。

SDE 傳送檔中指定的空間參照系統，將和 Spatial Extender 中已登記的空間參照系統做比較。若指定的系統符合一個已登記的系統，傳送資料中的負數值和小數值於載入時，將依照已登記之系統所預設的方法來修改。若指定的系統不符合任何已登記的系統，則 Spatial Extender 將建立一個新的空間參照系統，以訂定修改方式。

授權

當您在現存的表格中匯入資料時，呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含要匯入資料的表格。
- 此表格中的 CONTROL 專用權。

當您要匯入資料的表格必須建立時，呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含要建立的表格。

輸入參數

表 20. db2gse.gse_import_sde 儲存程序的輸入參數。

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格或概略表所屬於的綱目名稱。 此參數可為 NULL。 它必須小於或等於 30 個字元。 註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 db2gse.gse_import_sde 儲存程序的使用者 ID。

表 20. *db2gse.gse_import_sde* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
layerTable	VARCHAR(128)	表格的名稱，指要載入 SDE 傳送資料的表格。 此參數不可為 NULL。
layerColumn	VARCHAR(30)	已登記成爲層的表格的名稱，指要載入 SDE 傳送檔之空間資料的表格。 此參數不可為 NULL。 它必須小於或等於 30 個字元。
fileName	VARCHAR(128)	要匯入的 SDE 傳送檔的名稱。 此參數不可為 NULL。
commitScope	INTEGER	每一個檢查點的記錄數。 此參數可為 NULL。

輸出參數

表 21. *db2gse.gse_import_sde* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_import_shape

使用此儲存程序可將一個 ESRI 形狀檔匯入到一個資料庫中，此資料庫已啟用空間作業的功能。此儲存程序有兩種操作方法：

- 若形狀檔的目標是一個含有已登記之層直欄的現存表格，則 Spatial Extender 將在表格中載入檔案的資料。
- 否則，Spatial Extender 將建立一個含有空間直欄的表格，將此直欄登記成一個層，在該層和表格的其它直欄中載入檔案的資料。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式碼中的 C 函數 gseImportShape。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

當您匯入一組 ESRI 形狀表示法時，您至少會收到兩個檔案。所有檔案的檔名字首都相同，但副檔名不同。例如，您一定會收到的兩個檔案的副檔名是 .shp 及 .shx。

若要接受一組形狀表示法的檔案，請對 fileName 參數指定檔案共有的名稱。請勿指定副檔名。這樣，您就可以確定所有您需要的檔案（.shp 檔、.shx 檔及所有其它可能包含的檔案）都會匯入。

例如，假設一組 ESRI 形狀表示法儲存在檔案 Lakes.shp 及 Lakes.shx 中。在匯入這些表示法時，您只能對 fileName 參數指定名稱 Lakes。

SDE 傳送檔具有名稱，但沒有副檔名。因此，在匯入 SDE 傳送檔時，您要對 fileName 參數指定其名稱，但不要指定副檔名。

授權

呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含要載入形狀資料的表格。
- 此表格中的 CONTROL 專用權。

輸入參數

表 22. *db2gse.gse_import_shape* 儲存程序的輸入參數。

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格或概略表所屬於的綱目名稱。 此參數可為 NULL。 註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 db2gse.gse_import_shape 儲存程序的使用者 ID。
layerTable	VARCHAR(128)	表格的名稱，指要載入已匯入之形狀檔的表格。 此參數不可為 NULL。
layerColumn	VARCHAR(30)	已登記成為層的表格的名稱，指要載入形狀資料的表格。 此參數不可為 NULL。
fileName	VARCHAR(128)	要匯入的形狀檔的名稱。 此參數不可為 NULL。
exceptionFile	VARCHAR(128)	檔案的路徑和名稱，此檔案儲存著無法匯入的形狀。此為 db2gse.gse_import_shape 儲存程序執行時將建立的一個新檔案。 請對 exceptionFile 參數指定檔名，但不指定副檔名。 此參數不可為 NULL。
srid	INTEGER	空間參照系統的識別字，指要載入形狀資料的層所用的空間參照系統。 此參數可為 NULL。 註解： 若未指定此識別字，則內部轉換將設定成形狀檔的最大可能解析度。
commitScope	INTEGER	每一個檢查點的記錄數。 此參數可為 NULL。

輸出參數

表 23. *db2gse.gse_import_shape* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_register_gc

使用此儲存程序可登記非預設的地理編碼程式。若要判斷是否已登記一個地理編碼程式，請查詢 DB2GSE.SPATIAL_GEOCODER 型錄概略表（說明於第120頁的『DB2GSE.SPATIAL_GEOCODER』）。

授權

呼叫此儲存程序的使用者 ID，對於含有此儲存程序所登記之地理編碼程式的資料庫，必須具備 SYSADM 或 DBADM 權限。

輸入參數

表 24. db2gse.gse_register_gc 儲存程序的輸入參數。

名稱	資料類型	說明
gcId	INTEGER	您所登記之地理編碼程式的數值識別字。 此參數不可為 NULL。 註解： 此識別字在資料庫中必須是唯一的。
gcName	VARCHAR(64)	您所登記之地理編碼程式的簡短說明。 此參數不可為 NULL。 註解： 此說明在資料庫中必須是一個唯一字串。
vendorName	VARCHAR(64)	供應商的名稱，指提供您所登記之地理編碼程式的供應商。 此參數不可為 NULL。
primaryUDF	VARCHAR(256)	您所登記之地理編碼程式的完整名稱。 此參數不可為 NULL。
precisionLevel	INTEGER	為了讓地理編碼程式能夠順利處理來源資料，來源資料和相對應參照資料之間的相配程度。 此參數不可為 NULL。 註解： 精準度的範圍從 1 至 100 百分比。
vendorSpecific	VARCHAR(256)	供應商所提供的技術資訊；例如，供應商用來設定參數的檔案路徑和名稱。 此參數可為 NULL。

表 24. *db2gse.gse_register_gc* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
geoArea	VARCHAR(256)	要進行地理編碼的地理區。 此參數可為 NULL。
說明	VARCHAR(256)	供應商提供的備註。 此參數可為 NULL。

輸出參數

表 25. *db2gse.gse_register_gc* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_register_layer

使用此儲存程序可將空間直欄登記成爲一個層。處理此儲存程序時，所要登記之層的相關資訊會新增到 DB2GSE.GEOMETRY_COLUMNS 型錄概略表中。關於此概略表的資訊，請參閱第120頁的『DB2GSE.GEOMETRY_COLUMNS』。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 gseRegisterLayer。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

此儲存程序將不會在下列表格類型中運作：

- A = 別名
- H = 階層表格
- N = 暱名
- S = 摘要表格
- U = 類型表格
- W = 類型概略表

授權

呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 在表格層方面：
 - 資料庫的 SYSADM 或 DBADM 權限，資料庫包含此層所屬於的表格。
 - 此表格的 CONTROL 或 ALTER 專用權。
- 在概略表層方面：
 - 基本表格的 SELECT 專用權，表格包含 (1) 此層中要進行地理編碼的位址資料 (2) 地理編碼所產生的空間資料。

輸入參數

表 26. *db2gse.gse_register_layer* 儲存程序的輸入參數。

名稱	資料類型	說明
layerSchema	INTEGER(30)	<p>layerTable 參數中指定之表格或概略表所屬於的綱目名稱。</p> <p>此參數可為 NULL。</p> <p>註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 <i>db2gse.gse_register_layer</i> 儲存程序的使用者 ID。</p>
layerTable	VARCHAR(128)	<p>表格或概略表的名稱，表格中包含要登記為層的直欄。</p> <p>此參數不可為 NULL。</p>
layerColumn	VARCHAR(128)	<p>指要登記成一個層的直欄的名稱。若是表格，如果直欄不存在，則 Spatial Extender 會使用 ALTER 陳述式來新增它。若是概略表，則直欄必須已經存在。</p> <p>只能對 layerColumn 參數指定一個直欄。因此，當您登記表格或概略表的多個直欄作為層時，您必須分別對每一個直欄執行此儲存程序。</p> <p>此參數不可為 NULL。</p>
layerTypeName	VARCHAR(64)	<p>指要登記成一個層的直欄的資料類型。只接受 Spatial Extender 提供的資料類型。您必須以大寫指定資料類型；例如：</p> <p>ST_POINT</p> <p>您不需要指定綱目名稱，因為它會自動新增。</p> <p>唯有當此直欄是處理此儲存程序時所要建立的表格直欄時，此參數不可為 NULL。否則，如果直欄為表格或概略表中的現存直欄，則此參數可為 NULL。</p>
srId	INTEGER	<p>使用於此層的空間參照系統的識別字。</p> <p>在表格層中，此參數不可為 NULL。當您登記一個概略表層時，Spatial Extender 會忽略此參數。</p>

表 26. *db2gse.gse_register_layer* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
geoSchema	VARCHAR(30)	<p>當您將一個概略表直欄登記成爲一個層時引用。geoSchema 參數指表格的綱目，表格是直欄所屬於之概略表的基礎。</p> <p>當您將一個概略表直欄登記成爲一個層時，此參數可爲 NULL。當您將一個表格直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>此參數不支援以一個以上的基本表格爲基礎的概略表或其它概略表。</p> <p>註解： 若您未提供 geoSchema 參數的值，則預設爲 layerSchema 參數的值。</p>
geoTable	VARCHAR(128)	<p>當您將一個概略表直欄登記成爲一個層時引用。geoTable 參數指表格的名稱，表格是直欄所屬於之概略表的基礎。</p> <p>此參數不支援以一個以上的基本表格爲基礎的概略表或其它概略表。</p> <p>當您將一個概略表直欄登記成爲一個層時，此參數不可爲 NULL。當您將一個表格直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p>
geoColumn	VARCHAR(128)	<p>當您將一個概略表直欄登記成爲一個層時引用。geoColumn 參數指表格直欄的名稱，表格直欄是此概略表直欄的基礎。</p> <p>此參數不支援以一個以上的基本表格爲基礎的概略表或其它概略表。</p> <p>當您將一個概略表直欄登記成爲一個層時，此參數不可爲 NULL。當您將一個表格直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p>
nAttributes	SMALLINT	<p>直欄數，直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p>

表 26. *db2gse.gse_register_layer* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
attr1Name	VARCHAR(128)	<p>第一個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>如果您打算使用預設地理編碼程式，則必須將街道位址儲存在 attr1Name 直欄中。</p>
attr2Name	VARCHAR(128)	<p>第二個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>如果您打算使用預設地理編碼程式，則必須將城市名稱儲存在 attr2Name 直欄中。</p>
attr3Name	VARCHAR(128)	<p>第三個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>如果您打算使用預設地理編碼程式，則必須將州名或其縮寫儲存在 attr3Name 直欄中。</p>
attr4Name	VARCHAR(128)	<p>第四個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>如果您打算使用預設地理編碼程式，則必須將郵遞區號儲存在 attr4Name 直欄中。</p>

表 26. *db2gse.gse_register_layer* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
attr5Name	VARCHAR(128)	<p>第五個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>預設的地理編碼程式會忽略 Attr5Name 直欄。</p>
attr6Name	VARCHAR(128)	<p>第六個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>預設的地理編碼程式會忽略 Attr6Name 直欄。</p>
attr7Name	VARCHAR(128)	<p>第七個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>預設的地理編碼程式會忽略 Attr7Name 直欄。</p>
attr8Name	VARCHAR(128)	<p>第八個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>預設的地理編碼程式會忽略 Attr8Name 直欄。</p>
attr9Name	VARCHAR(128)	<p>第九個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。</p> <p>當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。</p> <p>預設的地理編碼程式會忽略 Attr9Name 直欄。</p>

表 26. *db2gse.gse_register_layer* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
attr10Name	VARCHAR(128)	第十個直欄的名稱，該直欄中包含要針對此層進行地理編碼的來源資料。 當您將一個表格直欄登記成爲一個層時，此參數可爲 NULL。當您將一個概略表直欄登記成爲一個層時，Spatial Extender 會忽略此參數。 預設的地理編碼程式會忽略 Attr10Name 直欄。

輸出參數

表 27. *db2gse.gse_register_layer* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

限制

- 若您要將一個概略表直欄登記成爲一個層，則必須以已登記成爲層的表格直欄爲基礎。
- 對於您所登記的層來進行地理編碼的資料，最多只有十個屬性直欄可包含這種資料。

db2gse.gse_run_gc

使用此儲存程序可在批次模式下執行地理編碼程式。關於此作業的資訊，請參閱第48頁的『以批次模式執行地理編碼程式』。

關於呼叫此儲存程序的範例程式碼，請參閱範例程式中的 C 函數 `gseRunGC`。關於此程式的資訊，請參閱第61頁的『第8章 撰寫 Spatial Extender 的應用程式』。

授權

呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 資料庫的 SYSADM 或 DBADM 權限，資料庫包含要操作的指定地理編碼程式上的表格。
- 此表格的 CONTROL 或 UPDATE 專用權。

輸入參數

表 28. `db2gse.gse_run_gc` 儲存程序的輸入參數。

名稱	資料類型	說明
<code>layerSchema</code>	VARCHAR(30)	<code>layerTable</code> 參數中指定之表格或概略表所屬於的綱目名稱。 此參數可為 NULL。 註解： 若您未提供 <code>layerSchema</code> 參數的值，則預設為呼叫 <code>db2gse.gse_run_gc</code> 儲存程序的使用者 ID。
<code>layerTable</code>	VARCHAR(128)	表格的名稱，表格含有要插入地理編碼資料的直欄。 此參數不可為 NULL。
<code>layerColumn</code>	VARCHAR(128)	直欄的名稱，指要插入地理編碼資料的直欄。 此參數不可為 NULL。
<code>gcId</code>	INTEGER	您要執行的地理編碼程式的識別字。 此參數可為 NULL。 若要找出已登記之地理編碼程式的識別字，請查詢 <code>DB2GSE.SPATIAL_GEOCODER</code> 型錄概略表。

表 28. *db2gse.gse_run_gc* 儲存程序的輸入參數。(繼續)

名稱	資料類型	說明
precisionLevel	INTEGER	爲了讓地理編碼程式能夠順利處理來源資料，來源資料和相對應參照資料之間的相配程度。 此參數可爲 NULL。 註解：精準度的範圍從 1 至 100 百分比。
vendorSpecific	VARCHAR(256)	供應商所提供的技術資訊；例如，供應商用來設定參數的檔案路徑和名稱。 此參數可爲 NULL。
whereClause	VARCHAR(256)	SQL WHERE 子句的主體。對於要進行地理編碼的記錄集，定義限制。此子句可參照地理編碼程式所處理之表格內的任何屬性直欄。 此參數可爲 NULL。
commitScope	INTEGER	每一個檢查點的記錄數。 此參數可爲 NULL。

輸出參數

表 29. *db2gse.gse_run_gc* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_unregist_gc

使用此儲存程序可取消登記非預設的地理編碼程式。

關於您要取消登記之地理編碼程式的資訊，請查詢 DB2GSE.SPATIAL_GEOCODER 型錄概略表；請參閱第120頁的『DB2GSE.SPATIAL_GEOCODER』。

授權

呼叫此儲存程序的使用者 ID，對於要取消登記之地理編碼程式的資料庫，必須具備 SYSADM 或 DBADM 權限。

輸入參數

表 30. *db2gse.gse_unregist_gc* 儲存程序的輸入參數。

名稱	資料類型	說明
gcId	INTEGER	指要取消登記之地理編碼程式的識別字。 此參數不可為 NULL。

輸出參數

表 31. *db2gse.gse_unregist_gc* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

db2gse.gse_unregist_layer

使用此儲存程序可取消登記一個層。以下是此儲存程序的作法：

- 從 Spatial Extender 型錄表格中除去層的定義。
- 刪除 Spatial Extender 在此層的基本表格上所加諸的核對限制，確定此層的空間資料符合其空間參考系統的需求。
- 捨棄觸發函式，指新增、變更或除去位址資料時，用來更新空間直欄的觸發函式。

在地理編碼表格橫列中的位址資料時，結果空間資料會放在相同的橫列中。因此，如果刪除橫列，同時會刪除位址資料與空間資料。觸發函式不會刪除空間資料。

在處理儲存程序時，會從 DB2GSE.GEOMETRY_COLUMNS 型錄概略表中除去層的相關資訊。關於此概略表的資訊，請參閱第120頁的『DB2GSE.GEOMETRY_COLUMNS』。

授權

呼叫此儲存程序的使用者 ID，必須擁有下列其中一個權限或專用權：

- 在表格層方面：
 - 資料庫的 SYSADM 或 DBADM 權限，資料庫包含此層的基本表格。
 - 此表格的 CONTROL 或 ALTER 專用權。
- 在概略表層方面：
 - 基本表格的 SELECT 專用權，表格包含 (1) 針對此層進行地理編碼的位址資料 (2) 地理編碼所產生的空間資料。

輸入參數

表 32. *db2gse.gse_unregist_layer* 儲存程序的輸入參數。

名稱	資料類型	說明
layerSchema	VARCHAR(30)	layerTable 參數中指定之表格所屬於的綱目名稱。 此參數可為 NULL。 註解： 若您未提供 layerSchema 參數的值，則預設為呼叫 db2gse.gse_unregister_layer 儲存程序的使用者 ID。 您必須以大寫字體指定所有欲指定給參數的綱目名稱、表格名稱、概略表名稱、直欄名稱或層名稱。
layerTable	VARCHAR(128)	表格的名稱，表格包含 layerColumn 參數中指定的直欄。 此參數不可為 NULL。
layerColumn	VARCHAR(128)	空間直欄的名稱，直欄已定義成爲您要取消登記的層。 此參數不可為 NULL。 註解： 只能對 layerColumn 參數指定一個層。因此，當您在表格或概略表中取消登記多層時，您必須分別對每一個層執行此儲存程序。

輸出參數

表 33. *db2gse.gse_unregist_layer* 儲存程序的輸出參數。

名稱	資料類型	說明
msgCode	INTEGER	此儲存程序的呼叫程式可傳回之訊息的相關字碼。
msgText	VARCHAR(1024)	在 Spatial Extender 伺服器上所建構的完整錯誤訊息。

限制

若已定義成概略表層的概略表直欄是基於已定義成表格層的表格直欄，則在取消登記此概略表層之前，您無法取消登記此表格層。

第10章 訊息

DB2 Spatial Extender 產生的訊息會由下列所傳回：

- 控制中心
- 儲存程序
- 空間函數

每一個訊息都有一個由字首及訊息碼所組成的訊息 ID。訊息碼也稱為 *SQLCODE*。

共有三種訊息類型：錯誤、警告及參考。以 *E* 結尾的訊息 ID 表示錯誤訊息。以 *W* 結尾的訊息 ID 表示警告訊息。以 *I* 結尾的訊息 ID 表示參考訊息。

「控制中心」傳回的訊息

下列訊息是由「控制中心」傳回。其 *SQLCODE* 是以字母 "DBA" 起首。

DBA7200E 選取 10 個以上的直欄做為地理編碼程式的輸入

解說： 可選取 10 個以上的直欄做為地理編碼程式的輸入。

使用者回應： 將「所選取直欄」方框中的直欄名稱移動到「可用直欄」方框中，直到「所選取直欄」方框列示十個 (或少於十個) 名稱為止。

DBA7201E 未啓用資料庫供 Spatial Extender 使用。

解說： 使用 Spatial Extender 之前，必須先啓用資料庫供 Spatial Extender 使用。

使用者回應： 在資料庫上按一下右鍵，從功能表中選取「Spatial Extender」->「啓用」。

儲存程序傳回的訊息

下列訊息是由儲存程序傳回。其 *SQLCODES* 是以字母 "GSE" 起首，其後接著從 0000 到 2035 範圍內的數字。

警告： 兩種錯誤訊息 --GSE2022E 及 GSE2035E-- 不在 DB2 Spatial Extender 訊息型錄中。因此，當相關聯的儲存程序發生錯誤時，就會在線上以下列訊息通知您表示無法擷取：SQL10007N 無法擷取訊息 "<SQLCODE -2022 或 -2035>"。原因碼："4"。

GSE0000I 作業順利完成。

GSE0001E Spatial Extender 無法以使用者 ID “<user-id>” 的身份執行所要求的作業 (“<operation-name>”)。

解說： 您要求此作業時所用的使用者 ID，不具備執行此作業的專用權或權限。

使用者回應： 請查閱文件來找出適當的授權，或向 Spatial Extender 管理者取得授權。

GSE0002E “<value>” 不是 “<argument-name>” 引數的有效值。

解說： 您所輸入的值不正確或拼錯。

使用者回應： 請參閱文件或詢問 Spatial Extender 管理者，找出您需要指定的值或範圍值。

GSE0003E Spatial Extender 無法執行所要求的作業，因為尚未指定引數 “<argument-name>”。

解說： 您未指定此作業的必要引數。

使用者回應： 請以您想要的值來指定引數 “<argument-name>”，然後重新要求作業。

GSE0004W 未評估引數 “<argument-name>”。

解說： 您所要求的作業不使用引數 “<argument-name>”。

使用者回應： 皆非必要的。

GSE0005E Spatial Extender 無法處理您的要求來建立一個稱為 “<object-name>” 的物件。

解說： 可能是物件 “<object-name>” 已存在，或是您沒有適當的許可權可建立它。此物件可能是表格、直欄、觸發函式、索引、檔案、或其它類型的物件。

使用者回應： 若 “<object-name>” 就是您所要的物件，不用執行其它動作。否則，請正確地指定名稱，並確定您有正確的許可權來建立該物件。

GSE0006E Spatial Extender 無法對已啓用或已登記的物件 “<object-name>” 執行所要求的作業。

解說： 物件 “<object-name>” 已啓用或登記，或已存在。此物件可能是層、索引、空間參照系統、座標系統、地理編碼程式、或其它類型的物件。

使用者回應： 請確定物件 “<object-name>” 已存在，然後重新提出您的要求。

GSE0007E Spatial Extender 無法對您尚未啓用或登記的物件 “<object-name>” 執行所要求的作業。

解說： 物件 “<object-name>” 尚未啓用或登記。此物件可能是層、索引、空間參照系統、空間座標系統、地理編碼程式、或其它類型的物件。

使用者回應： 請啓用或登記物件 “<object-name>”。然後重新提出您的要求。

GSE0008E 發生非預期的 SQL 錯誤 (“<sql-error-message>”)。

使用者回應： 請查閱 SQL 錯誤訊息 “<sql-error-message>” 中和 SQLCODE 有關的詳細訊息。必要時，請洽您的 IBM 服務人員。

GSE0009E 無法對已存在的物件 “<object-name>” 執行所要求的作業。

解說： “<object-name>” 已存在於資料庫或作業系統中。此物件可能是檔案、表格、概略表、直欄、索引、觸發函式、或其它類型的物件。

使用者回應： 請確定已正確地指定所要嘗試存取的物件。必要時，請刪除該物件。

**GSE0010E 無法對可能不存在的物件
“<object-name>” 執行所要求的作
業。**

解說：“<object-name>” 不存在於資料庫或作業系統中。此物件可能是檔案、表格、概略表、直欄、索引、觸發函式、或其它類型的物件。

使用者回應：請確定您擁有正確的許可權來存取物件。若您具備此許可權，但物件不存在，則您需要建立物件。

**GSE0011E Spatial Extender 無法停用或取消
登記物件 “<object-name>”。**

解說：“<object-name>” 依附於另一個物件。“<object-name>” 可能是空間參照系統、層、地理編碼程式、或其它類型的物件。

使用者回應：請查閱文件來找出可依附的物件類型“<object-name>”。然後除去“<object-name>”所依附的特定物件。

**GSE0012E Spatial Extender 無法處理您的要
求，因為完整的空間直欄
“<layer-schema.layer-name.
layer-column>” 未登記成一個表格
層。**

解說：完整的空間直欄

“<layer-schema.layer-name.layer-column>” 必須先登記為一個表格層，您才能夠執行相關的作業（例如，啟用索引、啟用地理編碼程式來以批次模式移入、或自動更新）。

使用者回應：請在 Spatial Extender 型錄中檢查 DB2GSE.GEOMETRY_COLUMNS 概略表，確定完整的空間直欄

“<layer-schema.layer-name.layer-column>” 已登記成一個表格層。另外，請確定含有此直欄的表格亦包含相對應的有效屬性直欄。

GSE0013E 未啟用資料庫空間作業的功能。

解說：未啟用資料庫空間作業的功能。因此，Spatial Extender 型錄不存在。

使用者回應：請啟用資料庫的空間作業功能。

GSE0014E 資料庫已為空間作業而啟用。

解說：資料庫已為空間作業而啟用。

使用者回應：請驗證資料庫是否如您所期望地啟用。必要時，請停用資料庫。

**GSE0498E 發生下列錯誤：
“<error-message>”。**

**GSE0499W Spatial Extender 發出下列警告：
“<warning-message>”。**

**GSE0500E 您所指定的作業模式
（“<operation-mode>”）無效。**

解說：您所要求的作業不支援指定的模式。

使用者回應：請查閱文件來找出此作業支援的模式。

**GSE1001E Spatial Extender 無法登記一個稱
為 “<schema-name.view-name.
column-name>” 且基於空間直欄
“<schema-name.table-name.
column-name>” 的概略表層。**

解說：您所指定的空間直欄（“<schema-name.table-name.column-name>”）尚未登記成一個表格層。

使用者回應：將直欄

“<schema-name.table-name.column-name>” 登記成一個表格層。

GSE1002E Spatial Extender 無法登記一個稱為 “<schema-name.view-name.column-name>” 且基於表格 “<schema-name.table-name>” 的概略表層。

解說： 您所指定的表格 (“<schema-name.table-name>”) 不直接或間接做為概略表 “<schema-name.view-name.column-name>” 的基礎。

使用者回應： 找出概略表 “<schema-name.view-name.column-name>” 的基本表格，然後指定此表格。

GSE1003E Spatial Extender 無法存取表格或概略表 “<schema-name.object-name>” 中的直欄 “<column-name>”。

解說： 表格或概略表 “<schema-name.object-name>” 中沒有一個稱為 “<column-name>” 的直欄。

使用者回應： 檢查表格或概略表 “<schema-name.object-name>” 的定義，找出直欄的適當名稱。

GSE1004E Spatial Extender 無法將完整的空間直欄 “<schema-name.table-name.column-name>” 登記成為一個表格層。

解說： 直欄 “<schema-name.table-name.column-name>” 不是空間資料類型，或和基本表格不相關。

使用者回應： 請為 “<schema-name.table-name.column-name>” 定義一個空間資料類型，或確定此直欄是區域基本表格的一部份。

GSE1005E 您為概略表層所指定的空間參照系統 (“<view-layer-spatial-reference-id>”)，不同於此層的基礎表格層所使用的空間參照系統 (“<table-layer-spatial-reference-id>”)。

解說： 概略表層的空間參照系統和基礎表格層的空間參照系統必須相同。

使用者回應： 請為概略表層指定表格層的空間參照系統。

GSE1006E 因為 “<spatial-reference-id>” 不是有效的空間參照系統 ID，Spatial Extender 無法登記您所要求的層。

解說： 尚未啟用或登記您所指定的空間參照系統 (“<spatial-reference-id>”)。

使用者回應： 請啟用或登記空間參照系統。然後重新提出您的要求來登記此層。

GSE1007E Spatial Extender 嘗試新增空間直欄 (“<column-name>”) 到表格 “<schema-name.table-name>” 失敗，可能發生 SQL 錯誤 (SQLSTATE “<sqlstate>”)。

使用者回應： 查閱 SQLSTATE “<sqlstate>” 的相關訊息。

GSE1008E Spatial Extender 無法登記概略表層 “<layer-schema.layer-name.layer-column>”，因為概略表層的空間資料類型 “<layer-column-type>”，不符合基礎表格層 “<geo-schema.geo-name.geo-column>” 的空間資料類型 “<geo-column-type>”。

解說： 概略表層 “<layer-schema.layer-name.layer-column>” 的空間資料類型，必須符合此層之基礎表格層

“<geo-schema.geo-name.geo-column>”的空間資料類型。若這兩種資料類型不一致，則處理空間資料時將發生不明確情形。

使用者回應： 請確定概略表層及其基礎表格層的空間資料類型皆相同。

GSE1020E “<’spatial-reference-id>”是無效的空間參照系統 ID。

解說： 尚未啓用 “<’spatial-reference-id>” 識別字的空間參照系統。

使用者回應： 請確定已啓用指定的空間參照。

GSE1021E Spatial Extender 無法啓用空間參照系統 “<spatial-reference-id>”，因為相對應的空間座標系統 ID “<spatial-coordinate-id>”無效。

解說： 識別字 “<spatial-coordinate-id>” 的座標系統未定義於 Spatial Extender 型錄中。

使用者回應： 檢查 Spatial Extender 型錄中的 DB2GSE.COORD_REF_SYS 概略表，驗證座標系統識別字 “<spatial-coordinate-id>”。

GSE1030E 因為 “<schema-name.table-name>”不是基本表格，所以 Spatial Extender 無法啓用地理編碼程式。

解說： 包含您要地理編碼的來源資料的物件必須是一個基本表格。

使用者回應： 請確定包含您要地理編碼的來源資料的直欄是基本表格的一部份。

GSE1031E Spatial Extender 無法讓地理編碼程式 “<geocoder-id>” 在建立模式中自動為 “<layer-schema.layer-name.layer-column>” 這一層操作。

解說： 可能的原因如下：

- 已啓用地理編碼程式來自動更新層 “<layer-schema.layer-name.layer-column>”。
- 地理編碼程式對此層已暫時失效。
- 未對此層定義來源資料的任何直欄。

使用者回應： 若地理編碼程式暫時失效，請讓它以「重建」模式自動操作。

GSE1032E Spatial Extender 無法啓用地理編碼程式 “<geocoder-id>” 以重建模式為 “<layer-schema.layer-name.layer-column>” 這一層自動操作。

解說： 可能的原因如下：

- 已啓用地理編碼程式來自動更新層 “<layer-schema.layer-name.layer-column>”。
- 地理編碼程式先前未因此層而停用。
- 未對此層定義來源資料的任何直欄。

使用者回應： 若地理編碼程式先前在捨棄模式中停用，或尚未定義給此層，請使它能夠在「建立」模式自動操作。

GSE1033E 當 Spatial Extender 嘗試新增觸發函式到一個表格，且表格含有 “<layer-schema.layer-name.layer-column>” 這一層的直欄時，發生了 SQL 錯誤 (SQLSTATE “<sqlstate>”)。

解說： 觸發函式的目的在於維護屬性直欄 (地理編碼程式的輸入來源) 和空間直欄 (地理編碼程式的輸出目標) 之間的資料完整性。當 DB2 嘗試建立這些觸發函式時，發生 SQL 錯誤。

使用者回應： 查閱 SQLSTATE “<sqlstate>” 的相關訊息。

GSE1034E Spatial Extender 無法為 “<layer-schema.layer-name.layer-column>” 這一層，停用捨棄模式的地理編碼程式 “<geocoder-id>”。

解說： 可能的原因如下：

- 從未啓用地理編碼程式來自動更新 “<layer-schema.layer-name.layer-column>” 這一層。
- 地理編碼程式已在捨棄模式中停用。

使用者回應： 嘗試停用地理編碼程式之前，請先判斷其狀態。例如，是否已登記？是否已啓用？然後決定是否需要以捨棄模式停用。例如，若從未啓用，則完全不需要停用。

GSE1035E Spatial Extender 無法為 “<layer-schema.layer-name.layer-column>” 這一層，停用無效模式的地理編碼程式 “<geocoder-id>”。

解說： 可能的原因如下：

- 從未啓用地理編碼程式來自動更新 “<layer-schema.layer-name.layer-column>” 這一層。
- 地理編碼程式已在無效模式或捨棄模式中停用。

使用者回應： 嘗試停用地理編碼程式之前，請先判斷其狀態。例如，是否已登記？是否已啓用？然後決定是否需要以無效模式停用。例如，若已於無效模式中停用，則不需要再以此模式來停用。

GSE1036E 當 Spatial Extender 嘗試從一個表格中捨棄觸發函式，且表格含有 “<layer-schema.layer-name.layer-column>” 這一層的直欄時，發生了 SQL 錯誤 (SQLSTATE “<sqlstate>”)。

解說： 建立觸發函式的目的在於維護屬性直欄 (地理編碼程式的輸入來源) 和空間直欄 (地理編碼程式的輸出目標) 之間的資料完整性。當 DB2 嘗試捨棄這些觸發函式時，發生 SQL 錯誤。

使用者回應： 查閱 SQLSTATE “<sqlstate>” 的相關訊息。

GSE1037E Spatial Extender 無法為表格層 “<layer-schema.layer-name.layer-column>” 的來源資料進行地理編碼程式，可能因為指定不正確的值 “<number-of-attributes>” 給引數，此引數指定要為此層的來源資料提供多少屬性直欄。

解說： 不正確地指定此層相關的屬性直欄數，或不正確地指定這些直欄的名稱。

使用者回應： 請確定以正確的屬性直欄數和名稱來登記此層，或驗證地理編碼程式的輸入和輸出資料是否正確。

GSE1038E 當 Spatial Extender 嘗試以批次模式對表格層 “<layer-schema.layer-name.layer-column>” 的來源資料進行地理編碼時，發生了 SQL 錯誤 (SQLSTATE “<sqlstate>”)。

使用者回應：

- 查閱 SQLSTATE “<sqlstate>” 的相關訊息。
- 請確定此層的內容和 primaryUDF 引數已正確定義。

GSE1050E 您所指定的格線大小 (“<grid-size>”)，對於第一個格線層次而言無效。

解說： 您為第一個格線層次指定零或負數做為格線大小。

使用者回應： 請指定正數做為格線大小。

GSE1051E 您所指定的格線大小 (“<grid-size>”)，對於第二和第三個格線層次而言無效。

解說： 您為第二或第三個格線層次指定一個負數做為格線大小。

使用者回應： 請指定零或正數做為格線大小。

GSE1052E 當 **Spatial Extender** 嘗試為表格層 “<layer-schema.layer-name.layer-column>” 建立空間索引 “<index-schema.index-column>” 時，發生了 **SQL 錯誤 (SQLSTATE “<sqlstate>”)**。

使用者回應：

- 請確定已正確指定空間索引，且空間直欄無相關的索引。
 - 查閱 **SQLSTATE “<sqlstate>”** 的相關訊息。
-

GSE1500I 來源記錄 “<record-number>” 順利完成地理編碼。

解說： 含有屬性資料的記錄已順利完成地理編碼。

GSE1501W 來源記錄 “<record-number>” 未進行地理編碼。

解說： 精準度太高。

使用者回應： 請以較低的精準度來進行地理編碼。

GSE1502W 找不到來源記錄 “<record-number>”。

使用者回應： 判斷記錄是否存在於資料庫中。

GSE2001E **Spatial Extender** 無法執行您要求的作業。

使用者回應： 請洽詢您的資料庫管理員。

GSE2002E 發生資料庫管理系統錯誤。

使用者回應： 請洽詢您的資料庫管理員。

GSE2003E 您呼叫的儲存程序無法連接到您的工作站。

解說： 儲存程序無法存取辨識工作站的資訊。

使用者回應： 請洽詢您的資料庫管理員。

GSE2004E **Spatial Extender** 無法在您正在匯入的 **SDE** 傳送檔中驗證指定的座標系統 ID。

使用者回應： 請嘗試下列一或多種方法：

- 請確定 **SDE** 傳送檔中的空間參照系統 ID 是指向正確的座標系統 ID。
 - 請確定正確的座標系統 ID 是否列示在 **DB2GSE.COORD_REF_SYS** 型錄概略表中。如果 ID 不在此概略表中，請告知您的資料庫管理員。
 - 請判斷 **SDE** 傳送檔是否受損。如果受損，請試著取得並匯入其完整副本。
-

GSE2005E **Spatial Extender** 無法驗證您要匯出的檔案。

解說： 此問題的發生原因可能有一或多個。例如，您可能未獲授權存取檔案。或者，**Spatial Extender** 可能找不到或無法讀取它，或無法辨識其中的資料類型。

使用者回應： 請務必指定檔案的完整路徑。亦請確定您用來執行 **db2gse.gse_export_shape** 儲存程序的使用者 ID 對路徑中的每一個目錄都具有讀取及寫入存取權。請驗證含有這些目錄的磁碟已裝載在執行 **DB2** 的同一節點上，且使用的裝載點與路徑中所指定的相同。亦請驗證 **Spatial Extender** 是否可以辨識檔案所含的資料之資料類型。

如果錯誤再次發生，請試著判斷檔案是否受損。如果受損，請試著取得並匯出檔案的完整副本。

GSE2006E 檔案 “<filename>” 發生 I/O 錯誤。

使用者回應： 驗證檔案是否存在、對檔案是否具備適當的存取權限、確定其他使用者目前未使用該檔案。

GSE2007E **Spatial Extender** 無法驗證您要匯入資料的層。

解說： 定義此層的直欄名稱可能未正確地指定，或可能未遵循標準命名慣例。同樣地，直欄所屬的表格名稱可能未正確地指定，或可能未遵循標準命名慣例。

使用者回應： 請確定該層已列示在 DB2GSE.GEOMETRY_COLUMNS 型錄概略表中，直欄名稱及其所屬的表格已正確地指定，且這些名稱均遵循標準命名慣例。

GSE2008E **Spatial Extender** 嘗試將 NULL 插入有 NOT NULL 限制的層中。

使用者回應： 請將含有 NULL 的直欄匯入可以接受 NULL 的層，或要求資料庫管理員除去 NOT NULL 限制。

GSE2012E **Spatial Extender** 無法存取您要匯入資料的層。

解說： 您想用來存取層的使用者 ID 未獲授權變更定義層的直欄。

使用者回應： 請要求資料庫管理員授與您必要的權限（例如，您在直欄所屬的表格中可能需要 INSERT 或 SELECT 專用權）。

GSE2014E **Spatial Extender** 無法匯入資料至指定的層，或無法從其中匯出資料。

解說： Spatial Extender 找不到您要匯入資料或從其中匯出資料的層。

使用者回應： 請確定層是否列示在 DB2GSE.GEOMETRY_COLUMNS 概略表中。如果不是，請使用 db2gse.gse_register_layer 儲存程序或

「控制中心」的「建立層」視窗來登記層。如果層已列示在 DB2GSE.GEOMETRY_COLUMNS 中，請向您的資料庫管理員報告問題。

GSE2016E **Spatial Extender** 無法將您要求的形狀檔匯入您指定的層。

解說： 您要匯入的空間資料之資料類型與此空間資料指定的層之資料類型不相容。

使用者回應： 建立其資料類型與您要匯入的空間資料之資料類型相容的新層。然後，將資料匯入此新層。另外，匯入不同的形狀檔案，其空間資料與您要移入的層相容。

GSE2021E **Spatial Extender** 無法存取您要匯入的形狀檔案。

解說： 此問題的發生原因有數種可能。例如，Spatial Extender 可能不知道形狀檔案的完整路徑，或可能無法辨識檔案的格式，或可能含有該檔案的磁碟未正確地裝載。

使用者回應： 請務必指定檔案的完整路徑。如果您執行此動作且錯誤再次發生，則請驗證檔案是否確實是形狀檔，而非被誤指為形狀檔的其它種類的檔案。如果檔案確實是形狀檔，請嘗試下列其中一項恢復：

- 請判斷檔案是否受損。如果受損，請試著取得並匯入檔案的完整副本。
- 如果您是要從另一個工作站存取檔案，請確定：
 - 已裝載含有檔案的磁碟。
 - 此磁碟使用的裝載點與檔案路徑名稱中所指定的相同。
 - 您在另一個工作站上使用的使用者 ID 具有檔案的讀取權。

GSE2022E 指定的空間參照系統 ID 不存在。

解說： 為您要匯入的形狀檔所指定的空間參照系統 ID (SRID) 未列示在 Spatial Extender 型錄中。

使用者回應： 請採取下列一項動作：

- 對形狀檔指定一個相容的空間參照系統，且它的 SRID 已經記錄在 DB2GSE.SPATIAL_REF_SYS 型錄概略表中。
- 另外，建立形狀檔的新空間參照系統。然後，請驗證此系統的 SRID 是否已記錄在 Spatial Extender 型錄中。

此訊息不在 DB2 Spatial Extender 訊息型錄中。當發生相關錯誤時，會在線上通知您無法擷取。

GSE2023E Spatial Extender 無法從您指定的形狀檔中匯入屬性資料。

解說： 形狀檔中的屬性直欄定義無法轉換為您要匯入資料的表格中的對應直欄定義。

使用者回應： 請確定此屬性直欄的資料類型、最大長度及其它特性可以轉換為您要匯入資料的屬性直欄的相等設定或副本。

GSE2026E Spatial Extender 無法建立檔案以包含無法匯入的資料。

解說： 當您匯入形狀檔時，Spatial Extender 會在此檔案中收集無法匯入的所有記錄，以便日後能加以修正及匯入。在此情況下，Spatial Extender 沒有足夠的資訊或授權可以建立檔案以併入被拒絕的記錄。

使用者回應： 請針對 Spatial Extender 將要為被拒絕的記錄所建立的檔案，指定完整路徑。請確定具有相同路徑與名稱的檔案已經不存在。亦請確定您用來執行 db2gse.gse_import_shape 儲存程序的使用者 ID 對路徑中的每一個目錄都具有讀取及寫入權限。請驗證含有這些目錄的磁碟已裝載在執行 DB2 的同一節點上，且使用的裝載點與路徑中所指定的相同。

GSE2027E Spatial Extender 無法執行您要求的匯入或匯出作業。

解說： 沒有足夠的記憶體可以完成該作業。您要匯入或匯出的檔案可能已受損，造成排除過多的記憶體。

使用者回應： 請重試匯入或匯出檔案。如果錯誤持續發生，請試著判斷檔案是否受損。如果受損，請取得檔案的完整副本，並匯入或匯出此副本。如果問題持續發生，請向資料庫管理員報告該問題。

GSE2030E Spatial Extender 無法將資料匯入您指定的直欄。

解說： 您指定的直欄尚未登記為層。

使用者回應： 如果您要匯入 SDE 資料，請使用「DB2 控制中心」或 db2gse.gse_import_sde 儲存程序，將直欄登記為層並匯入資料。如果您要匯入形狀檔，請使用「控制中心」或 db2gse.gse_import_shape 儲存程序，將直欄登記為層並匯入資料。

GSE2031E Spatial Extender 無法將資料匯入您指定的層。

解說： 定義層的表格不再存在。

使用者回應： 如果您要匯入 SDE 資料，請使用「DB2 控制中心」或 db2gse.gse_import_sde 儲存程序，以重建表格並匯入資料。如果您要匯入形狀資料，請使用「控制中心」或 db2gse.gse_import_shape 儲存程序，以重建表格並匯入資料。

GSE2032E Spatial Extender 無法匯入或匯出屬性資料。

解說： 可能的解釋是您正試著將屬性資料匯入表格，但在匯入檔案中指定的一或多個屬性直欄在此表格中沒有副本。另一種解釋是您正試著從表格或概略表中匯出資料，但此表格或概略表中的一或多個屬性直欄在您要匯出資料的目的檔案中沒有副本。

使用者回應： 如果您要匯入屬性資料，請識別匯入檔案中的直欄（或每一個直欄），該檔案在要載入檔案的表格中沒有任何相對應的直欄。然後，將遺漏的直欄提供給表格。另外，您可以將匯入目標變更為層以及非原來指定的屬性直欄組。

如果您是要從表格或概略表中匯出屬性資料，請識別在匯出檔案中沒有任何相對應直欄的每一個表格

或概略表直欄。然後，以遺漏的直欄提供給匯出檔案，或提供新的匯出檔案 -- 該檔案具有您要匯出的每一個資料直欄的直欄。

GSE2033E Spatial Extender 無法讀取您要匯入的完整檔案。

解說： 檔案可能受損或被截斷。

使用者回應： 請重試匯入檔案。如果錯誤再次發生，請試著取得及匯入檔案的完整副本。

GSE2034E Spatial Extender 無法匯入您要求的 SDE 傳送檔案。

解說： 您要匯入的空間資料之資料類型與此空間資料指定的層之資料類型不相容。

使用者回應： 建立其資料類型與您要匯入的空間資料之資料類型相容的新層。然後，將資料匯入此新層。另外，匯入不同的 SDE 傳送檔案，其空間資料與您要移入的層相容。

空間函數傳回的訊息

由空間函數傳回的訊息 `SQLCODE` 是以字母 "GSE" 起首，其後接著從 3001 到 3042 範圍內的數字。

當空間函數傳回訊息時，亦會傳回其相關的 `SQLSTATE` 值，而非其 `SQLCODE`。如需了解如何尋找其 `SQLCODE` 的相關資訊，請參閱頁

給複查者的注意事項： 此處將插入交互參照。

GSE3001E 不明的系統失敗。

解說： 發生非預期的系統錯誤。

使用者回應： 請更正語法，然後重新呼叫函數。如果仍然發生該問題，請洽詢技術支援。

GSE3002E 無效的「知名文字」字串

解說： 您輸入無效的「知名文字」字串，故您呼叫的函數無效。

使用者回應： 請更正字串，並重新呼叫函數。若要判斷「知名文字」字串的格式是否正確，請參照「DB2 Spatial Extender 使用與參考手冊」。

GSE2035E 指定的座標超出界限。

解說： `Spatial Extender` 發現座標太大或太小，不在指定給指定層的座標系統界限內。如果此座標系統與層資料或與其相關的空間參照系統不相容，則會發生此問題。另外，問題的發生可能是由於形狀或 `SDE` 傳送檔案受損，或由於插入層的不規則資料錯誤。

使用者回應： 請驗證您是否對指定層使用的空間參照系統指定了正確的 ID。如果有，則此系統可能是與層資料或與其基礎座標系統不相容。基於這些可能性，請對層選取或建立不同的空間參照系統。如果問題重複發生，請向資料庫管理員報告該問題。

此訊息不在 `DB2 Spatial Extender` 訊息型錄中。當發生相關錯誤時，會在線上通知您無法擷取。

GSE3003E 無效的 SRID

解說： 您試圖傳送給此函數的空間參照系統 ID (`SRID`) 未列示在 `DB2 Spatial Extender` 系統型錄中。

使用者回應： 請指定目前記錄在 `DB2GSE.SPATIAL_REF_SYS` 型錄概略表中的 `SRID`，或建立具有您要指定的 `SRID` 的空間參照系統。

GSE3004E 記憶體不足

解說： 沒有足夠的記憶體可用。DB2 Spatial Extender 最多需要 1 MB 的記憶體上限。

使用者回應： 請重新配置記憶體，讓 DB2 Spatial Extender 有更多的記憶體可用。如果您無法重新配置記憶體，請新增更多的實體記憶體。

GSE3005E 幾何的 SRID 不同。

解說： 傳送給 DB2 Spatial Extender 函數的幾何必須共用相同的空間參照系統 ID (SRID)。

使用者回應： 請重建其中一項幾何，使其空間參照系統與其它的相符。

GSE3006E 無效的二進位字串。

解說： 在您呼叫的函數輸入中，輸入的「知名二進位」字串或「ESRI 二進位」字串之建構不正確。

使用者回應： 請以正確的格式重新建構字串。若要確定正確的格式，請參照 DB2 Spatial Extender Users' Guide and Reference。

GSE3007E 未指定有效的幾何。

解說： 未傳送幾何的有效類型到您呼叫的函數。有效的類型是幾何、點、線串、多邊形、多點、多線串及多重多邊形。

使用者回應： 請以有效的幾何類型重新提出 SQL 陳述式。

GSE3008E 括弧不對稱。

解說： 在「知名文字」代表字串中的左括弧數與右括弧數不相等。

使用者回應： 請重新輸入字串，並讓每一個左括弧都有相對應的右括弧。

GSE3009E 指定了太多組件。

解說： 在二進位或字串中所指出的組件數大於實際提供的組件數。

使用者回應： 請以正確的組件數輸入字串。

GSE3010E 幾何類型不正確。

解說： 傳送到您呼叫的函數之幾何類型錯誤。例如，可能傳送了線串到採用多邊形作為輸入的函數。

使用者回應： 請將函數可以處理的幾何類型傳送給函數，或使用能接受您要傳送的幾何類型的函數。

GSE3011E 字串太長。

解說： 幾何字串超出 4000 個字元的長度上限。

使用者回應： 幾何含有太多要轉換成文字的明細。但是，您可以將它轉換成 WKB 格式或 ESRI 形狀二進位格式。

GSE3012E 無效的參數值。

解說： 傳送給函數的參數無效。

使用者回應： 請與「DB2 Spatial Extender 使用與參考手冊」中所列出的函數比較函數語法。請更正無效的參數，然後重新提出函數。

GSE3013E 無效的格線大小。

解說： 下列其中一項的規格無效：

- 對第一、第二或第三格線層次所指定的格線大小是負數。
- 對第一格線層次所指定的格線大小為零。
- 指定給第二格線層次的格線大小小於第一格線層次的格線大小。
- 指定給第三格線層次的格線大小小於第二格線層次的格線大小。

使用者回應： 請使用「建立索引」視窗或 db2gse.gse_enable_idx 儲存程序來指定有效的格線大小。如需了解有效格線大小的相關資訊，請參照「DB2 Spatial Extender 使用與參考手冊」。

GSE3014E 格線太小。

解說： 已指定的格線大小造成在每個幾何中有 1000 個以上的格線資料格。

使用者回應： 請使用「建立索引」視窗或 db2gse.gse_enable_idx 儲存程序來增加格線大小或新增另一個格線層次。

GSE3015E 產生的幾何無效。

解說： 輸入的參數所產生的幾何無效。例如，以 LineFromShape 函數輸入的參數產生的幾何無效。無效的幾何即表示違反幾何內容。

使用者回應： 請更正參數，然後重新提出幾何。

GSE3016E 提交的幾何錯誤。

解說： 函數預期會有兩個某種類型的幾何，但沒有收到。例如，ST_Union 函數預期會有兩個相同維度的幾何並收到一個點及一個線串，但是其維度不同。

使用者回應： 請指定函數能接受的幾何作為有效輸入。若要判斷何種類型的幾何對此函數有效，請參照「DB2 Spatial Extender 使用與參考手冊」。

GSE3017E 幾何完整性錯誤。

解說： 由於幾何的內容違反了完整性限制，函數無法處理傳送給它的幾何。

使用者回應： 請重新提出內容定義正確的幾何。如需有關幾何內容的其餘相關資訊，請參照「DB2 Spatial Extender 使用與參考手冊」。

GSE3018E 太多點。

解說： 幾何的建構已超出 1MB 儲存體限制；幾何有太多點。

使用者回應： 請除去不必要的點。基於效能及儲存體考量，您應只併入那些必要的點以提交幾何。應排除所有不重要的點。

GSE3019E 幾何太小。

解說： 由 ST_Difference、ST_Intersection、ST_SymmetricDiff 或 ST_Union 函數傳回的幾何太小，無法代表現行座標系統的值。

使用者回應： 如果需要結果，請使用 db2gse.gse_enable_sref 儲存程序來增加來源幾何空間參照系統的 xyunits 參數。然後，重建儲存來源幾何的表格。

GSE3020E 緩衝區超出界限。

解說： 緩衝區函數建立的緩衝區在座標系統以外。

使用者回應： 請減少緩衝區距離，或變更來源幾何的座標系統。在大部份的情況中，變更座標系統需要重新載入空間系統。

GSE3021E 比例係數無效。

解說： 比例係數 (XY 單元, Z 單元或 M 單元) 不能小於 1。

使用者回應： 請使用 db2gse.gse_enable_sref 儲存程序，在 DB2GSE.SPATIAL_REF_SYS 型錄概略表中更正所有小於 1 的比例係數。

GSE3022E 座標超出界限。

解說： 座標太大或太小，不在座標系統的界限內。

使用者回應： 請確定座標是否正確。如果正確，請判斷它是否在您使用的座標系統界限內。如需此座標系統的相關資訊，請參閱 DB2GSE.COORD_REF_SYS 型錄概略表。

GSE3023E 座標系統 ID 無效。

解說： Spatial Extender 無法驗證指定的座標系統 ID。

使用者回應： 請確定 ID 是否列示在 DB2GSE.COORD_REF_SYS 型錄概略表中。如果不在概略表中，請驗證它是否正確，並要求資料庫管理員將它記錄在 Spatial Extender 系統型錄中。

GSE3024E 附註文字無效。

解說： 定義指定座標系統的附註文字無法轉換成有效的投射。

使用者回應： 請在 DB2GSE.COORD_REF_SYS 型錄概略表中查閱此座標系統的附註文字。請確定定義系統的文字是否正確。如需提供協助的相關資訊，請參閱「DB2 Spatial Extender 使用與參考手冊」中有關座標系統的章節。

GSE3025E 投射錯誤

解說： 在嘗試投射幾何時發生錯誤。

使用者回應： 請確定幾何是在投射的標準尺寸網域中。

GSE3026E 多邊形的環重疊。

解說： 多邊形的環不能重疊，但可以在正切上交集。

使用者回應： 請更正多邊形的座標，然後重新提出。

GSE3027E 太少點。

解說： 線串必須至少由兩個點所組成，且多邊形必須至少由四個點組成。

使用者回應： 請以正確的點數重新提出幾何。

GSE3028E 多邊形未封閉。

解說： 多邊形的起點及終點座標不相同。

使用者回應： 請編輯多邊形的座標清單，並確定起點與終點相同，然後重新提出。

GSE3029E 外部環無效。

解說： 外部環未含括內部環。內部環完全在外部環外且沒有重疊。

使用者回應： 請確定內部環的座標完全在外部環之內。如果內部環事實上是代表另一個多邊形的外部環，則輸入幾何作為多重重邊形。

GSE3030E 多邊形沒有任何區域。

解說： 幾何只有在在其座標在空間中展開成兩個維度時才是多邊形。

使用者回應： 請編輯多邊形的座標，使其形成一封閉區域，然後重新提出多邊形。或者，如果適當的話，請提交線串。

GSE3031E 多邊形包含 spike。

解說： 只有多邊形的終點及起點可以相同。多邊形環的所有其它座標必須不同，且集合封閉成一個區域。

使用者回應： 請尋找具有相同 X 及 Y 值的座標對組。請編輯這些點，使多邊形能封閉成一個單一區域，然後重新提出多邊形。

GSE3032E 外部環重疊。

解說： 多重重邊形的外部環可以在正切上交集，但不能重疊。

使用者回應： 請編輯外部環的座標，使它們不重疊，然後重新提出多重重邊形。

GSE3033E 多邊形本身交集。

解說： 多邊形的環不能自己交集。

使用者回應： 請編輯本身交集的環座標，然後重新提出多邊形。

GSE3034E 測量數無效。

解說： 二進位字串的測量數參數所含的測量數與字串提供的測量數不同。

使用者回應： 請編輯測量數參數，使它對應對二進位字串中提供的數。

GSE3035E 組件數無效。

解說： 二進位字串的組件數參數所指定的組件數與字串隨附的組件數不同。

使用者回應: 請編輯組件數參數，使它對應到二進位字串中所提供的數。

GSE3036E 組件偏移無效。

解說: 二進位字串的組件偏移參數所指定的組件偏移與字串中所提供的組件偏移不同。

使用者回應: 請編輯組件偏移參數，使它對應到二進位字串中所提供的組件偏移。

GSE3037E 投射錯誤。

解說: 發現非法幾何；其組件分隔字元無效。

使用者回應: 請電洽 IBM 客戶服務代表。

GSE3038E BLOB 太小。

解說: 在指定的二進位大型物件 (BLOB) 中的位元組數小於隨附的 BLOB 中的位元組數。

使用者回應: 請讓 BLOB 長度與 BLOB 中的位元組數相等，然後重新提出函數。

GSE3039E 無效的實體類型。

解說: 發現非法幾何；其相關的實體類型無效。

使用者回應: 請電洽 IBM 客戶服務代表。

當空間函數傳回訊息時，DB2 會顯示訊息的短表格及訊息 SQL0443N 中相對應的 SQLSTATE 值。下面顯示範例：

DB21034E 由於指令不是有效的「指令行處理器」指令，所以當成 SQL 陳述式處理。在 SQL 處理程序期間，傳回：SQL0443N 常式 "DB2GSE.ST_POINTFROMTEX" (特定名稱 "SQL000503150228187") 已傳回錯誤 SQLSTATE，其診斷文字為 "SRID 無效"。SQLSTATE=38601

若要找出哪一個 SQLCODE 與訊息 SQL0443N 中傳回的 SQLSTATE 相關聯，請參閱表34。若要察看與 SQLCODE 相關聯的全文，請參閱本章或發出下列指令：

DB2 ? [SQLCODE]

表 34. 空間函數傳回的訊息 SQLSTATE 值與 SQLCODE 值

如果 SQLSTATE 值是：	... 則 SQLCODE 是：
38600	GSE3002E
38601	GSE3003E

GSE3040E 無效的位元組次序。

解說: 位元組次序必須是 0 或 1。

使用者回應: 請編輯位元組次序，使 0 適用於 little endian 或 1 適用於 big endian。

GSE3041E 組件無效。

解說: 函數參數索引的組件不存在。例如，如果 ST_GeometryN 函數傳送 3 以在多點中傳回第三點，而當多點只含有兩點時即會發生此錯誤。

使用者回應: 請編輯參數，然後重新提出函數。

GSE3042E 空的幾何。

解說: 即使函數不接受，但傳送給 ST_AsBinary 函數作為輸入的是空幾何。

使用者回應: 請編輯您提交的 SQL 陳述式，使只有非空幾何會傳送給 ST_AsBinary 函數。例如，您可以使用 WHERE 子句來限定 ST_IsEmpty 函數不能有空幾何。

表 34. 空間函數傳回的訊息 *SQLSTATE* 值與 *SQLCODE* 值 (繼續)

如果 SQLSTATE 值是：	... 則 SQLCODE 是：
38602	GSE3004E
38603	GSE3005E
38604	GSE3006E
38605	GSE3007E
38606	GSE3008E
38607	GSE3009E
38608	GSE3010E
38609	GSE3011E
38610	GSE3012E
38612	GSE3013E
38613	GSE3014E
38800	GSE3015E
38801	GSE3016E
38802	GSE3017E
38803	GSE3018E
38804	GSE3019E
38805	GSE3020E
38806	GSE3021E
38807	GSE3022E
38808	GSE3023E
38809	GSE3024E
38810	GSE3025E
38811	GSE3026E
38812	GSE3027E
38813	GSE3028E
38814	GSE3029E
38815	GSE3030E
38816	GSE3031E
38817	GSE3032E
38818	GSE3033E
38819	GSE3034E
38820	GSE3035E

表 34. 空間函數傳回的訊息 *SQLSTATE* 值與 *SQLCODE* 值 (繼續)

如果 SQLSTATE 值是：	... 則 SQLCODE 是：
38821	GSE3036E
38822	GSE3037E
38823	GSE3038E
38824	GSE3039E
38825	GSE3040E
38826	GSE3041E
38827	GSE3042E
38999	GSE3043E

第11章 型錄概略表

Spatial Extender 的型錄概略表包含下列位置的 meta 資料：

- 您可使用的座標系統。關於這些系統識別字和附註文字的資訊，請參閱『DB2GSE.COORD_REF_SYS』。
- 已登記成爲層的空間直欄。關於這些直欄的名稱、資料類型及相關空間參照系統的資訊，請參閱第120頁的『DB2GSE.GEOMETRY_COLUMNS』。
- 您可使用的地理編碼程式。關於這些地理編碼程式識別字的資訊，以及地理編碼程式所處理的位置範圍，請參閱第120頁的『DB2GSE.SPATIAL_GEOCODER』。
- 您可使用的空間參照系統。關於識別字和說明的資訊，請參閱第121頁的『DB2GSE.SPATIAL_REF_SYS』。

DB2GSE.COORD_REF_SYS

當您啓用資料庫的空間作業功能時，Spatial Extender 會登記您可在型錄表格中使用的座標系統。自此表格中所選取的直欄，構成表35所述的DB2GSE.COORD_REF_SYS 型錄概略表。

表 35. DB2GSE.COORD_REF_SYS 型錄概略表中的直欄

名稱	資料類型	可為 NULL?	內容
CSID	INTEGER	否	此座標系統的唯一數值識別字。
CS_NAME	VARCHAR(64)	否	此座標系統的名稱。
AUTH_NAME	VARCHAR(256)	是	此座標系統所遵循的組織名稱；例如，European Petroleum Survey Group (EPSG)。
AUTH_SRID	INTEGER	是	由 AUTH_NAME 直欄中的組織指定給此座標系統的數值識別字。
DESC	VARCHAR(256)	是	此座標系統的說明。
SRTEXT	VARCHAR(2048)	否	此座標系統的附註文字。

DB2GSE.GEOMETRY_COLUMNS

當您建立一個層時，Spatial Extender 會在型錄表格中登記其識別字和相關資訊。自此表格中所選取的直欄，構成表36所述的 DB2GSE.GEOMETRY_COLUMNS 型錄概略表。

表 36. DB2GSE.GEOMETRY_COLUMNS 型錄概略表中的直欄

名稱	資料類型	可為 NULL?	內容
LAYER_CATALOG	VARCHAR(30)	是	NULL. Spatial Extender 中沒有 LAYER_CATALOG 觀念。
LAYER_SCHEMA	VARCHAR(30)	否	表格或概略表的綱目，此表格或概略表含有已登記成爲此層的直欄。
LAYER_TABLE	VARCHAR(128)	否	表格或概略表的名稱，此表格或概略表含有已登記成爲此層的直欄。
LAYER_COLUMN	VARCHAR(128)	否	已登記成爲此層的直欄名稱。
GEOMETRY_TYPE	INTEGER	否	已登記成爲此層的直欄資料類型。
SRID	INTEGER	否	空間參照系統的識別字，使用於已登記成爲此層的直欄值。
STORAGE_TYPE	INTEGER	是	關於 DB2 如何在已登記成爲此層的直欄中儲存值的資訊。例如，STORAGE_TYPE 中的資料可能指出值儲存成爲大型物件 (LOB)。

DB2GSE.SPATIAL_GEOCODER

可用的地理編碼程式登記在一個型錄表格中。自此表格中所選取的直欄，構成表37所述的 DB2GSE.SPATIAL_GEOCODER 型錄概略表。

表 37. DB2GSE.SPATIAL_GEOCODER 型錄概略表中的直欄

名稱	資料類型	可為 NULL?	內容
GCID	INTEGER	否	地理編碼程式的數值識別字。
GC_NAME	VARCHAR(64)	否	地理編碼程式的數值識別字。
VENDOR_NAME	VARCHAR(128)	否	提供地理編碼程式的供應商名稱。
PRIMARY_UDF	VARCHAR(256)	否	地理編碼程式的完整名稱。
PRECISION_LEVEL	INTEGER	否	來源資料與相對應的參照資料之間的相符程度，以利地理編碼程式順利處理。

表 37. DB2GSE.SPATIAL_GEOCODER 型錄概略表中的直欄 (繼續)

名稱	資料類型	可為 NULL?	內容
VENDOR_SPECIFIC	VARCHAR(256)	是	一個檔案的路徑和名稱，供應商可使用此檔案來設定地理編碼程式支援的任何特殊參數。
GEO_AREA	VARCHAR(256)	是	地理區，含有要地理編碼的位置。
DESCRIPTION	VARCHAR(256)	是	地理編碼程式的說明。

DB2GSE.SPATIAL_REF_SYS

當您建立一個空間參照系統時，Spatial Extender 會在型錄表格中登記其識別字和相關資訊。自此表格中所選取的直欄，構成表38所述的 DB2GSE.SPATIAL_REF_SYS 型錄概略表。

表 38. DB2GSE.SPATIAL_REF_SYS 型錄概略表中的直欄

名稱	資料類型	可為 NULL?	內容
SRID	INTEGER	否	此空間參照系統的使用者定義識別字。
SR_NAME	VARCHAR(64)	否	此空間參照系統的名稱。
CSID	INTEGER	否	此空間參照系統的基礎座標系統之數值識別字。
CS_NAME	VARCHAR(64)	否	此空間參照系統的基礎座標系統之名稱。
AUTH_NAME	VARCHAR(256)	是	為此空間參照系統設定標準的組織名稱。
AUTH_SRID	INTEGER	是	由 AUTH_NAME 直欄中的組織指定給此空間參照系統的識別字。
SRTEXT	VARCHAR(2048)	否	此空間參照系統的附註文字。
FALSEX	FLOAT	否	一個數字，從負數 X 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。
FALSEY	FLOAT	否	一個數字，從負數 Y 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。
XYUNITS	FLOAT	否	一個數字，乘以一個小數 X 座標或小數 Y 座標時，得到一個整數，可儲存成 32 位元的資料項目。
FALSEZ	FLOAT	否	一個數字，從負數 Z 座標值中扣除時，產生一個非負數 (亦即，一個正數或零)。
ZUNITS	FLOAT	否	一個數字，乘以一個小數 Z 座標時，得到一個整數，可儲存成 32 位元的資料項目。

表 38. DB2GSE.SPATIAL_REF_SYS 型錄概略表中的直欄 (繼續)

名稱	資料類型	可為 NULL?	內容
FALSEM	FLOAT	否	一個數字，從負數單位中扣除時，產生一個非負數 (亦即，一個正數或零)。
MUNITS	FLOAT	否	一個數字，乘以一個小數單位時，得到一個整數，可儲存成 32 位元的資料項目。

第12章 空間索引

因為空間直欄含有二維地理資料，所以查詢那些直欄的應用程式需要一個索引策略，能夠快速識別在某個延伸範圍內的所有幾何。由於這個原因，Spatial Extender 根據格線提供三層空間索引。

本章說明這種索引並提供其使用準則。涵蓋的主題有：

- 『範例程式片斷』
- 第124頁的『B 樹狀結構索引』
- 第124頁的『建立空間索引的方法』
- 第125頁的『如何產生空間索引』
- 第129頁的『使用空間索引的準則』

範例程式片斷

看看下例如何建立索引並用於 SQL。關於 CREATE INDEX 和 CREATE INDEX EXTENSION 指令的其它資訊，請參閱 *SQL 參考手冊*。請注意：建立索引之後，您可以發出使用空間函數和述詞的標準 DDL 和 DML 陳述式。

```
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.S
create index customersx1 on customers(loc) extend using db2gse.spatial_index(10e0,
100e0, 1000e0)
create index storesx1 on stores(loc) extend using db2gse.spatial_index(10e0, 100e0,
1000e0)
create index storesx2 on stores(zone) extend using db2gse.spatial_index(10e0, 100e0,
1000e0)

insert into customers (cid, addr, loc) values (:cid, :addr, sdeFromBinary(:loc))
insert into customers (cid, addr, loc) values (:cid, :addr, geocode(:addr))
insert into stores (sid, addr, loc) values (:sid, :addr, sdeFromBinary(:loc))
update stores set zone = db2gse.ST_Buffer (loc, 2)

select cid, loc from customers
  where db2gse.ST_Within(loc, :polygon) = 1

select cid, loc from customers
  where db2gse.ST_Within(loc, :circle1) = 1 OR
        db2gse.ST_Within(loc, :circle2) = 1

select c.cid, loc from customers c, stores s
  where db2gse.ST_Contains(s.zone, c.loc) = 1 selectivity 0.01
```

```
select avg(c.income) from customers c
where not exist (select * from stores s
                where db2gse.ST_Distance(c.loc, s.loc) < 10)
```

B 樹狀結構索引

空間索引技術是根據傳統階層式 B 樹狀結構索引，但差異頗大。空間索引使用格線索引，它專門索引二維空間直欄。B 樹狀結構索引只能處理一維資料而且無法與 GIS 資訊一起使用。本節說明如何建構和使用 B 樹狀結構索引。

B 樹狀結構索引的頂層 (稱為 root 節點) 含有下一層的每一個節點的一個索引鍵。每一個索引鍵值是下一層的對應節點的最大現存索引鍵值。根據基本表格中的值數目，可能需要一些中間節點。這些節點形成 root 節點與存放實際基本表格橫列 ID 的葉節點之間的橋站。

資料庫管理程式從 root 節點開始搜尋 B 樹狀結構索引。然後繼續搜尋中間節點直到具有基本表格的橫列 ID 的葉節點。

B 樹狀結構索引無法應用到空間直欄，因為空間直欄的二維特性需要空間索引結構。由於相同原因，所以您無法對非空間直欄應用空間索引。而且空間索引無法根據多重直欄來建立。

建立空間索引的方法

有一些方法可建立空間索引：

- 從「建立空間索引」視窗定義空間索引。關於說明，請參閱第55頁的『第6章 建立空間索引』。
- 在應用程式中呼叫 `db2gse.gse_enable_idx` 儲存程序。關於本儲存程序的資訊，請參閱第69頁的『第9章 儲存程序』。
- 在 USING 子句中，對 `spatial_index` 函數發出 `db2 create index` 指令。例如：

```
create index storesx1 on customers (loc) using db2gse.spatial_index(10e0,
100e0, 1000e0)
```

空間資料的本質需要資料庫設計者瞭解它的相對大小分佈。設計者必須決定用來建立空間索引的最佳格線層次大小和數目。

透過增加資料格大小輸入格線層次 (<格線層次 1>、<格線層次 2> 和<格線層次 3>)。因此第二層必須有資料格大小大於第一層，而第三層則大於第二層。第一個格線層次是必要層次，但您可使用倍準度 0 值 (0.0e0) 停用第二層和第三層。

如何產生空間索引

使用包封產生空間索引。包封是幾何本身而且代表幾何的最小和最大 X 與 Y 範圍。就大部份幾何而言，包封是一個方框，但對於水平和垂直線串來說，包封是兩點線串。就點而言，包封是點本身。關於包封的其它資訊，請參閱第136頁的『包封』。

透過為幾何包封與格線的交點製作一個或多個登錄，在空間直欄建構空間索引。交點記錄成幾何的內部 ID 以及相交的格線資料格的最小 X 和 Y 座標。例如，第126頁的圖7中的多邊形在座標 (20,30)、(30,30)、(40,30)、(20,40)、(30,40)、(40,40)、(20,50)、(30,50) 和 (40,50) 相交格線。關於第126頁的圖7中的幾何的最小 X 和 Y 座標，請參閱第126頁的表39。

若有多重格線層次，那麼 Spatial Extender 會使用可能的最低格線層次。一個幾何在給定的層次有相交的 4 個或更多個格線資料格時，它會升級至下一個更高層次。因此，有 3 個格線層次 10.0e0、100.0e0 和 1000.0e0 的空間索引，Spatial Extender 會先與層次 10.0e0 格線的幾何相交。若一個幾何與 4 個或更多個 10.0e0 格線資料格相交，它會升級而且與層次 100.0e0 格線相交。若在 100.0e0 層次產生 4 個或更多個交點，該幾何就會升級至 1000.0e0 層次。在 1000.0e0 層次上，必須將交點輸入此空間索引，因為這是最高的層次。

第126頁的圖7說明 4 種不同幾何如何與 10.0e 格線相交。在此空間索引記錄這 4 個幾何的全部 23 個交點。

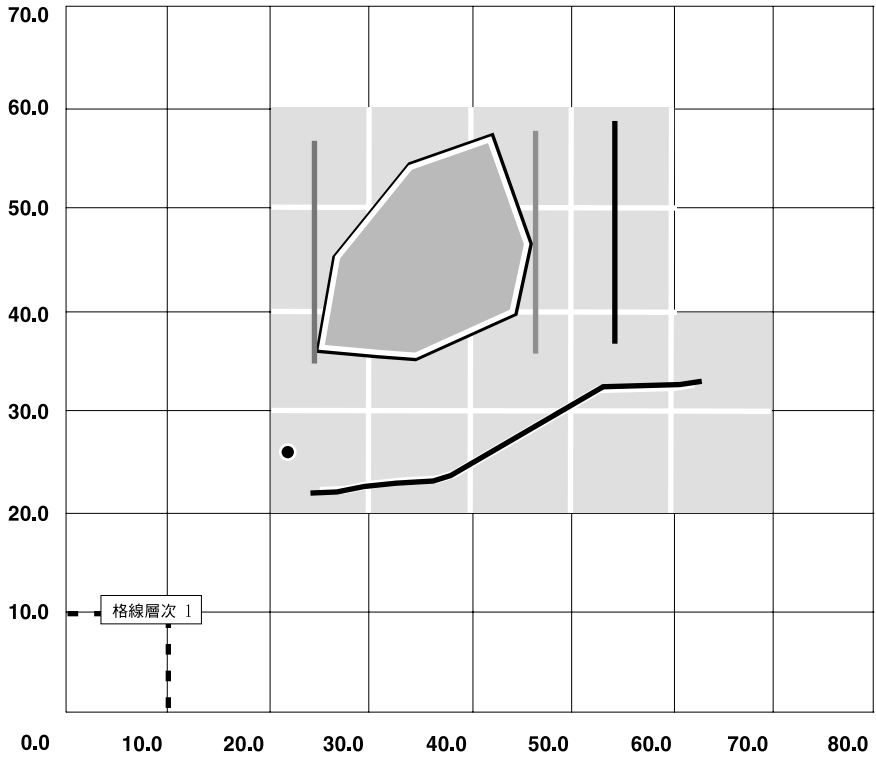


圖 7. 10.0e0 格線層次的應用程式

表39列示幾何與它們的對應格線交點。4 個不同幾何類型包封與 10.0e 格線相交。它相交的每一個格線資料格的最小 X 和 Y 座標會輸入到此空間索引。

表 39. 範例幾何的 10.0e0 格線資料格登錄

幾何	格線 X	格線 Y
多邊形	20.0	30.0
多邊形	30.0	30.0
多邊形	40.0	30.0
多邊形	20.0	40.0
多邊形	30.0	40.0
多邊形	40.0	40.0
多邊形	20.0	50.0
多邊形	30.0	50.0
多邊形	40.0	50.0
垂直線串	50.0	30.0

表 39. 範例幾何的 10.0e0 格線資料格登錄 (繼續)

幾何	格線 X	格線 Y
垂直線串	50.0	40.0
垂直線串	50.0	50.0
點	20.0	20.0
水平線串	20.0	20.0
水平線串	30.0	20.0
水平線串	40.0	20.0
水平線串	50.0	20.0
水平線串	60.0	20.0
水平線串	20.0	30.0
水平線串	30.0	30.0
水平線串	40.0	30.0
水平線串	50.0	30.0
水平線串	60.0	30.0

第128頁的圖8顯示如何透過新增格線層次 30.0e0 和 60.0e0，將交點數目大幅減至 8 個。本情況中，識別成幾何 1 的多邊形會升級至格線層次 30.0e0，識別成幾何 4 的線串會升級至格線層次 60.0e0。在 10.0e0 層次上的幾何原本有 9 個和 10 個交點，升級之後它們只有兩個交點。

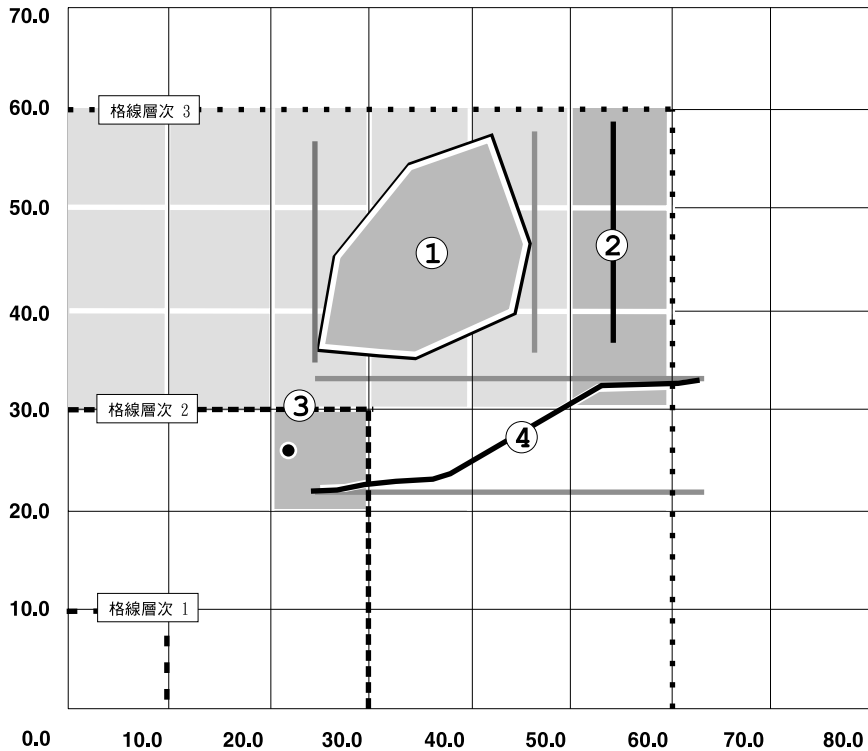


圖 8. 新增格線層次 30.0e0 和 60.0e0 的效果。識別成幾何 1 的多邊形包封與 9 個格線資料格相交。識別成幾何 2 的垂直線串包封與 3 個格線資料格相交。識別成幾何 3 的點包封只與一個格線資料格相交。識別成幾何 4 的線串包封與 10 個格線資料格相交。

Spatial Extender 選取在 CREATE INDEX 陳述式指定的格線層次參數，然後檢查每一個空間物件來決定此物件所在的格線區塊的座標和數目。在圖8中，格線層次 10.0e0、30.0e0 和 60.0e0 連同不斷增加的線加權以及不同灰階一起顯示。垂直線串和點包封資料格交點是輸入到 10.0e0 格線層次上的索引，因為這兩者產生的交點數目小於 4 個。多邊形與 9 個 10.0e0 格線資料格相交，因此升級至 30.0e0 格線層次。在本層次上，多邊形與兩個格線資料格相交，這兩個資料格輸入到此索引。識別成幾何 4 的線串與 10 個 10.0e0 格線資料格相交，因此升級至 30.0e0 格線層次。但是在本層次上，它與 6 個格線資料格相交，所以它再度升級至 60.0e0 格線層次，在該層次中它產生兩個交點。然後線串 60.0e0 格線交點輸入到此索引。若在本層次上線串產生 4 個或更多個交點，那麼這些交點仍會被輸入到此索引，因為這是可以升級幾何的最高層次。

表 40. 三層索引中的幾何交點

幾何	格線 X	格線 Y
層次 1 中的垂直線串與點之間的交點 (10.0e0 格線大小)		

表 40. 三層索引中的幾何交點 (繼續)

幾何	格線 X	格線 Y
2	50.0	30.0
2	50.0	40.0
2	50.0	50.0
3	20.0	20.0
層次 2 中的多邊形交點 (30.0e0 格線大小)		
1	0.0	30.0
1	30.0	30.0
層次 3 中的線串交點 (60.0e0 格線大小)		
4	0.0	0.0
4	60.0	0.0

Spatial Extender 不實際建立多邊形格線結構。Spatial Extender 透過在直欄空間參照系統的 X,Y 偏移上定義原點，以參數式表示格線層次。然後它會延伸格線到正座標空間。使用參數式格線時，Spatial Extender 會以數學方式產生交點。

使用空間索引的準則

Spatial Extender 使用空間索引增進空間查詢效能。想想最基本和最常見的空間查詢：方框查詢。本查詢要求 Spatial Extender 傳回一個使用者定義方框內的所有幾何。若沒有索引，Spatial Extender 就必須比較全部幾何與此方框。不過，有了一個索引之後，Spatial Extender 就可尋找索引登錄，找出其左下方座標大於或等於此方框座標及其右上方座標小於或等於此方框座標的索引登錄。因為索引是按座標系統排序，所以 Spatial Extender 能快速取得候選的幾何列示。剛才說明的處理稱為第一個流程。

第二個流程決定每一個候選的幾何包封是否與此方框相交。因為格線資料格包封與方框相交而通過第一個流程的幾何本身，可能有一個沒有相交的包封。

第三個流程比較候選的幾何的實際座標與方框，判斷方框內是否有此幾何的任何部份。前面兩個流程大幅減少總個體群後，對總個體群的一個子集所構成的候選幾何列示，執行這個最後且複雜的比較處理。

除了 EnvelopesIntersect 函數以外，全部空間查詢執行這三個流程。EnvelopesIntersect 函數只執行前面兩個流程。它的設計是專門為了顯示作業，這些作業經常使用自己的內建剪輯常式而且不需要第三個流程的細密處理。

選取格線資料格大小

不規則的幾何包封形狀使格線資料格大小的選擇變得複雜。因為不規則，所以有些幾何包封會與一些格線相交，有些包封則位於單一格線資料格內。根據資料的空間分配而定，相反地，有些格線資料格會與許多幾何包封相交。

爲了使空間索引順利運作，您一定要選取正確的格線數目和大小。試想一個含有一致大小的幾何的空間直欄。在此案例中，單一格線層次就足夠。從包含平均幾何包封的一個格線資料格大小開始。測試應用程式時，您會發現增加格線資料格大小會增進查詢效能。這是因爲每一個格線資料格含有一些幾何，而且第一個流程能夠更快捨棄不合格的幾何。不過，您會發現繼續增加資料格大小時，效能會開始降低。這是因爲第二個流程終究要與更多候選的幾何競爭。

選取層次數

若要索引的是關於相同相對大小的物件，那麼您可使用單一格線層次。雖然如此，但並非全部直欄都含有相同相對大小的幾何。通常一些空間直欄的幾何可分成一些大小間隔。例如，有一個道路網路，在該網路中幾何分割成街道、主要道路和公路。街道的長度相同而且可以集中在一個大小間隔。對於主要道路和公路來說，也是如此。因此，街道（代表一個大小間隔）可以分類到第一個格線層次，道路網路分類到第二個格線層次，主要公路分類到第三個格線層次。另一個範例則包括一個郡直欄，該直欄含有被較大鄉村地區包圍的小都心。在本案例中，有兩個大小間隔和兩個格線層次，一個代表較小都市地區，另一個代表較大鄉村地區。這些狀況很普遍而且需要使用多層次格線。

若要選取每一個格線層次的資料格大小，請選取那些稍大於每一個大小間隔的格線資料格大小。透過查詢空間直欄測試索引。

每一個附加的層次需要額外索引掃描。嘗試稍微向上或向下調整格線大小以判斷是否能大幅增進效能。

第13章 幾何與相關的空間函數

本章討論稱為幾何的資訊單元，由座標所組成，象徵符號式地理特徵。本章亦簡介採用幾何作為輸入及傳回結果的空間函數，協助您分析地理特徵及在地理資訊系統之間移動空間資料。涵蓋的主題有：

- 幾何的本質
- 幾何的內容；可傳回與這些內容相關之資訊的函數
- 可引證的幾何；可對幾何操作的函數
- 下列函數：
 - 顯示地理特徵之間的關係與比較
 - 產生幾何
 - 將幾何值轉換成可匯入及可匯出格式

關於幾何

Oxford American Dictionary 將幾何 定義為『涉及線條、角度、表面及實心內容與關係之數學分支。』1997 年 8 月 11 日，Open GIS Consortium Inc. (OGC) 在其出版品 *Open GIS Features for ODBC (SQL) Implementation Specification* 中，曾創造出此術語的另一個定義。幾何被用來表示幾何圖徵，過去一千多年以來，製圖者都是用幾何來製作世界地圖。幾何的這個新含意所代表的抽象定義是：『象徵地表上一個圖徵的一個點或一些點的聚集。』

在 Spatial Extender 中，幾何的可操作性定義可以是『地理特徵的模型。』此模型可以用圖徵的座標來表示，在某些情況下，也可以用視覺化符號來表示。此模型傳達一些資訊；例如，座標可識別相對於參照定點的圖徵位置，以及描繪其外形的符號。此外，此模型可用來產生資訊；例如，ST_Overlaps 函數可使用兩個最接近區域的座標作為輸入，並傳回區域是否重疊的資訊。

幾何所象徵的圖徵之座標，被視為幾何的內容。有些幾何還有其它內容，例如：

- 內部代表幾何象徵的圖徵內容。
- 外部代表圖徵四周的空間。
- 界限代表內容結束與環繞空間開始的界限。

這些和其它的內容均描述於第133頁的『內容與相關函數』中。

Spatial Extender 支援的幾何形成一個階層，如圖9所示。階層的六個成員為可引證的；它們可用視覺符號表示，本圖中也有顯示。

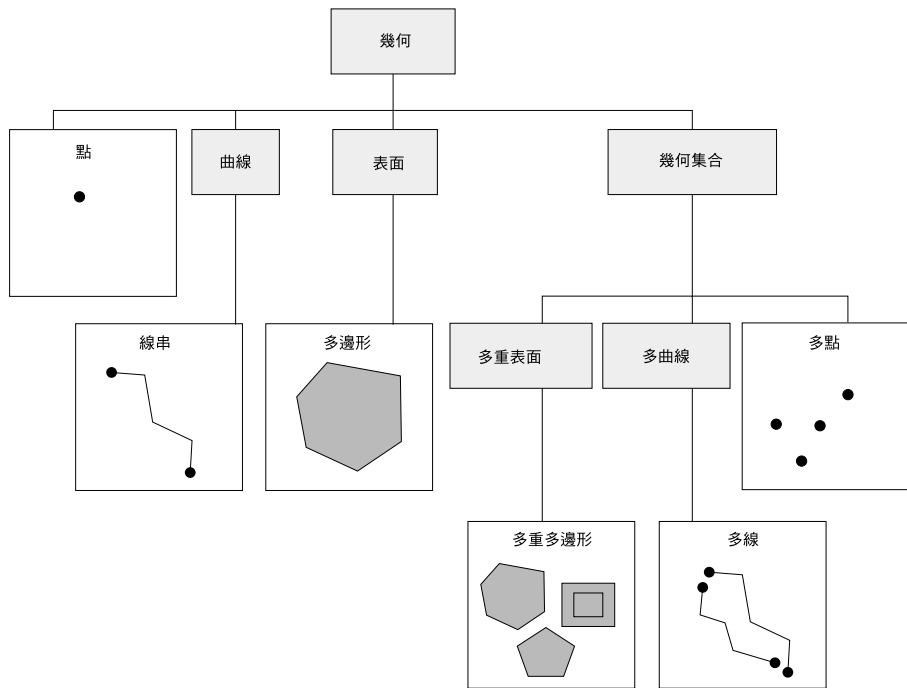


圖 9. Spatial Extender 支援的幾何階層. 可引證的幾何可用視覺符號表示。這些符號顯示於這些幾何名稱的下方。

如圖9所示，稱為幾何的超類別為階層的 Root。次類型分成兩個種類：基本幾何次類型和同類集合次類型。基本幾何包括：

- 點，象徵不連續圖徵，看起來佔據了東西座標線 (例如平行線) 與南北座標線 (例如子午線) 交集的位置。例如，假設某大型比例圖上的記號，顯示地圖上每一個城市位於一條平行線及一條子午線的交集。在這個比例圖上，每一個城市可以用一點來表示。
- 線串，象徵線性地理特徵 (例如，街道、溝渠及輸油管)。
- 多邊形，象徵多邊的地理特徵 (例如，福利區、森林及野生動植物產地)。

同類集合包括：

- 多點，象徵元件分別位於東西座標線與南北座標線交集點的多部份圖徵 (例如，成員分別位於平行線與子午線交集點的群島)。
- 多線串，象徵由長度單位或元件組成的多重圖徵 (例如，河川系統及公路系統)。

- 多重多邊形，象徵由多邊單元或元件組成的多部份圖徵 (例如，特定區域中的集體農地或湖泊系統)。

顧名思義，同類集合即是基本幾何的集合。除共用基本幾何內容之外，同類集合還有一些屬於自己的內容。

Spatial Extender 支援的空間資料類型皆為幾何的施行，如第132頁的圖9所示。有關這些資料類型的說明，請參閱第39頁的『關於空間資料類型』。

內容與相關函數

本節說明幾何的內容以及與這些內容相關的空間函數。這些內容如下：

- 幾何屬於哪一種類別
- 座標和測量
- 幾何的內部、界限及外部
- Z 座標
- 測量
- 單純或不單純的品質
- 空白或非空白的品質
- 幾何的包封
- 維度
- 幾何相關的空間參照系統之識別字

類別

每一個幾何皆屬於階層中的一個類別，如第132頁的圖9中所示。如第131頁的『關於幾何』中指示，階層中的六個次類型--點、線串、多邊形、多點、多線串及多重多邊形--皆為可引證的。超類別與其它次類型則不是可引證的。

| ST_GeometryType 函數會使用幾何，然後傳回可引證的次類型之字串 ID。關於詳細資訊，請參閱第223頁的『ST_GeometryType』。

| ST_IsValid() 函數會接收 ST_Geometry 值作為輸入參數。如果幾何有效，此函數將傳回 1 (TRUE)，如果幾何無效，則此函數將傳回 0 (FALSE)。關於詳細資訊，請參閱第243頁的『ST_IsValid』。

座標和測量

所有幾何至少都含有一個 X 座標及一個 Y 座標。此外，一個幾何學可以含有一或多個 Z 座標及測量。下列子區段說明：

- X 與 Y 座標
- Z 座標和測量
- ST_CoordDim 函數

X 與 Y 座標

X 座標值表示一個位置，相對於東西參照點。Y 座標值表示一個位置，相對於南北參照點。進一步資訊，請參閱第5頁的『空間資料的性質』和第33頁的『關於座標和空間參照系統』。

Z 座標和測量

本節說明 Z 座標、測量及在 DB2 Spatial Extender 中使用這些的方法。

Z 座標

部份幾何具有相關的高度或深度。形成特性的幾何之每一個點可包括一個可選用的 Z 座標，它代表垂直於地球表面的高度或深度。

AsShape 函數可將幾何值轉換成 ESRI 形狀表示法。如果幾何值含有任何 Z 座標或測量，則會以形狀表示法來保留它們。

Is3d 述詞函數使用幾何，如果此函數具有 Z 座標，則將傳回 1，否則會傳回 0。關於詳細資訊，請參閱第173頁的『Is3d』。

測量

所謂測量是指傳達地理特徵相關資訊的一個值，它與定義圖徵位置的座標儲存在一起。例如，假設您在 GIS 中呈現運輸系統。如果您要應用程式處理表示線性距離或里程標示的值，您可以將這些值連同定義系統位置的座標一起儲存。測量是以倍準數儲存。

如何使用 Z 座標及測量

在 DB2 Spatial Extender 中，Z 座標及測量是用來將資訊轉換成應用程式，而不是用來定義位置。（例如，請參閱第 3 章「設置資源」中的次區段「座標系統、座標及測量」。）因此，當大部份的空間函數處理含有 Z 座標、測量或兩者的點時，函數會忽略這些值。但是，其它的空间函數確實會在 Z 座標及測量上作業：

- 您可以使用 Is3d、IsMeasured 及 ST_CoordDim 函數來找出幾何是否包含 Z 座標、測量或兩者。
- 您可以使用 M 及 Z 函數來找出點的測量及 Z 座標。

- IsMeasured 述詞使用幾何，如果述詞中含有測量則傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第174頁的『IsMeasured』。

ST_CoordDim 函數

ST_CoordDim 會傳回一個值表示幾何的座標類型，及幾何是否也含有任何測量。此值稱為座標維度。

ST_CoordDim 可以傳回座標維度 2、3 或 4：

- 如果 ST_CoordDim 的輸入是一個點，則回覆值 2 表示該點是由一個 X 座標及一個 Y 座標所組成。如果輸入是點以外的幾何類型，則 2 表示此幾何中的每一個點都是由 X 座標及一個 Y 座標組成。
- 如果 ST_CoordDim 的輸入是一個點，則回覆值 3 表示該點是由一個 X 座標、一個 Y 座標及 Z 座標或測量所組成。如果輸入是點以外的幾何類型，則 3 的值表示此幾何中的每一個點都是由 X 座標、Y 座標及 Z 座標或測量所組成。
- 如果 ST_CoordDim 的輸入是一個點，則回覆值 4 表示該點是由 X 座標、Y 座標、Z 座標及測量所組成。如果輸入是點以外的幾何類型，則 4 的值表示此幾何中的每一個點都是由 X 座標、Y 座標、Z 座標及測量所組成。

內部、界限及外部

所有幾何在空間中都佔有一個位置，由它們的內部、界限及外部所定義。幾何的外部指該幾何未佔用的所有空間。幾何的界限是作為其內部與外部之間的介面。內部指幾何所佔有的空間。次類型直接繼承內部與外部內容，而界限內容則各異。

ST_Boundary 函數使用幾何，並傳回一個代表來源幾何界限的幾何。關於詳細資訊，請參閱第197頁的『ST_Boundary』。

單純或不單純

幾何的次類型（線串、多點及多線串）有些單純，有些不單純。如果次類型遵守對它加諸的所有拓樸規則，則屬單純；否則屬不單純。如果線串沒有與其內部交集，則屬單純。如果多點沒有任何元素佔用相同的座標空間，則屬單純。如果多線串的元素內部都沒有與它本身的內部交集，則屬單純。

ST_IsSimple 述詞函數使用幾何，如果幾何單純，則將傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第242頁的『ST_IsSimple』。

空白或非空白

如果幾何不含任何點，則為空白。空白幾何的包封、界限、內部及外部皆為 NULL。空白幾何一定單純，而且可具有 Z 座標或測量。空白線串及多線串的长度為 0。空白多邊形及多重多邊形的區域為 0。

`ST_IsEmpty` 述詞函數使用幾何，如果幾何為空白，則將傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第239頁的『`ST_IsEmpty`』。

包封

幾何的包封是由最小及最大 (X、Y) 座標形成的界限幾何。除下列例外狀況之外，幾何的包封都是形成界限矩形：

- 點的包封即是該點本身，因為它的最小及最大 X 座標一樣，最小及最大 Y 座標也一樣。
- 水平線或垂直線的包封即來源線串界限 (端點) 所代表的線串。

`ST_Envelope` 函數使用幾何，並傳回一個代表其包封的界限幾何。關於詳細資訊，請參閱第217頁的『`ST_Envelope`』。

維度

一個幾何的維度可以是 0、1 或 2。這些維度列示如下：

- 0** 既無長度也無區域
- 1** 具有長度
- 2** 包含區域

點及多點次類型的維度皆為 0。點代表可用單一座標建立模型的維度圖徵，而多點次類型代表必須用不相連座標叢集建立模型的資料。

線串與多線串次類型的維度皆為 1。它們儲存公路段、支流系統以及其它屬於線性本質的圖徵。

多邊形及多重多邊形次類型的維度為 2。多邊形或多重多邊形資料類型可描繪四周圍住一個可定義區域 (例如森林、陸地及湖泊) 的圖徵。

維度很重要，它不僅是次類型的內容，而且還擔任決定兩圖徵空間關係的角色。產生的圖徵的維度決定作業是否成功。`Spatial Extender` 可找出圖徵的維度，作為如何比較圖徵的一個判斷。

`ST_Dimension` 函數使用幾何，並以整數傳回其維度。關於詳細資訊，請參閱第211頁的『`ST_Dimension`』。

空間參照系統識別字

空間參照系統可識別每一個幾何的座標轉換。

資料庫已知的所有空間參照系統，皆可透過 `DB2GSE.SPATIAL_REF_SYS` 型錄概略表來存取。關於此概略表的資訊，請參閱第121頁的『`DB2GSE.SPATIAL_REF_SYS`』。

`ST_SRID` 函數使用幾何，並以整數傳回其空間參照識別字。關於詳細資訊，請參閱第277頁的『`ST_SRID`』。

`ST_Transform` 函數可將幾何指定給一個空間參照系統，此空間參照系統不同於幾何目前指定的空間參照系統。關於詳細資訊，請參閱第282頁的『`ST_Transform`』。

可引證的幾何與相關函數

本節簡介可引證的幾何的六個次類型，同時說明它們所使用的函數。這些次類型如下：

- 點
- 線串
- 多邊形
- 多點
- 多線串
- 多重多邊形

關於這些次類型所屬的階層以及與它們相關的視覺符號圖例，請參閱第132頁的圖9。

Points

點代表一個維度 0 的幾何，它佔用座標空間中單一位置。點包括定義此位置的 X 座標和 Y 座標。也包括 Z 座標和測量。

點很單純且有一個 NULL 界限。通常，點用來定義像油井、地標及高地的圖徵。

只對點次類型操作的函數如下：

PointFromShape

使用點類型的形狀與空間參照系統識別字，並傳回點。關於詳細資訊，請參閱第187頁的『`PointFromShape`』。

ST_Point

使用 X 座標、其相關的 Y 座標，以及這些座標所屬的空間參照系統識別字，並傳回座標所定義的點。關於詳細資訊，請參閱第265頁的『ST_Point』。

ST_PointFromText

使用點的 OGC 知名文字 (WKT) 表示法並傳回該點。關於詳細資訊，請參閱第266頁的『ST_PointFromText』。

ST_PointFromWKB

使用多邊形類型的知名二進位表示法和空間參照系統識別字，並傳回一個多邊形。關於詳細資訊，請參閱第267頁的『ST_PointFromWKB』。

ST_X 傳回 ST_Point 資料類型的 X 座標值為倍準數。關於詳細資訊，請參閱第289頁的『ST_X』。

ST_Y 傳回 ST_Point 資料類型的 Y 座標值為倍準數。關於詳細資訊，請參閱第290頁的『ST_Y』。

Z 傳回 ST_Point 資料類型的 Z 座標值為倍準數。關於詳細資訊，請參閱第291頁的『Z』。

M 傳回 ST_Point 資料類型的測量為倍準數。關於詳細資訊，請參閱第181頁的『M』。

線串

線串是儲存為一連串點之一維物件，定義線性內插路徑。如果線串沒有與內部交集，則屬單純。封閉線的端點 (界限) 佔用空間中相同的點。如果線串被封閉且其內部沒有與其本身交集，則是一個環。除了繼承超類別幾何的其它內容之外，線串還有長度。通常，線是用來定義線性圖徵，例如公路、河流及電線。

起點與終點相同的單純線串稱為環。

除非在界限為 NULL 的情況下線串被封閉，否則通常一些端點會形成線串的界限。除非在內部為連續的情況下線串被封閉，否則線串的內部是端點之間的連接路徑。

線串所使用的函數：

ST_StartPoint

使用線串並傳回其第一點。關於詳細資訊，請參閱第278頁的『ST_StartPoint』。

ST_EndPoint

使用線串並傳回其最後一點。關於詳細資訊，請參閱第216頁的『ST_Endpoint』。

ST_PointN

使用線串及第 n 點的指標並傳回該點。關於詳細資訊，請參閱第269頁的『ST_PointN』。

ST_Length

使用線串並以倍準數傳回其長度。關於詳細資訊，請參閱第245頁的『ST_Length』。

ST_NumPoints

使用線串並以整數依序傳回點數。關於詳細資訊，請參閱第260頁的『ST_NumPoints』。

ST_IsRing

使用線串，如果線為環，則傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第241頁的『ST_IsRing』。

ST_IsClosed

使用線串，如果線串被封閉，則傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第237頁的『ST_IsClosed』。

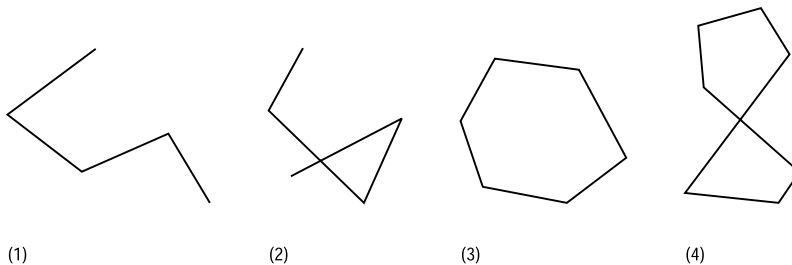


圖 10. 線串物件。

1. 單純非封閉線串。
2. 不單純非封閉線串。
3. 封閉單純線串，一個環。
4. 封閉非單純線串。不是一個環。

多邊形

多邊形為一種二維表面，它儲存一連串點，定義其外部界、限環及 0 或更多內部環。多邊形的環不能重疊。因此，依定義而言，多邊形一定是單純的。通常定義為陸地、湖泊及其它具有空間範圍的圖徵。

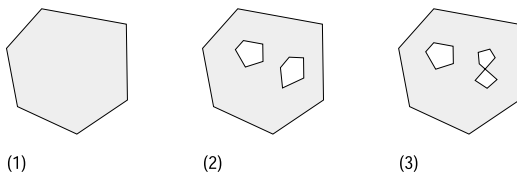


圖 11. 多邊形.

1. 界限由外部環定義的多邊形。
2. 界限由外部環與兩個內部環定義的多邊形。內部環內的區域為多邊形外部的一部份。
3. 因環在單一正切點上交集而形成的標準多邊形。

外部環及任何內部環定義多邊形的界限，而環之間括住的空間定義多邊形的內部。多邊形的環可交集在一個正切點上，但決不交叉。除了繼承自超類別幾何的其它內容之外，多邊形還有區域。

多邊形所使用的函數：

ST_Area

使用多邊形並以倍準數傳回其區域。關於詳細資訊，請參閱第193頁的『ST_Area』。

ST_ExteriorRing

使用多邊形並以線串傳回其外部環。關於詳細資訊，請參閱第220頁的『ST_ExteriorRing』。

ST_NumInteriorRing

使用多邊形並傳回其所包含的內部環數目。關於詳細資訊，請參閱第259頁的『ST_NumInteriorRing』。

ST_InteriorRingN

使用多邊形及指標並以線串傳回第 n 個內部環。關於詳細資訊，請參閱第229頁的『ST_InteriorRingN』。

ST_Centroid

使用多邊形並傳回一個點，即多邊形延伸範圍的中心。關於詳細資訊，請參閱第201頁的『ST_Centroid』。

ST_PointOnSurface

使用多邊形並傳回保證一定位於多邊形表面上的一個點。關於詳細資訊，請參閱第270頁的『ST_PointOnSurface』。

ST_Perimeter

使用多邊形並傳回其表面的周長。關於詳細資訊，請參閱第264頁的『ST_Perimeter』。

多點

多點是點的集合，如同其元素一般，它的維度是 0。如果多點沒有任何元素佔用相同的座標空間，則屬單純。多點的界限為 NULL。多點可用來定義現象，例如空中廣播型式及流行性傳染病爆發的事件。

多點所使用的函數：

ST_NumGeometries

使用同類集合並傳回其所包含的基本幾何元素數目。關於詳細資訊，請參閱第258頁的『ST_NumGeometries』。

ST_GeometryN

使用同類集合及索引，並傳回第 n 個基本幾何。關於詳細資訊，請參閱第222頁的『ST_GeometryN』。

多線串

多線串為許多線串的集合。如果多線串僅交集在線串元素端點上，則屬單純。如果線串元素內部有交集，則多線串不單純。

多線串的界限為線串元素的非交集端點。如果多線串的線串元素被封閉，則它也會被封閉。如果所有元素的端點有交集，則多線串的界限為 NULL。除了繼承自超類別幾何的其它內容之外，多線串還有長度。多線串可用來定義河川或公路網。

多線串所使用的函數：

ST_Length

使用多線串並以倍準數傳回其所有線串元素的累計長度。關於詳細資訊，請參閱第245頁的『ST_Length』。

ST_IsClosed

使用多線串，如果多線串被封閉，則將傳回 1 (TRUE)，否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第237頁的『ST_IsClosed』。

ST_NumGeometries

使用同類集合並傳回其所包含的基本幾何元素數目。關於詳細資訊，請參閱第258頁的『ST_NumGeometries』。

ST_GeometryN

使用同類集合及索引，並傳回第 n 個基本幾何。關於詳細資訊，請參閱第 222 頁的『ST_GeometryN』。

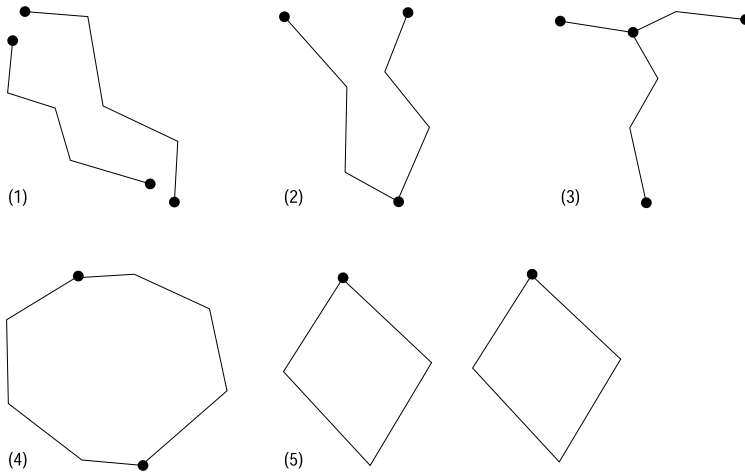


圖 12. 多線串。

1. 界限由其兩個線串元素的四個端點所定義之單純多線串。
2. 僅線串元素端點交集的單純多線串。其界限由兩個非交集端點所定義。
3. 其中一個線串元素的內部有交集的不單純多線串。此多線串的界限是由四個端點 (包括交集點) 所定義。
4. 單純非封閉多線串。它不會被封閉，因為它的元素線串不被封閉。因為其任何元素線串的內部沒有交集，所以屬單純。
5. 單純封閉多線串。它是封閉的，因為其所有元素都被封閉。因為其元素沒有交集在內部，所以屬單純。

多重多邊形

多重多邊形的界限為其元素的外部與內部環的累計長度。多重多邊形的內部被定義為其元素多邊形的累計內部。多重多邊形元素的界限僅能交集在一個正切點上。除了繼承自 `ST_MultiSurface` 類型的其它內容之外，多重多邊形還有區域。多重多邊形定義一些圖徵，例如森林層或非相連陸地 (像孤島系)。

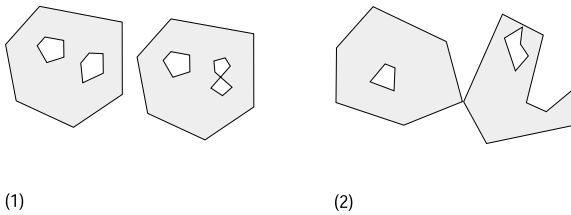


圖 13. 多重多邊形。

1. 具有兩個多邊形元素的多重重邊形。其界限是由兩個外部環與三個內部環所定義。
2. 具有兩個多邊形元素的多重重邊形。其界限是由兩個外部環與兩個內部環所定義。這兩個多邊形元素交集在一個正切點上。

多重多邊形所使用的函數：

ST_Area

使用多重多邊形並以倍準數傳回其多邊形元素的累計區域。關於詳細資訊，請參閱第193頁的『ST_Area』。

ST_Centroid

使用多重多邊形並傳回一個點，此點即它的幾何加權中心。關於詳細資訊，請參閱第201頁的『ST_Centroid』。

ST_NumGeometries

使用同類集合並傳回其所包含的基本幾何元素數目。關於詳細資訊，請參閱第258頁的『ST_NumGeometries』。

ST_GeometryN

使用同類集合及索引，並傳回第 n 個基本幾何。關於詳細資訊，請參閱第222頁的『ST_GeometryN』。

顯示關係與比較、產生幾何及轉換值的格式之函數

前面幾節討論三種空間函數：

- 與幾何內容相關的函數
- 與特定幾何相關的函數

本節將介紹更多的函數類別：

- 決定地理特徵關聯或比較方式的函數
- 產生新幾何的函數
- 將幾何值轉換成可匯入或匯出格式的函數

顯示地理特徵之間關係或比較的函數

數個函數傳回有關地理特徵彼此相關或彼此比較的方式之資訊。大部份函數會傳回一個整數值。本節說明常用的一些述詞函數，然後個別討論每一個函數。

述詞函數

如果某個比較符合函數的準則，則述詞函數將傳回 1 (TRUE)，若比較失敗，則會傳回 0 (FALSE)。測試空間關係的述詞比較幾何配對，這些幾何可以是不同的類型或維度。

述詞會比較已提交的幾何的 X 與 Y 座標。Z 座標與測量 (若有的話) 會被忽略。這會使具有 Z 座標或測量的幾何與不具有 Z 座標或測量的幾何做比較。

*Dimensionally Extended 9 Intersection Model (DE-9IM)*¹ 是一個數學方法，它定義具有不同類型和維度的幾何之間的配對空間關係。此模型把所有幾何類型之間的空間關係表達成其內部、界限和外部配對交集，同時把產生的交集之維度列入考量。

給定的幾何 a 和 b : $I(a)$ 、 $B(a)$ 和 $E(a)$ 分別代表 a 的內部、界限和外部。而 $I(b)$ 、 $B(b)$ 和 $E(b)$ 則代表 b 的內部、界限和外部。 $I(a)$ 、 $B(a)$ 和 $E(a)$ 與 $I(b)$ 、 $B(b)$ 和 $E(b)$ 的交集產生一個 3×3 矩陣。每一個交集可能產生不同維度的幾何。例如，兩個多邊形界限的交集是由一個點和一條線串組成，此時 dim 函數會傳回最大維度 1。

dim 函數傳回 -1、0、1 或 2 的值。1 對應至 NULL 設定或 $\text{dim}(\text{null})$ ，當找不到交集時會傳回它。

	內部	界限	外部
內部	$\text{dim}(I(a) \cap I(b))$	$\text{dim}(I(a) \cap B(b))$	$\text{dim}(I(a) \cap E(b))$
界限	$\text{dim}(B(a) \cap I(b))$	$\text{dim}(B(a) \cap B(b))$	$\text{dim}(B(a) \cap E(b))$
外部	$\text{dim}(E(a) \cap I(b))$	$\text{dim}(E(a) \cap B(b))$	$\text{dim}(E(a) \cap E(b))$

空間關係述詞的結果，可藉由比較述詞結果與代表 DE-9IM 可接受值的型樣矩陣來驗證或瞭解。

1. DE-91M 是由 Clementini 和 Felice 所開發出來的，他們在維度上延伸了 Egenhofer 和 Herring 的 9 Intersection Model。DE-91M 是由四位作者合作開發的，他們是：Clementini、Eliseo、Di Felice 和 van Osstrom。他們將此模型公佈於：“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. Springer-Verlag Singapore (1993) Egenhofer M.J. 和 Herring, J. 的 9 Intersection 模型公佈於 “Categorizing binary topological relationships between regions, lines, and points in geographic databases,” *Tech. Report, Department of Surveying Engineering, University of Maine, Orono, ME 1991*.

型樣矩陣包含每一個交集矩陣資料格的可接受值。可能的型樣值如下：

- T** 交集必須存在， $\text{dim} = 0、1$ 或 2 。
- F** 交集不可存在， $\text{dim} = -1$ 。
- *** 交集是否存在不重要， $\text{dim} = -1、0、1$ 或 2 。
- 0** 交集必須存在且其最大維度必須為 0 ， $\text{dim} = 0$ 。
- 1** 交集必須存在且其最大維度必須為 1 ， $\text{dim} = 1$ 。
- 2** 交集必須存在且其最大維度必須為 2 ， $\text{dim} = 2$ 。

例如，`ST_Within` 述詞的下列型樣矩陣包括值 `T`、`F` 及 `*`。

表 41. `ST_Within` 的矩陣. 用於幾何組合的 `ST_Within` 述詞之型樣矩陣。

		b		
		內部	界限	外部
a	內部	T	*	F
	界限	*	*	F
	外部	*	*	*

當兩個幾何的內部有交集，而且 `a` 的內部與界限沒有與 `b` 的外部交集時，`ST_Within` 述詞會傳回 `TRUE`。所有其它條件不重要。

每一個述詞至少有一個型樣矩陣，但某些述詞需要不止一個型樣矩陣來描述各種幾何類型組合的關係。

ST_Equals

如果兩個相同類型的幾何具有完全相同的 `X,Y` 座標值，則 `ST_Equals` 傳回 `1` (`TRUE`)。

點 / 點	多點 / 多點
線串 / 線串	多線 / 多線
多邊形 / 多邊形	多重多邊形 / 多重多邊形

圖 14. *ST_Equals*. 如果一些幾何具有符合的 X,Y 座標，則它們相等。

表 42. 相等矩陣. 相等的 DE-9IM 型樣矩陣可確保內部有交集，以及某幾何的內部或界限不會與另一幾何的外部交集。

		b		
		內部	界限	外部
a	內部	T	*	F
	界限	*	*	F
	外部	F	F	*

關於詳細資訊，請參閱第219頁的『*ST_Equals*』。

ST_OrderingEquals

ST_OrderingEquals 比較兩個幾何，如果幾何相等且座標順序相同，則將傳回 1 (TRUE)；否則會傳回 0 (FALSE)。關於詳細資訊，請參閱第261頁的『*ST_OrderingEquals*』。

ST_Disjoint

如果兩個幾何的交集是空集合，則 ST_Disjoint 傳回 1 (TRUE)。

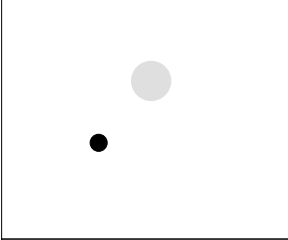
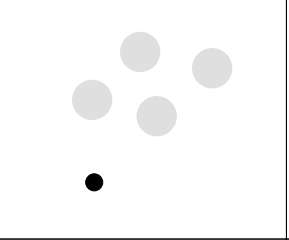
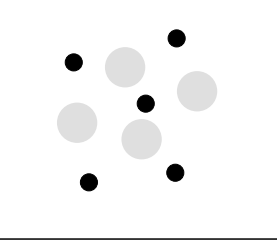
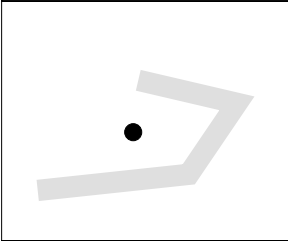
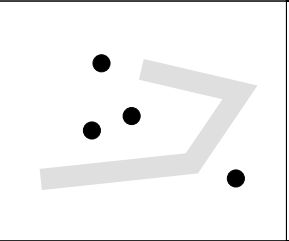
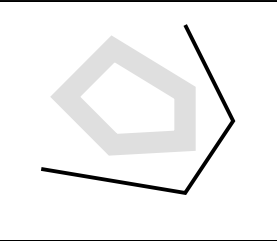
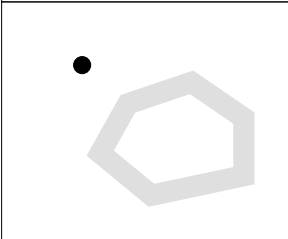
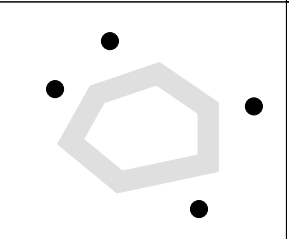
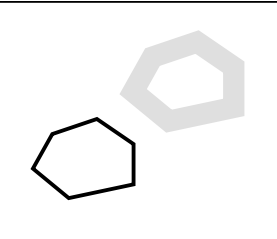
		
點 / 點	點 / 多點	多點 / 多點
		
點 / 線串	多線 / 線串	多邊形 / 線串
		
點 / 多邊形	多點 / 多重多邊形	多邊形 / 多邊形

圖 15. ST_Disjoint. 如果幾何在任何情況下彼此沒有交集，則它們是隔開的。

表 43. ST_Disjoint 的矩陣. ST_Disjoint 述詞的型樣矩陣簡單描述任一幾何的內部和界限均不交集。

		b		
		內部	界限	外部
a	內部	F	F	*
	界限	F	F	*
	外部	*	*	*

關於詳細資訊，請參閱第213頁的『ST_Disjoint』。

ST_Intersects

如果交集沒有導致空集合，則 ST_Intersects 傳回 1 (TRUE)。交集會傳回完全相反的 ST_Disjoint 結果。

如果下列任一型樣矩陣的條件傳回 TRUE，則 ST_Intersects 述詞傳回 TRUE。

表 44. ST_Intersects (1) 的矩陣. 如果兩個幾何的內部有交集，則 ST_Intersects 述詞傳回 TRUE。

		b		
		內部	界限	外部
a	內部	T	*	*
	界限	*	*	*
	外部	*	*	*

表 45. ST_Intersects (2) 的矩陣. 如果第一個幾何的界限與第二個幾何的界限有交集，則 ST_Intersects 述詞傳回 TRUE。

		b		
		內部	界限	外部
a	內部	*	T	*
	界限	*	*	*
	外部	*	*	*

表 46. ST_Intersects (3) 的矩陣. 如果第一個幾何的界限與第二個幾何的內部有交集，則 ST_Intersects 述詞傳回 TRUE。

		b		
		內部	界限	外部
a	內部	*	*	*
	界限	T	*	*
	外部	*	*	*

表 47. ST_Intersects (4) 的矩陣. 如果任一幾何的界限有交集，則 ST_Intersects 述詞傳回 TRUE。

		b		
		內部	界限	外部
a	內部	*	*	*
	界限	*	T	*
	外部	*	*	*

關於詳細資訊，請參閱第236頁的『ST_Intersects』。

EnvelopesIntersect

如果兩個幾何的包封有交集，此函數傳回 1 (TRUE)。它是有效執行 `ST_Intersects` 的便利函數 (`ST_Envelope(g1)`、`ST_Envelope(g2)`)。關於詳細資訊，請參閱第169頁的『`EnvelopesIntersect`』。

ST_Touches

如果兩個幾何共同的點沒有與其內部交集，則 `ST_Touches` 傳回 1 (TRUE)。至少一個幾何必須是線串、多邊形、多線串或多重多邊形。

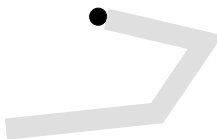
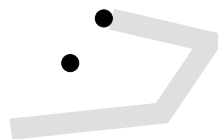
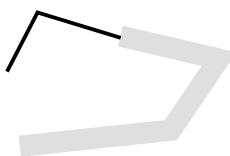
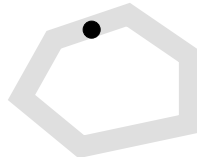
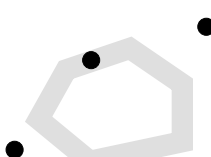
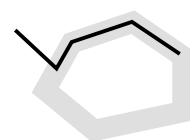
		
點 / 線串	多點 / 線串	線串 / 線串
		
點 / 多邊形	多點 / 多邊形	線串 / 多邊形

圖 16. `ST_Touches`

當幾何的內部沒有交集，但幾何的界限與另一幾何的內部或界限有交集時，此型樣矩陣顯示 `ST_Touches` 述詞傳回 TRUE。

表 48. `ST_Touches (1)` 的矩陣

	b		
	內部	界限	外部
a	內部 F	T	*
	界限 *	*	*
	外部 *	*	*

表 49. *ST_Touches* (2) 的矩陣

		b		
		內部	界限	外部
a	內部	F	*	*
	界限	T	*	*
	外部	*	*	*

表 50. *ST_Touches* (3) 的矩陣

		b		
		內部	界限	外部
a	內部	F	*	*
	界限	*	T	*
	外部	*	*	*

關於詳細資訊，請參閱第281頁的『*ST_Touches*』。

ST_Overlaps

ST_Overlaps 比較相同維度的兩個幾何。如果它們的交集導致某個幾何與它們不同，但有相同維度，則傳回 1 (TRUE)。

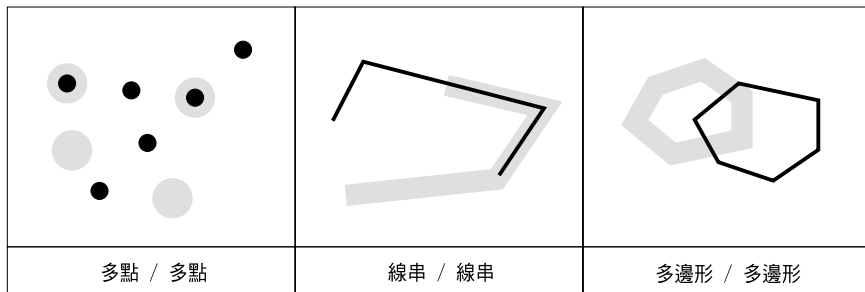


圖 17. *ST_Overlaps*

表51中的型樣矩陣適用於多邊形/多邊形、多點/多點，及多重多邊形/多重多邊形覆蓋。針對這些組合，如果兩個幾何的內部與其它幾何的內部和外部有交集，則此覆蓋述詞傳回 TRUE。

表 51. *ST_Overlaps* (1) 的矩陣

		b		
		內部	界限	外部
a	內部	T	*	T
	界限	*	*	*
	外部	T	*	*

表52中的型樣矩陣適用於線串/線串與多線串/多線串覆蓋。在此情形下，幾何的交集必須導致具有維度 1 的幾何 (另一線串)。如果內部交集的維度為 1，ST_Overlaps 述詞會傳回 FALSE，不過，ST_Crosses 述詞會傳回 TRUE。

表 52. ST_Overlaps (2) 的矩陣

		b		
		內部	界限	外部
a	內部	1	*	T
	界限	*	*	*
	外部	T	*	*

關於詳細資訊，請參閱第262頁的『ST_Overlaps』。

ST_Crosses

如果下列情況為真，則 ST_Crosses 會使用兩個幾何並傳回 1 (TRUE)：

- 交集造成幾何的維度小於來源幾何的最大維度。
- 已在兩個來源幾何的內部設定交集。

唯有進行多點/多邊形、多點/線串、線串/線串、線串/多邊形，及線串/多重多邊形比較時，ST_Crosses 才傳回 1 (TRUE)。

多點 / 線串	線串 / 線串
多點 / 多邊形	線串 / 多邊形

第152頁的表53中的型樣矩陣適用於多點/線串、多點/多線串、多點/多邊形、多點/多重多邊形、線串/多邊形及線串/多重多邊形。此矩陣描述內部必須交集，而且主要 (幾何 a) 的內部必須與次要 (幾何 b) 的外部交集。

表 53. *ST_Crosses (1)* 的矩陣

		b		
		內部	界限	外部
a	內部	T	*	T
	界限	*	*	*
	外部	*	*	*

表54中的型樣矩陣適用於線串/線串、線串/多線串及多線串/多線串。此矩陣描述內部交集的維度必須是 0 (交集在一點上)。如果此交集的維度是 1 (交集在一線串上)，則 *ST_Crosses* 述詞傳回 FALSE；不過，*ST_Overlaps* 述詞傳回 TRUE。

表 54. *ST_Crosses (2)* 的矩陣

		b		
		內部	界限	外部
a	內部	0	*	*
	界限	*	*	*
	外部	*	*	*

關於詳細資訊，請參閱第208頁的『*ST_Crosses*』。

ST_Within

如果第一個幾何完全在第二個幾何內，則 *ST_Within* 傳回 1 (TRUE)。 *ST_Within* 傳回 *ST_Contains* 的完全相反結果。

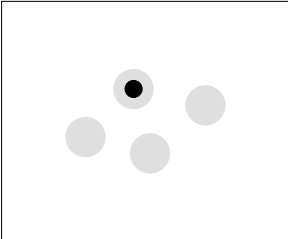
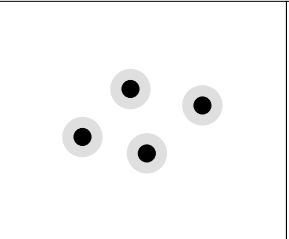
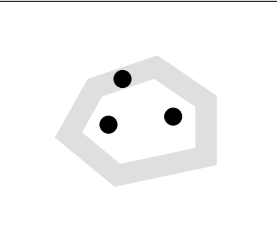
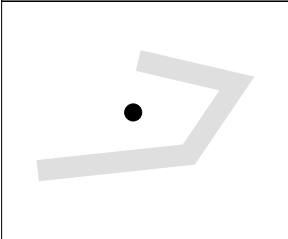
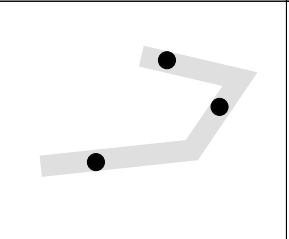
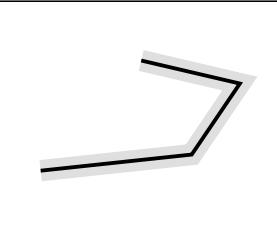
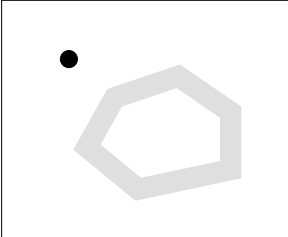
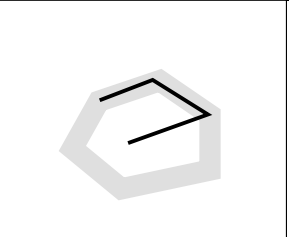
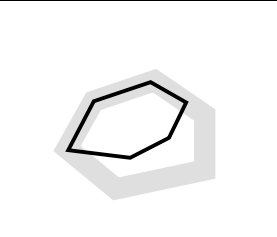
		
點 / 多點	多點 / 多點	多點 / 多邊形
		
點 / 線串	多點 / 線串	線串 / 線串
		
點 / 多邊形	線串 / 多邊形	多邊形 / 多邊形

圖 18. *ST_Within*

ST_Within 述詞型樣矩陣描述兩個幾何的內部必須交集，但主要幾何 (幾何 *a*) 的內部及界限，不可與次要幾何 (幾何 *b*) 的外部交集。

表 55. *ST_Within* 的矩陣

		b		
		內部	界限	外部
a	內部	T	*	F
	界限	*	*	F
	外部	*	*	*

關於詳細資訊，請參閱第285頁的『*ST_Within*』。

ST_Contains

如果第二個幾何完全包含在第一個幾何內，則 ST_Contains 傳回 1 (TRUE)。ST_Contains 述詞傳回與 ST_Within 述詞完全相反的結果。

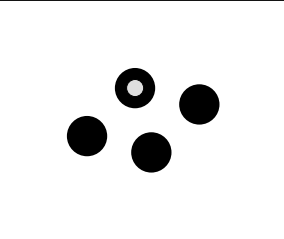
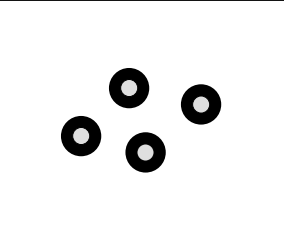
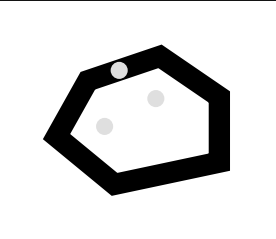
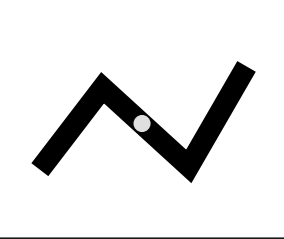
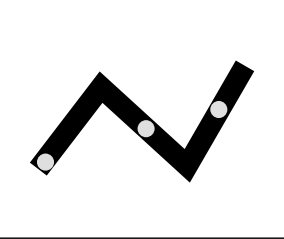
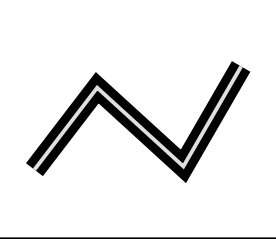
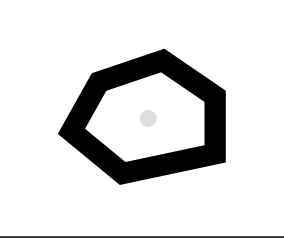
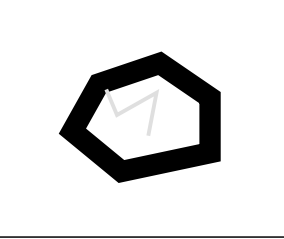
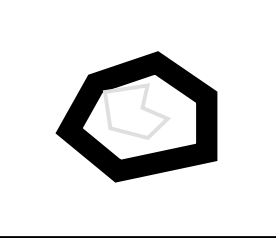
		
多點 / 點	多點 / 多點	多邊形 / 多點
		
線串 / 點	線串 / 多點	線串 / 線串
		
多邊形 / 點	多邊形 / 線串	多邊形 / 多邊形

圖 19. ST_Contains

ST_Contains 述詞的型樣矩陣描述兩個幾何的內部必須有交集，但次要幾何 (幾何 *b*) 的內部與界限，不可與主要幾何 (幾何 *a*) 的外部交集。

表 56. ST_Contains 的矩陣

		b		
		內部	界限	外部
a	內部	T	*	*
	界限	*	*	*
	外部	F	F	*

關於詳細資訊，請參閱第202頁的『ST_Contains』。

ST_Relate

ST_Relate 函數比較兩個幾何，如果幾何符合 DE-91M 型樣矩陣字串指定的條件，則傳回 1 (TRUE)；否則，此函數會傳回 0 (FALSE)。關於詳細資訊，請參閱第275頁的『ST_Relate』。

ST_Distance

ST_Distance 函數會報告兩個隔開圖徵相隔的最小距離。如果圖徵沒有隔開，此函數會報告最小距離 0。

例如，ST_Distance 報告兩位置之間飛機必須經過的最短距離。圖20會說明此資訊。

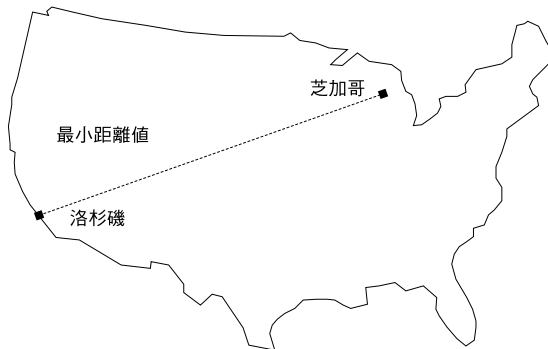


圖 20. 兩個都市間的最小距離。ST_Distance 可採用洛杉磯與芝加哥位置的座標為輸入，並傳回一值表示兩位置間的最小距離。

關於詳細資訊，請參閱第215頁的『ST_Distance』。

從現存的幾何產生新幾何的函數

Spatial Extender 提供從現存的幾何產生新幾何的述詞與轉換函數。

ST_Intersection

ST_Intersection 函數傳回兩個幾何的交集。此交集一定傳回為一個集合，即來源幾何的最小維度。以一條與多邊形交集的線串為例，此交集函數會傳回多邊形內部與界限共用的線串部份所構成的多線串。如果來源線串與含有兩個以上不連續區段的多邊形交集，則多線串包含不止一條線串。如果幾何沒有交集，或交集產生一個小於兩個來源幾何的維度，則傳回空白幾何。

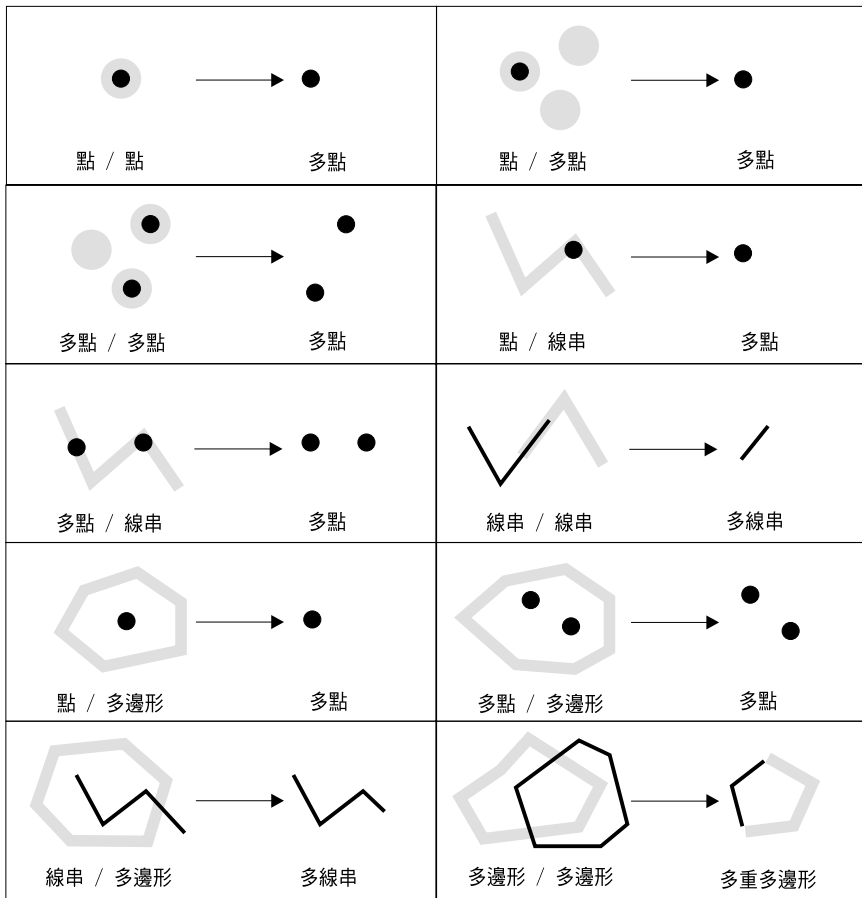


圖 21. *ST_Intersection*. *ST_Intersection* 函數的範例。

關於詳細資訊，請參閱第234頁的『*ST_Intersection*』。

ST_Difference

ST_Difference 使用兩個幾何作為輸入。第一個稱為主要幾何；第二個稱為次要幾何。*ST_Difference* 函數傳回未與次要幾何交集的主要幾何部份。此即所謂空間的邏輯 AND NOT。*ST_Difference* 函數僅對於類似維度的幾何操作，並傳回與來源幾何具有相同維度的集合。在來源幾何相等的情況下，會傳回空白幾何。如果提供給 *ST_Difference* 作為輸入的幾何維度不相符，則 *ST_Difference* 會傳回 NULL。

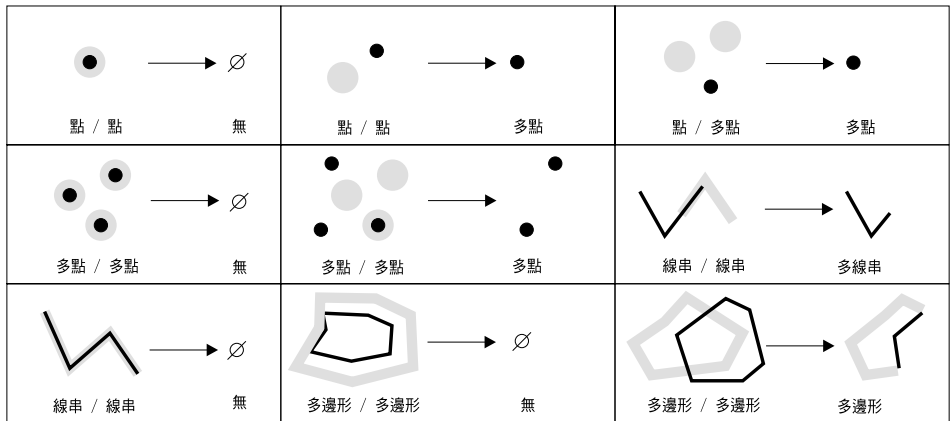


圖 22. *ST_Difference*

關於詳細資訊，請參閱第210頁的『*ST_Difference*』。

ST_Union

ST_Union 函數傳回兩個幾何的聯集。此即所謂空間的邏輯 OR。來源幾何必須具有類似維度。*ST_Union* 傳回的結果一定是一個集合。

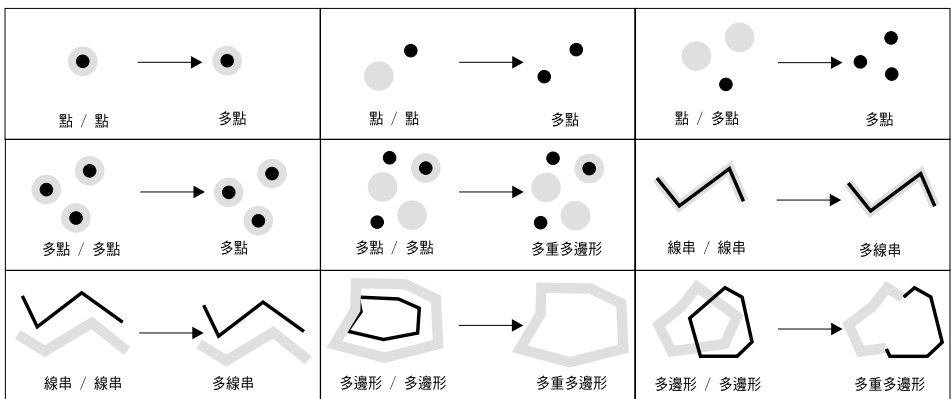


圖 23. *ST_Union*

關於詳細資訊，請參閱第284頁的『*ST_Union*』。

ST_SymmetricDiff

ST_SymmetricDiff 函數會傳回具有相同維度的兩個交集幾何間的對稱差異（空間的 Boolean 邏輯 XOR）。如果這些幾何是相等的，則 *ST_SymmetricDiff* 會傳回空幾何。如果它們不相等，則其中之一或兩者的一部份會超出交集區域。*ST_SymmetricDiff* 會以集合方式傳回非交集部份；例如，以多重多邊形。

如果已給定 `ST_SymmetricDiff` 不同維度的幾何作為輸入，則會傳回 `NULL`。

ST_Buffer

`ST_Buffer` 函數產生幾何的方式是以指定的距離包圍一個幾何。當主要幾何被放入緩衝區中，或某集合的元素很接近而足以讓所有緩衝區多邊形重疊時，便會產生一個多邊形。不過，當某個緩衝區集合的元素之間有足夠的距離時，便會產生個別緩衝區多邊形，在此情形下，`ST_Buffer` 函數會傳回一個多重多邊形。

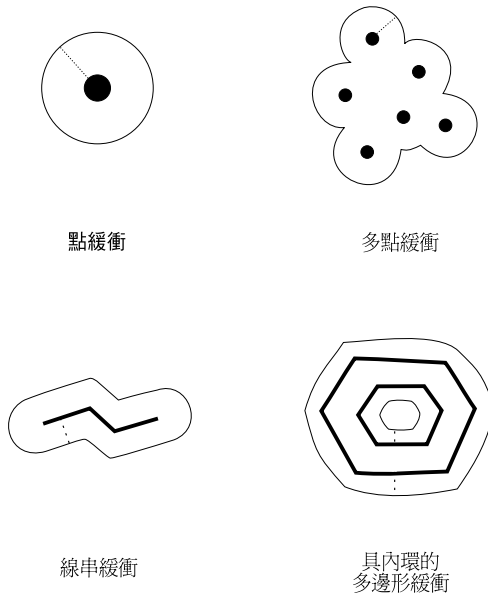


圖 24. `ST_Buffer`

`ST_Buffer` 函數接受正負距離，不過，只有二維的幾何 (多邊形與多重多邊形) 適用負緩衝區。每當來源幾何的維度小於 2 (多邊形或多重多邊形以外的所有幾何) 時，會使用緩衝區距離的絕對值。

一般而言，使用外部環時，正緩衝區距離會產生一些遠離來源幾何中心點的多邊形環；而負緩衝區會產生朝向中心點的多邊形或多重多邊形環。若是多邊形或多重多邊形的內部環，正緩衝區距離會產生朝向中心點的緩衝區環，而負緩衝區距離會產生遠離中心點的緩衝區環。

緩衝處理會合併一些重疊的多邊形。若負距離大於多邊形最大內部寬度的一半，則會產生空白幾何。

關於詳細資訊，請參閱第199頁的『`ST_Buffer`』。

LocateAlong

以具有測量的幾何而言，可利用 `LocateAlong` 函數找到特定測量的位置。`LocateAlong` 會將位置傳回為一個多點。如果來源幾何的維度是 0 (例如，一個點及一個多點)，則完全相符是必要的，而且會將那些具有相符測量值的點傳回為一個多點。不過，若來源幾何的維度大於 0，則位置會內插。例如，如果輸入的測量值是 5.5，而且測出的線串頂點為分別是 3、4、5、6 及 7，則會傳回剛好完全位於測量值 5 與 6 頂點之間的插入點。

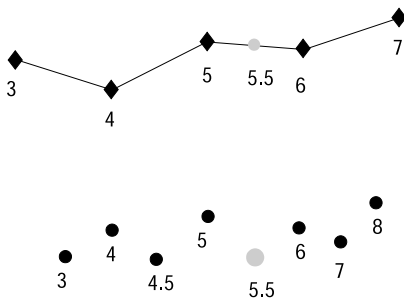


圖 25. `LocateAlong`

關於詳細資訊，請參閱第177頁的『`LocateAlong`』。

LocateBetween

`LocateBetween` 函數會從已測量的來源幾何傳回位於兩測量值之間的路徑或位置設定。如果來源幾何的維度是 0，則 `LocateBetween` 傳回一個多點，內含兩來源測量之間測得的所有點。若來源幾何的維度大於 0，如果可內插路徑，則 `LocateBetween` 傳回多線串；否則 `LocateBetween` 會傳回內含點位置的多點。若 `LocateBetween` 無法內插路徑或找出測量之間的位置，則傳回一個空白點。`LocateBetween` 會執行幾何的總括搜尋；因此，幾何的測量必須不小於從測量且不大於到測量。

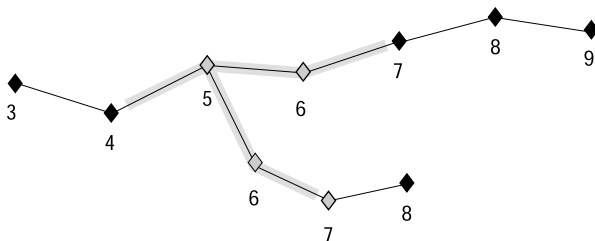


圖 26. `LocateBetween`

關於詳細資訊，請參閱第179頁的『`LocateBetween`』。

ST_ConvexHull

ST_ConvexHull 函數會傳回至少有三個頂點形成一個凸形的任何幾何之凸狀多邊形。如果幾何的頂點沒有形成一個凸形，則 ST_ConvexHull 傳回 NULL。在透過一組點建立 TIN 網路的格式排列中，ST_ConvexHull 通常是第一個步驟。

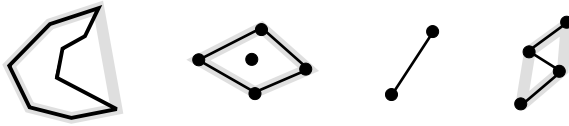


圖 27. ST_ConvexHull

關於詳細資訊，請參閱第204頁的『ST_ConvexHull』。

ST_Polygon

從線串產生多邊形。關於詳細資訊，請參閱第274頁的『ST_Polygon』。

轉換幾何值格式的函數

Spatial Extender 支援三個 GIS 資料交換格式：

- 知名文字表示法
- 知名二進位表示法
- ESRI 形狀表示法

知名文字表示法

Spatial Extender 具有從文字說明產生幾何的數個函數。

ST_WKTTToSQL

根據任何幾何類型的文字表示法來建立幾何。不必指定空間參照系統識別字。關於詳細資訊，請參閱第288頁的『ST_WKTTToSQL』。

ST_GeomFromText

根據任何幾何類型的文字表示法來建立幾何。必須指定空間參照系統識別字。關於詳細資訊，請參閱第225頁的『ST_GeomFromText』。

ST_PointFromText

根據點文字表示法來建立點。關於詳細資訊，請參閱第266頁的『ST_PointFromText』。

ST_LineFromText

根據線串文字表示法來建立線串。關於詳細資訊，請參閱第247頁的『ST_LineFromText』。

ST_PolyFromText

根據多邊形文字表示法來建立多邊形。關於詳細資訊，請參閱第271頁的『ST_PolyFromText』。

ST_MPointFromText

根據多點表示法來建立多點。關於詳細資訊，請參閱第253頁的『ST_MPointFromText』。

ST_MLineFromText

根據多線串表示法來建立多線串。關於詳細資訊，請參閱第250頁的『ST_MLineFromText』。

ST_MPolyFromText

根據多重多邊形表示法來建立多重多邊形。關於詳細資訊，請參閱第256頁的『ST_MPolyFromText』。

文字表示法是一種 ASCII 字串。它允許幾何以 ASCII 文字格式交換。這些函數不需要定義任何空間程式結構來對映二進位表示法。所以，可使用於 3GL 或 4GL 程式。

ST_AsText 函數可將現存的幾何值轉換成文字表示法。關於詳細資訊，請參閱第196頁的『ST_AsText』。

關於知名文字表示法的詳細說明，請參閱第303頁的『OGC 知名文字表示法』。

知名二進位表示法

Spatial Extender 具有從知名二進位 (WKB) 表示法產生幾何的數個函數。

ST_WKBTToSQL

根據任何幾何類型的知名二進位表示法來建立幾何。不必指定空間參照系統識別字。關於詳細資訊，請參閱第286頁的『ST_WKBTToSQL』。

ST_GeomFromWKB

根據任何幾何類型的知名二進位表示法來建立幾何。必須指定空間參照系統識別字。關於詳細資訊，請參閱第227頁的『ST_GeomFromWKB』。

ST_PointFromWKB

根據點的知名二進位表示法來建立點。關於詳細資訊，請參閱第267頁的『ST_PointFromWKB』。

ST_LineFromWKB

根據線串的知名二進位表示法來建立線串。關於詳細資訊，請參閱第248頁的『ST_LineFromWKB』。

ST_PolyFromWKB

根據多邊形的知名二進位表示法來建立多邊形。關於詳細資訊，請參閱第 272 頁的『ST_PolyFromWKB』。

ST_MPointFromWKB

根據多點的知名二進位表示法來建立多點。關於詳細資訊，請參閱第 254 頁的『ST_MPointFromWKB』。

ST_MLineFromWKB

根據多線串的知名二進位表示法來建立多線串。關於詳細資訊，請參閱第 251 頁的『ST_MLineFromWKB』。

ST_MPolyFromWKB

根據多重多邊形的知名二進位表示法來建立多重多邊形。關於詳細資訊，請參閱第 257 頁的『ST_MPolyFromWKB』。

知名二進位表示法為一連串的位元組。它允許幾何以二進位形式在 ODBC 從屬站與 SQL 資料庫之間交換。這些幾何函數需要定義 C 結構來對映二進位表示法。所以，適合使用於 3GL 程式，但不適合在 4GL 環境中使用。

ST_AsBinary 函數可將現存的幾何值轉換成知名二進位表示法。關於詳細資訊，請參閱第 195 頁的『ST_AsBinary』。

關於知名二進位表示法的詳細說明，請參閱第 308 頁的『OGC 知名二進位 (WKB) 表示法』。

ESRI 形狀表示法

Spatial Extender 具有從 ESRI 形狀表示法產生幾何的數個函數。除了文字與知名二進位表示法支援的二維表示法之外，ESRI 形狀表示法還支援 Z 座標與測量。

ShapeToSQL

根據任何幾何類型的形狀來建立幾何。不必指定空間參照系統識別字。關於詳細資訊，請參閱第 191 頁的『ShapeToSQL』。

GeometryFromShape

根據任何幾何類型的形狀來建立幾何。必須指定空間參照系統識別字。關於詳細資訊，請參閱第 171 頁的『GeometryFromShape』。

PointFromShape

根據點形狀來建立點。關於詳細資訊，請參閱第 187 頁的『PointFromShape』。

LineFromShape

根據折線形狀來建立線串。關於詳細資訊，請參閱第 175 頁的『LineFromShape』。

PolyFromShape

根據折線形狀來建立多邊形。關於詳細資訊，請參閱第189頁的『PolyFromShape』。

MPointFromShape

根據多點形狀來建立多點。關於詳細資訊，請參閱第184頁的『MPointFromShape』。

MLineFromShape

根據多重折線形狀來建立多線串。關於詳細資訊，請參閱第182頁的『MLine FromShape』。

MPolyFromShape

根據多重多邊形形狀來建立多重多邊形。關於詳細資訊，請參閱第186頁的『MPolyFromShape』。

這些函數的一般語法都相同。第一個引數是以 BLOB 資料類型輸入的形狀表示法。第二個引數是將指定給幾何的空間參照識別字。例如，GeometryFromShape 函數的語法如下：

```
GeometryFromShape(shapegeometry, SRID)
```

若要對映二進位表示法，則這些形狀函數需要定義 C 結構。所以，適合使用於 3GL 程式，但不適合在 4GL 環境中使用。

AsBinary 函數可將幾何值轉換成 ESRI 形狀表示法。關於詳細資訊，請參閱第168頁的『AsShape』。

關於形狀表示法的詳細說明，請參閱第311頁的『ESRI 形狀表示法』。

第14章 SQL 查詢的空間函數

本章列出可於查詢空間資料時呼叫的函數。每一區段說明一個函數，包含其語法、回覆類型及程式碼範例。本章的部份範例包括 CREATE TABLE 陳述式，其位於被定義為空間直欄的一個或多個直欄。

如第 4 章（「定義一些空間直欄、將它們登記為「層」，以及啓用地理編碼程式來維護它們」）中的「關於空間資料類型」所述，空間資料類型會形成一種階層並使用 ST_Geometry 作為 Root。當文字表示此階層中的次層類型值可以用來作為函數的輸入時，讀者應瞭解在此超層類型的次層中所有類型的值都可以用來作為輸入。

例如，在第 14 章，“Is3d” 一節中表示 Is3d 函數會採用 ST_Geometry 資料類型的值作為輸入。讀者應瞭解 Is3d 的輸入可以是在 ST_Geometry 的次層中所有支援資料類型的值：ST_Point、ST_Curve、ST_LineString 等等。如需所有支援的次層類型的圖表，請參閱使用與參考手冊中的圖 6。

將下列注意事項引用至空間函數：

- 本章的範例是以常式庫名稱 db2gse 定義。除明確地以 db2gse 定義每一個空間函數及類型外，您亦可以將函數路徑設定為包括 db2gse。
- 如果 ST_Union 使用具有相同座標的兩個點作為輸入，則會傳回一個點。然後，如果此點提供給 ST_IsSimple 作為輸入，則 ST_IsSimple 會傳回 1 (TRUE；亦即：是，點是簡式的)。
- 如果提供 NULL 給空間函數作為輸入參數，則會傳回 NULL 作為輸出參數。
- 在您可以將資料插入空間直欄之前：
 - 您可能需要增加 udf_mem_sz 參數。建議的起始設定值為 2048。如果 2048 無法勝任，請增加 udf_mem_sz 參數為 256 的增量。
 - 您必須將它登記為層。關於將空間直欄登記為層的詳細資訊，請參閱第39頁的『第4章 定義一些空間直欄、將它們登記為「層」，以及啓用地理編碼程式來維護它們』。

巢狀

如果內部巢狀函數傳回的幾何類型不被外部呼叫函數接受，則將巢狀幾何函數放在一起會出現問題。如果內部巢狀函數的幾何類型可以轉換成外部呼叫函數可以接受的幾何類型，則此問題即可解決。

將 ST_Geometry 轉換為次類型

可以使用 Treat 函數，將傳回超類型 ST_Geometry 的函數幾何類型轉換成次類型。例如，ST_Union 函數傳回 ST_Geometry 的值。當 ST_Union 是在 ST_PointOnSurface 函數中的巢狀時，ST_PointOnSurface 會傳回下列錯誤：

```
SQL00440N 在函數路徑中找不到  
任何具有相容引數的函數  
"ST_POINTONSURFACE" 。      SQLSTATE=42884
```

ST_PointOnSurface 函數會使用 ST_Polygon 或 ST_MultiPolygon 幾何類型，但不會使用由 ST_Union 傳回的 ST_Geometry，即使 ST_Union 函數所傳回的值是 ST_MultiPolygon。因此，在此情況下，有必要將 ST_Union 函數的幾何類型轉換為 ST_MultiPolygon。

例如，如果 COUNTIES 表格是利用聯集其 COUNTY 多邊形直欄來自我結合，必須將 Treat 函數套用於 ST_Union 函數的結果，以便在套用 ST_PointOnSurface 函數前，先將 ST_Geometry 類型轉換成 ST_MultiPolygon。

```
SELECT ST_Astext(ST_PointOnSurface(  
    TREAT ( ST_Union(c1.county, c2.county) AS ST_MultiPolygon)))  
FROM   counties AS c1, counties AS c2;
```

如果 ST_Union 函數傳回 ST_MultiPolygon 的值，則 Treat 函數會將它轉換成 ST_MultiPolygon 資料類型。如果 ST_Union 函數沒有傳回 ST_MultiPolygon 值，則 Treat 函數會傳回執行期錯誤。

如需次類型處理的其餘相關資訊，請參閱 *SQL 參考手冊*。

將集合轉換成基本幾何

請使用 ST_GeometryN 函數將幾何集合的元素轉換成外部呼叫函數需要的基本幾何。

例如，ST_Union 函數傳回的值一定是以 ST_Geometry 傳回的幾何集合。請使用 Treat 函數，將 ST_Geometry 類型轉換成下列其中一項次類型：ST_MultiPoint、ST_MultiLineString、ST_MultiPolygon、ST_GeomCollection、ST_MultiCurve 或 ST_MultiSurface。將 ST_GeometryN 函數套用到 Treat 函數的輸出，將幾何集合轉換成基本幾何。

例如，若要使用在上一節的範例中 ST_Union 函數結果中的 ST_ExteriorRing 函數，請先使用 ST_GeometryN 函數以取出多邊形元素。

```
SELECT ST_AsText(ST_ExteriorRing(ST_GeometryN(  
    TREAT ( ST_Union(c1.county, c2.county) AS ST_MultiPolygon ), 1)))  
FROM   counties AS c1, counties AS c2;
```

| 只有在從階層中的超類型進入次類型時才需要強制轉型。如需次類型處理的其餘
| 相關資訊，請參閱SQL 參考手冊。

AsShape

AsShape 使用幾何物件並傳回 BLOB。

語法

```
db2gse.AsShape(g db2gse.ST_Geometry)
```

回覆類型

BLOB(1m)

範例

下列程式碼片斷說明 AsShape 函數如何將 SENSITIVE_AREAS 表格的多邊形區域轉換成多邊形形狀。這些多邊形形狀將被傳遞到應用程式的 draw_polygon 函數，而顯示出來。

```
/* 建立 SQL 表示式。 */
strcpy(sqlstmt, "select db2gse.AsShape (zone) from SENSITIVE_AREAS
where db2gse.EnvelopesIntersect(zone, db2gse.PolyFromShape(cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 準備 SQL 陳述式。 */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* 設定形狀的 pcbvalue1 形狀。 */
pcbvalue1 = blob_len;

/* 連結形狀參數。 */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* 執行查詢。 */
rc = SQLExecute (hstmt);

/* 將查詢結果 (多邊形區域) 指定給
   fetched_binary 變數。 */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* 提取顯示視窗中的每一個多邊形並顯示。 */

while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
    draw_polygon(fetched_binary);
```

EnvelopesIntersect

EnvelopesIntersect 將在兩個幾何包封產生交集時傳回 1 (TRUE)；無交集時，則傳回 0 (FALSE)。

語法

```
db2gse.EnvelopesIntersect(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

get_window 函數從應用程式擷取顯示視窗的座標。實際上，視窗參數是多邊形形狀結構，它包含代表多邊形顯示畫面的座標字串。PolyFromShape 函數將顯示視窗形狀轉換成 Spatial Extender 多邊形，EnvelopesIntersect 函數便將它作為自己的交集包封。而傳回與顯示視窗的內部或界限交集的全部 SENSITIVE_AREAS 多邊形區域。每一個多邊形將從結果集提取，並傳遞給 draw_polygon 函數。

```
/* 取得多邊形形狀的顯示視窗座標。
```

```
get_window(&window)
```

```
/* 建立 SQL 表示式函數。db2gse.EnvelopesIntersect 函數
```

```
將用來限制結果集，僅包含那些與顯示視窗包封交集的
```

```
多邊形區域。*/
```

```
strcpy(sqlstmt, "select db2gse.AsShape(zone) from SENSITIVE_AREAS where  
db2gse.EnvelopesIntersect (zone, db2gse.PolyFromShape(cast(? as blob(1m)),  
db2gse.coordref()..srid(0)))");
```

```
/* 將 blob_len 設為 5 點多邊形形狀的位元組長度。*/
```

```
blob_len = 128;
```

```
/* 準備 SQL 陳述式。*/
```

```
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);
```

```
/* 將 pcbvalue1 設為視窗形狀 */
```

```
pcbvalue1 = blob_len;
```

```
/* 連結形狀參數。*/
```

```
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB,  
blob_len, 0, window, blob_len, &pcbvalue1);
```

```
/* 執行查詢。*/
```

```
rc = SQLExecute (hstmt);
```

```
/* 將查詢結果 (多邊形區域) 指定給
```

```
  fetched_binary 變數。*/
```

```
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);
```

```
/* 提取顯示視窗中的每一個多邊形並顯示。 */  
while(SQL_SUCCESS == (rc = SQLFetch(hstmt))  
      draw_polygon(fetched_binary);
```

GeometryFromShape

GeometryFromShape 使用形狀與空間參照系統識別字，以傳回幾何物件。

語法

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 C 程式碼片斷包含 Spatial Extender SQL 函數所內含的 ODBC 函數，這些函數在 LOTS 表格中插入資料。

LOTS 表格是以兩個直欄建立的：LOT_ID 直欄唯一定義每一塊用地，LOT 多邊形直欄包含每一塊用地的幾何。

```
CREATE TABLE LOTS ( lot_id integer,
                    lot db2gse.ST_Polygon);
```

GeometryFromShape 函數將形狀轉換成 Spatial Extender 幾何。整個 INSERT 陳述式將被複製到 shp_sql。INSERT 陳述式包含參數記號，以動態接受 LOT_ID 資料與 LOT 資料。

```
/* 建立 SQL INSERT 陳述式以移入 lot_id 及
   用地直欄。問號是參數記號，
   可指出執行期將擷取的 lot_id 及
   用地值。 */
strcpy (shp_sql,"insert into LOTS (lot_id,lot) values (?,
db2gse.GeometryFromShape (cast(? as blob(1m)), db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將整數索引鍵值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* 將形狀連結第二個參數。 */
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

Is3d

Is3d 使用幾何物件，如果物件具有 3D 座標時將傳回 1 (TRUE)；否則傳回 0 (FALSE)。

語法

```
db2gse.Is3d(g db2gse.ST_Geometry)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立具有兩個直欄的 THREEED_TEST 表格：整數類型的 GID 直欄及 G1 幾何直欄。

```
CREATE TABLE THREEED_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

INSERT 陳述式將兩點插入 THREEED_TEST 表格。第一點不包含 Z 座標，第二點則有包含。

```
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)))
```

下列 SELECT 陳述式列出 GID 直欄內容及 Is3d 函數的結果。函數在第一列傳回 0，表示沒有 Z 座標；在第二列傳回 1，表示有 Z 座標。

```
SELECT gid, db2gse.Is3d (g1) "Is it 3d?" FROM THREEED_TEST
```

傳回下列結果集。

gid	Is it 3d?
1	0
2	1

IsMeasured

IsMeasured 使用幾何物件，如果物件具有測量時將傳回 1 (TRUE)；否則傳回 0 (FALSE)。

語法

```
db2gse.IsMeasured(g db2gse.ST_Geometry)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立具有兩個直欄的 MEASURE_TEST 表格：GID 直欄唯一定義各列，G1 直欄儲存點幾何。

```
CREATE TABLE MEASURE_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式將兩筆記錄插入 MEASURE_TEST 表格。第一筆記錄儲存未測量的點。第二筆記錄的點已測量。

```
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)))
```

下列 SELECT 陳述式與其對應的結果集顯示 GID 直欄及 IsMeasured 函數的結果。IsMeasured 函數在第一列傳回 0，因為點未測量。在第二列中傳回 1，則是因為點已測量。

```
SELECT gid, db2gse.IsMeasured (g1) "Has measures?" FROM MEASURE_TEST  
gid      Has measures  
-----  -  
1         0  
2         1
```

LineFromShape

LineFromShape 使用點類型的形狀與空間參照系統識別字，並傳回線串。

語法

```
db2gse.Line FromShape(ShapeLineString Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_LineString
```

範例

下列程式碼片斷在 SEWERLINES 表格移入唯一的 ID、大小類別及水溝線的幾何。

CREATE TABLE 陳述式建立具有三個直欄的 SEWERLINES 表格。第一欄 SEWER_ID 唯一定義每一個水溝線。第二欄 CLASS (整數類型) 定義水溝線的類型，類型通常與水溝線的容量有關。第三欄 SEWER (線串類型) 儲存水溝線的幾何。

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer,
                          sewer db2gse.ST_LineString);

/* 建立 SQL insert 陳述式，以移入 sewer_id、大小類別及
   水溝線字串。問號是參數記號，
   可指出執行期將擷取的 sewer_id、類別及
   水溝幾何值。 */
strcpy (shp_sql, "insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.Line FromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將整數索引鍵值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* 將整數類別值連結第二個參數。 */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);

/* 將形狀連結第三個參數。 */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, blob_len, 0, sewer_shape, blob_len, &pcbvalue3);  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

LocateAlong

LocateAlong 使用幾何物件與測量，以多點方式傳回測量時發現的點集。

如果已提供多點及測量給 LocateAlong 作為輸入，且如果多點不包含此測量，則 LocateAlong 會傳回 POINT EMPTY。

語法

```
db2gse.LocateAlong(g db2gse.ST_Geometry, [measure] Double)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 CREATE TABLE 陳述式建立 LOCATEALONG_TEST 表格。LOCATEALONG_TEST 有兩個直欄：GID 直欄唯一定義每一列；G1 幾何直欄儲存範例幾何。

```
CREATE TABLE LOCATEALONG_TEST (gid integer, g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式插入兩列。第一列是多線串；第二列是多點。

```
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)))
```

在下列的 SELECT 陳述式與對應的結果集中，LocateAlong 函數將被導向，以尋找測量為 6.5 的點。第一列傳回含有兩點的多點。不過，第二列則傳回空點。針對線形圖徵（維度大於 0 的幾何），LocateAlong 可內插點；不過，對多點而言，目標測量必須完全相符。

```
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.

在下列的 **SELECT** 陳述式與對應的結果集中，**LocateAlong** 函數在兩列中皆傳回多點。目標測量 7 符合多線串與多點來源資料兩者。

```
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 使用一個幾何物件與兩個測量位置，並傳回一個幾何，代表兩個測量位置之間的不相連路徑集。

語法

```
db2gse.LocateBetween(g db2gse.ST_Geometry, [measure] Double, [measure]5 Double)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 CREATE TABLE 陳述式建立 LOCATEBETWEEN_TEST 表格。LOCATEALONG_TEST 有兩個直欄：GID 直欄唯一定義每一列；G1 多線串直欄儲存範例幾何。

```
CREATE TABLE LOCATEBETWEEN_TEST (gid integer, g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式在 LOCATEBETWEEN_TEST 表格中插入兩列。第一列是多線串，第二列是多點。

```
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))
```

下列 SELECT 陳述式與其對應的結果集顯示

LocateBetween 函數如何尋找測量 6.5 與 7.5 之間的測量。

第一列傳回含有數個線串的多線串。

第二列傳回多點，因為來源資料是多點。

來源資料有 0 維度（點或多點）時，測量必須完全相符。

```
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 22.14000000 6.50000000, 30.98000000 23.98000000 7.00000000,36.95000000 24.98000000 7.50000000)

```
2 MULTIPOINT M ( 30.19000000 18.47000000 7.00000000, 30.98000000 23.9  
8000000 7.00000000)
```

2 record(s) selected.

M

M 使用一點，並傳回其測量。

語法

```
db2gse.M(p db2gse.ST_Point)
```

回覆類型

Double

範例

下列 CREATE TABLE 陳述式建立 M_TEST 表格。M_TEST 有兩個直欄：GID 整數直欄唯一定義每一列；PT1 點直欄儲存範例幾何。

```
CREATE TABLE M_TEST (gid integer, pt1 db2gse.ST_Point)
```

下列 INSERT 陳述式插入一列含有測量的點，及一列不含測量的點。

```
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)))
```

在下列的 SELECT 陳述式與對應的結果集中，M 函數列出各點的測量值，因為第一點沒有測量，所以 M 函數傳回 NULL。

```
SELECT gid, db2gse.M (pt1) "The measure" FROM M_TEST
```

GID	The measure
1	-
2	+7.00000000000000E+000

2 record(s) selected.

MLine FromShape

MLineFromShape 使用多線串類型的形狀與空間參照系統識別字，並傳回多線串。

語法

```
db2gse.MLineFromShape(ShapeMultiLineString Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiLineString
```

範例

下列程式碼片斷在 WATERWAYS 表格中移入唯一的 ID、名稱及水路多線串。

WATERWAYS 表格是以 ID 與 NAME 直欄建立的，這些直欄定義表格中所儲存的每條溪流與河川系統。WATER 直欄是多線串，因為河川與溪流系統通常是數個多線串的聚集。

```
CREATE TABLE WATERWAYS (id          integer,
                          name       varchar(128),
                          water       db2gse.ST_MultiLineString);

/* 建立 SQL insert 陳述式，以移入 id、名稱及
   多線串。問號是參數記號，
   可指出執行期將擷取的 id、名稱及
   水路值。 */
strcpy (shp_sql,"insert into WATERWAYS (id,name,water)
values (?,?, db2gse.MLineFromShape (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將整數 ID 值連結第一個參數。 */

pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);
/* 將 VARCHAR 名稱值連結第二個參數。 */

pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);

/* 將形狀連結第三個參數。 */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);  
/* 執行 insert 陳述式。*/  
rc = SQLExecute (hstmt);
```

MPointFromShape

MPointFromShape 使用多點類型的形狀與空間參照系統識別字，以傳回多點。

語法

```
db2gse.MPointFromShape(ShapeMultiPoint (1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPoint
```

範例

下列程式碼片斷移入生物學家的 SPECIES_SITINGS 表格。

SPECIES_SITINGS 由三個直欄所建立。species 與 genus 直欄唯一定義每一列，sitings 多點則儲存發現生物種類的地點。

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                                genus varchar(32),
                                sitings db2gse.ST_MultiPoint);

/* 建立 SQL insert 陳述式，以移入 species、genus 及
   sitings。問號是參數記號，可指出
   執行期將擷取的名稱與水路值。 */
strcpy (shp_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.MPointFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將 VARCHAR species 值連結第一個參數。 */
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
                        SQL_CHAR, species_len, 0, species, species_len, &pcbvalue1);

/* 將 VARCHAR genus 值連結第二個參數。 */
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
                        SQL_CHAR, genus_len, 0, name, genus_len, &pcbvalue2);

/* 將形狀連結第三個參數。 */
pcbvalue3 = blob_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, sitings_len, 0, sitings_shape, sitings_len, &pcbvalue3);  
/* 執行 insert 陳述式。*/  
rc = SQLExecute (hstmt);
```

MPolyFromShape

MPolyFromShape 使用多重多邊形類型的形狀與空間參照系統識別字，以傳回多重多邊形。

語法

```
db2gse.MPolyFromShape(ShapeMultiPolygon Blob(1m), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPolygon
```

範例

下列程式碼片斷移入 LOTS 表格。

LOTS 表格儲存的 lot_id 可唯一定義每一塊用地；用地多重多邊形則包含用地行幾何。

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );

/* 建立 SQL insert 陳述式，以移入 lot id 及用地。
   問號是參數記號，可指出可指出執行期將擷取的
   lot_id 及用地值。 */
strcpy (shp_sql,"insert into LOTS (lot_id,lot)
values (?, db2gse.MPolyFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將 lot_id 整數值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* 將用地形狀連結第二個參數。 */
pcbvalue2 = lot_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
SQL_BLOB, lot_len, 0, lot_shape, lot_len, &pcbvalue2);

/* 執行 insert 陳述式。 */
rc = SQLExecute (hstmt);
```

PointFromShape

PointFromShape 使用點類型的形狀與空間參照系統識別字，以傳回點。

語法

```
db2gse.PointFromShape(db2gse.ShapePoint blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列程式片斷移入 HAZARDOUS_SITES 表格。

HAZARDOUS_SITES 表格由下列的 CREATE TABLE 陳述式所建立，儲存一些危險場所。location 直欄 (定義成一個點) 儲存一個位置，這位置是每一個危險場所的地理中心。

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name varchar(128),
                               location db2gse.ST_Point);

/* 建立 SQL insert 陳述式，以移入 site_id、名稱及
   位置。問號是參數記號，可指出
   執行期將擷取的 site_id、名稱及位置值。 */
strcpy (shp_sql,"insert into HAZARDOUS_SITES (site_id, name, location)
values (?,?, db2gse.PointFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將 site_id 整數值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* 將 VARCHAR 名稱值連結第二個參數。 */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* 將位置形狀連結第三個參數。 */
pcbvalue3 = location_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, location_len, 0, location_shape, location_len, &pcbvalue3);  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

PolyFromShape

PolyFromShape 使用多邊形類型的形狀與空間參照系統識別字，以傳回多邊形。

語法

```
db2gse.PolyFromShape (ShapePolygon Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Polygon
```

範例

下列程式片斷移入 SENSITIVE_AREAS 表格。問號代表 id、name、size、type 及 zone 值的參數記號 (在執行期可擷取這些值)。

SENSITIVE_AREAS 表格包含數個直欄，它們說明受威脅的機構，其中 zone 直欄包含機構的多邊形幾何。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);

/* 建立 SQL insert 陳述式，以移入 id、name、size、type 及
   區域。問號是參數記號，可指出
   執行期將擷取的 id、name、size、type 及 zone 值。 */
strcpy (shp_sql,"insert into SENSITIVE_AREAS (id, name, size, type, zone)
values (?, ?, ?, ?, db2gse.PolyFromShape (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將 id 整數值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site id, 0, &pcbvalue1);
/* 將 VARCHAR 名稱值連結第二個參數。 */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* 將 site 浮點連結第三個參數。 */
pcbvalue3 = 0;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
SQL_REAL, 0, 0, &size, 0, &pcbvalue3);
```

```
/* 將 varchar 類型連結第四個參數。 */  
pcbvalue4 = type_len;  
rc = SQLBindParameter (hstmt, 4, SQL_PARAM_INPUT, SQL_C_CHAR,  
    SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);  
  
/* 將多邊形區域連結第五個參數。 */  
pcbvalue5 = zone_len;  
rc = SQLBindParameter (hstmt, 5, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, zone_len, 0, zone_shp, zone_len, &pcbvalue5);  
  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ShapeToSQL

以給定的形狀表示法，供 ShapeToSQL 建構一個 db2gse.ST_Geometry 值。自動使用 SRID 值 0。

語法

```
db2gse.ShapeToSQL(ShapeGeometry blob(1M))
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 C 程式碼片斷包含 Spatial Extender SQL 函數所內含的 ODBC 函數，這些函數在 LOTS 表格中插入資料。LOTS 表格是以兩個直欄建立：一個是唯一識別每一塊用地的 lot_id，一個是含有每一塊用地的幾何的用地多重多邊形直欄。

```
CREATE TABLE lots (lot_id integer,  
                   lot      db2gse.ST_MultiPolygon);
```

ShapeToSQL 函數將形狀轉換成 Spatial Extender 幾何。整個 INSERT 陳述式將被複製到 shp_sql。INSERT 陳述式包含一些參數記號以便動態接受 LOT_id 和用地資料。

```
/* 建立 SQL INSERT 陳述式以移入用地 id 及  
   用地多邊形。問號是參數記號，  
   可指出執行期將擷取的 lot_id 及  
   用地值。 */  
strcpy (shp_sql,"insert into lots (lot_id, lot) values(?,  
db2gse.ShapeToSQL(cast(? as blob(1m)))");  
  
/* 為 SQL 陳述式 handle 配置記憶體，並結合此  
   陳述式 handle 與連接 handle。 */  
  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* 準備要執行的 SQL 陳述式。 */  
  
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);  
  
/* 將整數索引鍵值連結第一個參數。 */  
  
pcbvalue1 = 0;  
  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,  
                       SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);  
  
/* 將形狀連結第二個參數。 */  
  
pcbvalue2 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
  
/* 執行 insert 陳述式。 */  
  
rc = SQLExecute (hstmt);
```

ST_Area

ST_Area 使用一個多邊形或多重重多邊形並傳回它的區域。

語法

```
db2gse.ST_Area(s db2gse.ST_Surface)
db2gse.ST_Surface
db2gse.ST_Polygon
db2gse.ST_MultiSurface
db2gse.ST_MultiPolygon
```

回覆類型

Double

範例

都市工程師需要建築區列示。爲了取得此列示，GIS 技術人員需選取每一個建築物的覆蓋區域和建築物 ID。

建築物覆蓋區儲存在使用下列 CREATE TABLE 陳述式建立的 BUILDINGFOOTPRINTS 表格：

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id      integer,
    footprint   db2gse.ST_MultiPolygon);
```

爲了滿足都市工程師的要求，技術人員使用下列 SELECT 陳述式，從 BUILDINGFOOTPRINTS 表格選取唯一鍵、building_id 和每一個建築物覆蓋區：

```
SELECT building_id, db2gse.ST_Area (footprint) "Area"
FROM BUILDINGFOOTPRINTS;
```

SELECT 陳述式傳回下列結果集：

```
building_id  Area
-----
          506  +1.407680000000000E+003
          1208  +2.557590000000000E+003
           543  +1.807860000000000E+003
           178  +2.086710000000000E+003
           .
           .
           .
```

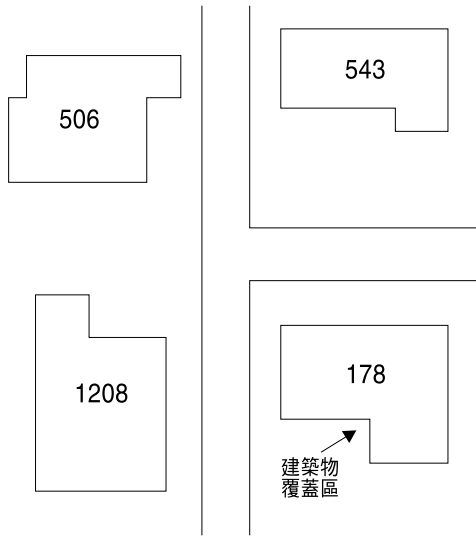


圖 28. 使用區域尋找建築物覆蓋區. 四個有標示建築物 ID 號碼的建築物覆蓋區沿著它們的相鄰街道顯示。

註: 如之前所說，`ST_Area` 可以使用多邊形或多重多邊形作為輸入。因此，輸入參數的資料類型可以是：

- 用於多邊形的資料類型（`db2gse.ST_Surface` 或 `db2gse.ST_Polygon`）
- 用於多重多邊形的資料類型（`db2gse.ST_MultiSurface` 或 `db2gse.ST_MultiPolygon`）

ST_AsBinary

ST_AsBinary 使用一個幾何物件並傳回它的知名二進位表示法。ST_AsBinary 不能使用空幾何作為輸入 (SQLSTATE 38827)。

語法

```
db2gse.ST_AsBinary(g db2gse.ST_Geometry)
```

回覆類型

BLOB(1m)

範例

下列程式碼片斷說明 ST_AsBinary 函數如何將 BUILDINGFOOTPRINTS 表格的覆蓋區多重多邊形轉換成 WKB 多重多邊形。這些多重多邊形傳到應用程式的 draw_polygon 函數以便顯示它們。

```
/* 建立 SQL 表示式。 */
strcpy(sqlstmt, "select db2gse.ST_AsBinary (footprint) from BUILDINGFOOTPRINTS
where db2gse.EnvelopesIntersect(footprint, db2gse.ST_PolyFromWKB
(cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 準備 SQL 陳述式。 */
SQLPrepare(hstmt, (UCHAR *)sqlstmt, SQL_NTS);

/* 設定形狀的 pcbvalue1 形狀。 */
pcbvalue1 = blob_len;

/* 連結形狀參數。 */
SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_BINARY, SQL_BLOB, blob_len,
0, shape, blob_len, &pcbvalue1);

/* 執行查詢。 */
rc = SQLExecute (hstmt);

/* 將查詢結果 (多邊形區域) 指定給
   fetched_binary 變數。
   */
SQLBindCol (hstmt, 1, SQL_C_Binary, fetched_binary, 100000, &ind_blob);

/* 提取顯示視窗中的每一個多邊形並顯示。 */
while(SQL_SUCCESS == (rc = SQLFetch(hstmt)))
    draw_polygon(fetched_binary);
```

ST_AsText

db2gse.ST_AsText 使用一個幾何物件並傳回它的知名文字表示法。

語法

```
db2gse.ST_AsText(g db2gse.ST_Geometry)
```

回覆類型

Varchar(4000)

範例

在後續實務範例中，db2gse.ST_AsText 函數將 HAZARDOUS_SITES 位置點轉換成它的文字說明：

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                                name      varchar(40),
                                location  db2gse.ST_Point);
INSERT INTO HAZARDOUS_SITES
VALUES (102,
        'W. H. Kleenare Chemical Repository',
        db2gse.ST_PointFromText('point (1020.12 324.02)',
        db2gse.coordref()..srid(0)));
```

```
SELECT site_id, name, cast(db2gse.ST_AsText(location) as varchar(40))
       "Location"
FROM HAZARDOUS_SITES;
```

SELECT 陳述式傳回下列結果集：

SITE_ID	Name	Location
102	W. H. Kleenare Chemical Repository	POINT (1020.00000000 324.00000000)

ST_Boundary

ST_Boundary 使用一個幾何物件並以幾何物件傳回它的合併界限。點及多點一定會造成一個空幾何的界限，且其維度為 0（不是 1 的維度）。

語法

```
db2gse.ST_Boundary(g db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

在下列程式碼片斷中，會建立名稱是 BOUNDARY_TEST 的表格。BOUNDARY_TEST 有兩個直欄：一個是定義成 varchar 的 GEOTYPE，一個是定義成超類別幾何的 G1。後面的 INSERT 陳述式插入每一個次類別幾何。ST_Boundary 函數擷取每一個次類別的界限，它儲存在 G1 幾何直欄。請注意：產生的幾何維度一定比輸入幾何少一個維度。點和多點一定會產生空白幾何（維度 1）的界限。線串和多線串傳回多點界限（維度 0）。多邊形或多重重多邊形一定會傳回多線串界限（維度 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	MULTIPOINT (10.02000000 20.01000000, 11.92000000 25.64000000)
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 使用一個幾何物件和距離並傳回圍繞來源物件的幾何物件。

語法

```
db2gse.ST_Buffer(g db2gse.ST_Geometry , [measure] Double)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

「縣行政官」需要危險場所列示，這些場所的 5 英哩半徑範圍涵蓋一些敏感區域，例如學校、醫院和療養院。敏感區域儲存在使用下列 CREATE TABLE 陳述式建立的表格 SENSITIVE_AREAS。ZONE 直欄被定義成多邊形，它儲存為每一個敏感區域的外框。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name        varchar(128),
                                size        float,
                                type        varchar(10),
                                zone        db2gse.ST_Polygon);
```

危險場所儲存在使用下列 CREATE TABLE 陳述式建立的 HAZARDOUS_SITES 表格。LOCATION 直欄 (定義成一個點) 儲存一個位置，這位置是每一個危險場所的地理中心。

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                                name      varchar(128),
                                location  db2gse.ST_Point);
INSERT INTO HAZARDOUS_SITES
VALUES (102, 'Allied Chemicals',
       db2gse.ST_PointFromText('Point(157000 475000)',coordref()..srid(0)))
```

db2gse.ST_Overlaps 函數合併 SENSITIVE_AREAS 及 HAZARDOUS_SITES 表格。若 SENSITIVE_AREAS 列的區域多邊形重疊 HAZARDOUS_SITES 位置點的緩衝的 5 英哩半徑範圍，對所有這些列，它傳回 1 (TRUE)。

```
SELECT sa.name "Sensitive Areas", hs.name "Hazardous Sites"
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Overlaps(sa.zone, db2gse.ST_Buffer (hs.location,(5 * 5280)))
= 1;
```

(5 * 5280 表示五英哩。這是從一英哩為 5280 英尺，且英尺是 VALUES 陳述式中指定的座標所屬的座標系統之長度單位)。

在圖29中，本行政區的部份敏感區域位於危險場所的 5 英哩緩衝區內。這兩個 5 英哩緩衝區與醫院相交，而且其中一個緩衝區與學校相交。不過療養院位於這兩個半徑範圍以外。

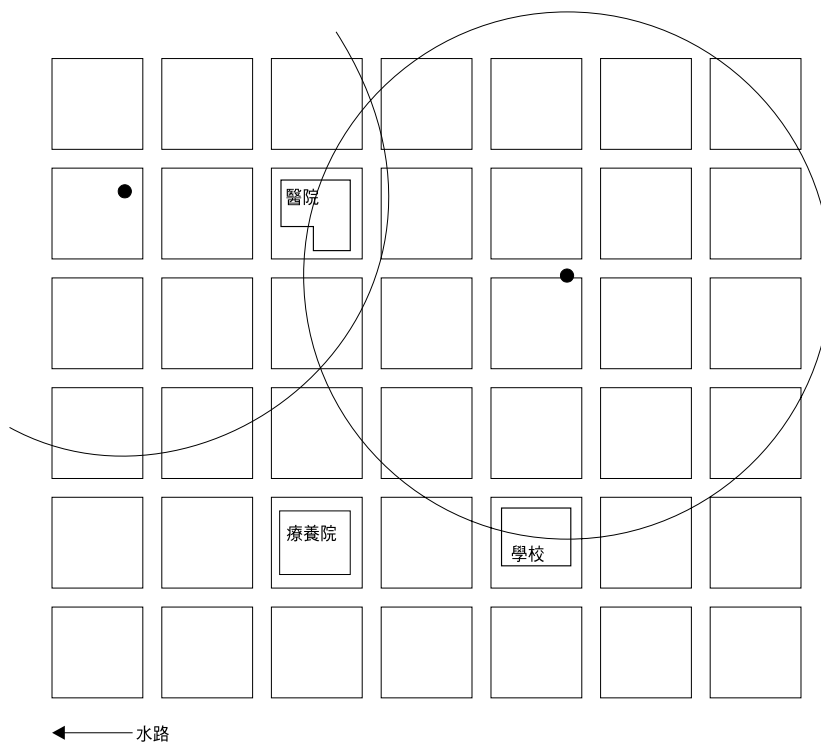


圖 29. 半徑 5 英哩範圍的緩衝區應用於一個點。

ST_Centroid

ST_Centroid 使用一個多邊形或多重重多邊形並以點傳回它的幾何中心。

語法

```
db2gse.ST_Centroid(s db2gse.ST_Surface)
db2gse.ST_Centroid(ms db2gse.ST_MultiSurface)
```

回覆類型

就表面而言：db2gse.ST_Point

範例

都市 GIS 技術人員想要在建築物密集圖形中以單一點顯示建築物覆蓋區的多重重多邊形。

建築物覆蓋區儲存在使用下列 CREATE TABLE 陳述式建立的 BUILDINGFOOTPRINTS 表格。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id      integer,
    footprint   db2gse.ST_MultiPolygon);
```

ST_Centroid 函數傳回每一個建築物覆蓋區多重重多邊形的距心。AsShape 函數將距心點轉換成一個形狀，即應用程式可辨識的外部表示法。

```
SELECT building_id,
       CAST(db2gse.AsShape(db2gse.ST_Centroid (footprint)) as blob(1m))
       "Centroid"
FROM BUILDINGFOOTPRINTS;
```

ST_Contains

ST_Contains 使用兩個幾何物件並傳回 1 (TRUE) (若第一個物件完全包含第二個物件)；否則它會傳回 0 (FALSE)。

語法

```
db2gse.ST_Contains(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

在下例中會建立兩個表格。一個表格含有一個都市的建築物覆蓋區，另一個表格含有它的用地。都市工程師想要確定全部建築物覆蓋區都在它們的用地內。

在這兩個表格中，多重多邊形資料類型儲存建築物覆蓋區和用地的幾何。資料庫設計者為這兩個圖徵選取多重多邊形。設計者瞭解用地可以被自然地形 (如河流) 隔開，而且建築物覆蓋區通常可以由一些建築物組成。

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                     lot_id      integer,
                                     footprint   db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (  lot_id integer, lot db2gse.ST_MultiPolygon );
```

都市工程師首先選取一些不是完全位於一塊用地內的建築物。

```
SELECT building_id
   FROM BUILDINGFOOTPRINTS, LOTS
  WHERE db2gse.ST_Contains(lot,footprint) = 0;
```

都市工程師瞭解第一個查詢會傳回全部建築物 ID 的列示，這些建築物 ID 有在用地多邊形以外的覆蓋區。但是都市工程師也知道本資訊不會指出其它建築物是否有指定正確的用地 ID。這個第二個查詢對 BUILDINGFOOTPRINTS 表格的 lot_id 直欄執行資料完整性檢查。

```
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;
```

在第203頁的圖30中，標示建築物 ID 的建築物覆蓋區位於自己的用地內。以點虛線說明用地線。雖然沒有顯示，但是這些線條延伸到街道中心線，而且完全包含用地和用地內的建築物覆蓋區。

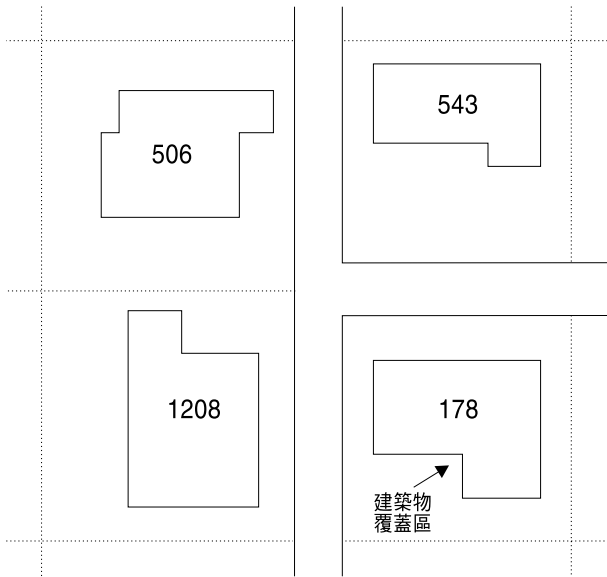


圖 30. 使用 `ST_Contains` 確定全部建築物位於自己的用地內。

ST_ConvexHull

ST_ConvexHull 使用一個幾何物件並傳回突多邊形區。

語法

```
db2gse.ST_ConvexHull(g db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

此範例建立 CONVEXHULL_TEST 表格，該表格有兩個直欄：GEOTYPE 和 G1。GEOTYPE 直欄 varchar(20) 會儲存幾何的次類別名稱（儲存在 G1），它是定義成幾何。

```
CREATE TABLE CONVEXHULL_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

每一個 INSERT 陳述式將每一個次類別類型的幾何插入 CONVEXHULL_TEST 表格。

```
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))
```

下列 SELECT 陳述式列示儲存在 GEOTYPE 直欄的次類別名稱和突多邊形區。

ST_AsText 函數將 ST_ConvexHull 函數產生的突多邊形區轉換成文字。

然後它會強制轉型為 varchar(256)，因為 ST_AsText 的預設輸出是 varchar(4000)。

```
SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_ConvexHull(G1))  
as varchar(256) "The convexhull"  
FROM CONVEXHULL_TEST
```

ST_CoordDim

ST_CoordDim 傳回 ST_Geometry 值的座標維度。關於座標維度的說明，請參閱第137頁的『Points』。

語法

```
db2gse.ST_CoordDim(g1 db2gse.ST_Geometry)
```

回覆類型

整數

範例

使用直欄 geotype 和 g1 建立 coorddim_test 表格。geotype 直欄儲存的幾何次類別名稱原本是儲存在 g1 幾何直欄。

```
CREATE TABLE coorddim_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

INSERT 陳述式將範例次類別插入 coorddim_test 表格。

```
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))
)

```

SELECT 陳述式列示儲存在 (含有該 geotype 的座標維度) 的次類別名稱。

```

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` 使用兩個幾何物件並傳回 1 (TRUE) (若這兩個物件的交集產生一個幾何物件，該物件的維度比來源物件的最大維度少一個維度)。此交集物件含有一些點，這些點是這兩個來源幾何的內部點而且不等於任何來源物件。否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Crosses(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

縣政府考慮實施一項新法規，規定縣內的危險廢棄物儲存設施距離水路要大於 5 英哩。縣 GIS 主管有河流的正確示意圖，這些河流以多線串儲存在 `WATERWAYS` 表格。但是對於每一個危險廢棄物儲存設施來說，GIS 主管只有單一點位置。

```
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);
```

若要判斷是否需要將違反此規定的設施通知縣行政官，那麼 GIS 主管應該緩衝危險場所位置，以瞭解有無河流經過緩衝區多邊形。`ST_Crosses` 述詞比較緩衝的 `HAZARDOUS_SITES` 與 `WATERWAYS`。所以只會傳回水路通過縣提議的規定半徑範圍的記錄。

```
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;
```

在第209頁的圖31中，危險廢棄物場所的 5 英哩緩衝區經過那些流經縣行政區的溪流網路。溪流網路被定義成多線串。所以結果集包括全部線串區段，這些線串區段是通過此半徑範圍的那些區段的一部份。

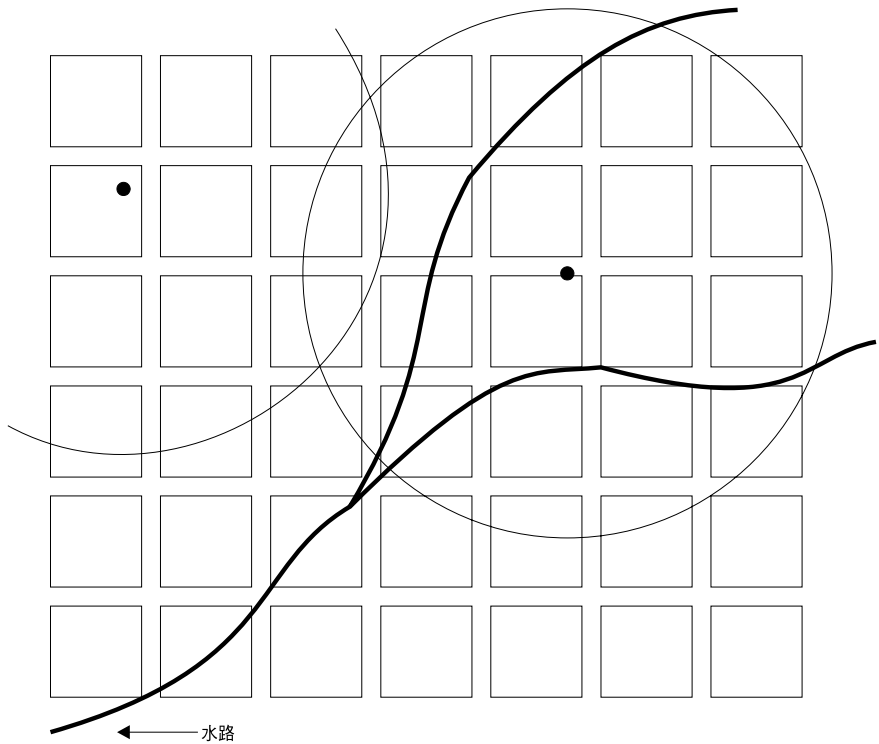


圖 31. 使用 *ST_Crosses* 尋找流經危險廢棄物區域的水路。

ST_Difference

ST_Difference 使用兩個幾何物件並傳回一個與來源物件不同的幾何物件。
ST_Difference 的兩個輸入幾何應有相同的維度，否則會傳回 NULL。

語法

```
db2gse.ST_Difference(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

都市工程師必須知道都市用地區域中尚未蓋建築物的總區域。亦即都市工程師想要除去建築區域之後的用地總計。

```
CREATE TABLE BUILDINGFOOTPRINTS (  building_id integer,
                                     lot_id      integer,
                                     footprint    db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id  integer,
                     lot     db2gse.ST_MultiPolygon);
```

都市工程師在 lot_id 上等式合併 BUILDINGFOOTPRINTS 與 LOTS 表格。然後工程師以用地的差異區域總和減去建築物覆蓋區。

```
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 使用一個幾何物件並傳回它的維度。

語法

```
db2gse.ST_Dimension(g1 db2gse.ST_Geometry)
```

回覆類型

整數

範例

使用直欄 GEOTYPE 和 G1 建立 DIMENSION_TEST 表格。GEOTYPE 直欄儲存的幾何次類別名稱原本是儲存在 G1 幾何直欄。

```
CREATE TABLE DIMENSION_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

INSERT 陳述式將範例次類別插入 DIMENSION_TEST 表格。

```
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)))
```

下列 `SELECT` 陳述式列示儲存在 `GEOTYPE` 直欄 (具有該 `geotype` 的維度) 的次類別名稱。

```
SELECT geotype, db2gse.ST_Dimension(g1) "The dimension"  
FROM DIMENSION_TEST
```

傳回下列結果集。

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 使用兩個幾何並傳回 1 (TRUE) (若這兩個幾何交點產生空集)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Disjoint(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

保險公司需要評估鎮的醫院、療養院和學校的承保範圍。包括判斷危險廢棄物場所對各個機構造成的威脅，也是本處理的一部份。保險公司想要只考慮無污染風險的機構。保險公司委託聘雇的 GIS 顧問尋找在危險廢棄物場所的 5 英哩半徑範圍以外的所有機構。

除了儲存機構的多邊形幾何的 ZONE 直欄以外，SENSITIVE_AREAS 表格還包含數個直欄，說明受威脅的機構。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

HAZARDOUS_SITES 表格在 SITE_ID 和 NAME 直欄儲存這些場所的識別字，每一個場所的實際地理位置儲存在 LOCATION 直欄。

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);
```

下列 SELECT 陳述式列示在危險廢棄物場所的 5 英哩半徑範圍以外的全部敏感區域名稱。ST_Intersects 函數可取代本查詢中的 ST_Disjoint 函數 (若此函數的結果集等於 0 而不是 1)。這是因為 ST_Intersects 和 ST_Disjoint 傳回相反的結果。

```
SELECT sa.name
FROM SENSITIVE_AREAS sa, HAZARDOUS_SITES hs
WHERE db2gse.ST_Disjoint(db2gse.ST_Buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

在第214頁的圖32中，敏感場所與危險廢棄物場所的 5 英哩半徑範圍做比較。療養院是 ST_Disjoint 函數會傳回 1 (TRUE) 的唯一敏感區域。每次有兩個幾何無法相交時，ST_Disjoint 函數就會傳回 1。

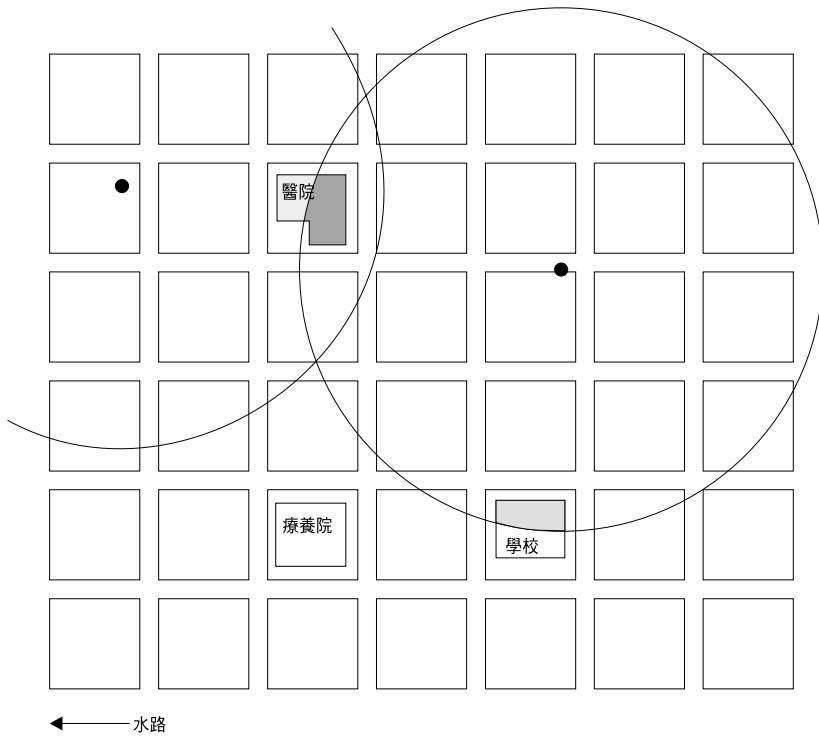


圖 32. 使用 *ST_Disjoint* 尋找不在 (相交) 危險廢棄物場所的建築物。

ST_Distance

ST_Distance 使用兩個幾何並傳回分隔它們的最短距離。

語法

```
db2gse.ST_Distance(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

Double

範例

都市工程師需要距任何用地線一英尺內的全部建築物的列示。

BUILDINGFOOTPRINTS 表格的 BUILDING_ID 直欄唯一識別每一棟建築物。LOT_ID 直欄識別建築物屬於的用地。覆蓋區多重多邊形儲存每一棟建築物覆蓋區的幾何。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

LOTS 表格儲存用地 ID，該 ID 唯一識別每一塊用地和含有用地線幾何的用地多重多邊形。

```
CREATE TABLE LOTS (
    lot_id integer,
    lot     db2gse.ST_MultiPolygon);
```

查詢傳回距用地線 1 英尺內的建築物 ID 列示。ST_Distance 函數執行用地多重多邊形的覆蓋區與界限的空間合併。不過 bf.lot_id 與 LOTS.lot_id 之間的等式合併，確保 ST_Distance 函數只會比較屬於相同用地的多重多邊形。

```
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 使用一條線串並傳回一個點 (這個點是該線串的最後一點)。

語法

```
db2gse.ST_Endpoint(c db2gse.ST_Curve)
```

回覆類型

```
db2gse.ST_Point
```

範例

ENDPOINT_TEST 表格儲存唯一識別每一列的 GID 整數直欄和儲存線串的 LN1 線串直欄。

```
CREATE TABLE ENDPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

INSERT 陳述式將線串插入 ENDPOINT_TEST 表格。第一條線串沒有 Z 座標或測量；第二條線串有 Z 座標和測量。

```
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))
```

下列 SELECT 陳述式列示具有 ST_Endpoint 函數輸出的 GID 直欄。

ST_Endpoint 函數產生一個點幾何，ST_AsText 函數將該幾何轉換成文字。

CAST 函數是用來將 ST_AsText 函數的預設 varchar(4000) 值縮短成 varchar(60)。

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_Endpoint(ln1)) AS varchar(60))
"Endpoint"
FROM ENDPOINT_TEST
```

傳回下列結果集。

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 使用一個幾何物件並傳回它的外框作為幾何。

語法

```
db2gse.ST_Envelope(g db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

ENVELOPE_TEST 表格中的 GEOTYPE 直欄儲存的幾何次類別名稱原本是儲存在 G1 幾何直欄。

```
CREATE TABLE ENVELOPE_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式將每一個幾何次類別插入 ENVELOPE_TEST 表格。

```
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)))

```

下列 SELECT 陳述式在次類別的包封旁邊列示次類別名稱。

因為 ST_Envelope 函數傳回點、線串或多邊形，

所以會使用 ST_AsText 函數將它的輸出轉換成文字。

CAST 函數將 ST_AsText 函數的預設 varchar(4000) 結果轉換成 varchar(280)。

```

SELECT GEOTYPE, CAST(db2gse.ST_AsText(db2gse.ST_Envelope(g1)) AS varchar(280))
"The envelope"
FROM ENVELOPE_TEST

```

傳回下列結果集。

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 比較兩個幾何並傳回 1 (TRUE) (若這兩個幾何相同)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Equals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

都市 GIS 技術人員懷疑 BUILDINGFOOTPRINTS 表格中有部份資料重複。技術人員查看表格，判斷是否任何覆蓋區的多重多邊形都相等。

使用下列陳述式建立 BUILDINGFOOTPRINTS 表格。BUILDING_ID 直欄唯一識別建築物；LOT_ID 直欄識別建築物的用地；FOOTPRINT 直欄儲存建築物的幾何。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

BUILDINGFOOTPRINTS 表格透過 ST_Equals 述詞進行對本身的空間合併，每次該述詞找到兩個相等的多重多邊形時就會傳回 1。需要 bf1.building_id <> bf2.building_id 條件去排除相同幾何的比較。

```
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 使用一個多邊形並以線串傳回它的外環。

語法

```
db2gse.ST_ExteriorRing(s db2gse.ST_Polygon)
```

回覆類型

```
db2gse.ST_LineString
```

範例

研究幾個南海島嶼上的鳥群的一位鳥類學家，知道某特定鳥類的哺育區侷限於海岸線。若要計算島嶼的裝載量，那麼鳥類學家需要島嶼的周邊長度。雖然部份島嶼有一些池塘，不過這些池塘的岸邊被另一種更具侵略性的鳥類獨佔。因此這位鳥類學家需要島嶼的外環周邊長度。

ISLANDS 表格的 ID 和 NAME 直欄識別島嶼，類型 ST_Polygon 的 LAND 直欄儲存島嶼的幾何。

```
CREATE TABLE ISLANDS (id integer,
                        name varchar(32),
                        land db2gse.ST_Polygon);
```

ST_ExteriorRing 函數從每一個島嶼多邊形取出外環作為線串。長度函數建立線串長度。SUM 函數彙總線串長度。

```
SELECT SUM(db2gse.ST_length(db2gse.ST_ExteriorRing (land))) FROM ISLANDS;
```

在第221頁的圖33中，這些島嶼的外環代表每一個島嶼與海洋共用的生態介面。部份島嶼有湖泊，多邊形的內環代表這些湖泊。

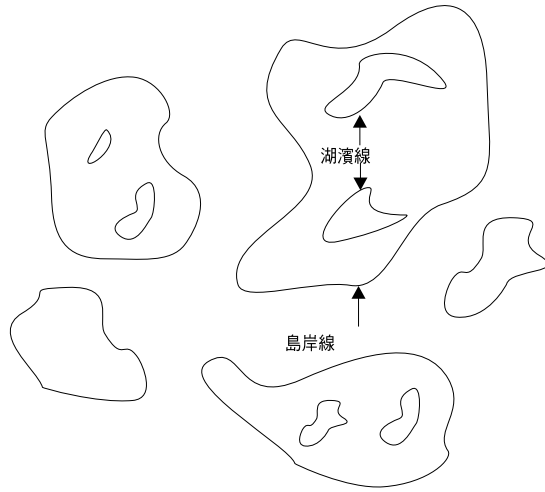


圖 33. 使用 `ST_ExteriorRing` 判斷一個島嶼海岸線長度。

ST_GeometryN

ST_GeometryN 使用一個集合和一個整數指標，然後將第 n 個幾何物件傳回該集合。

語法

```
db2gse.ST_GeometryN(g db2gse.ST_Geometry, n Integer)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

都市工程師必須知道建築物覆蓋區是否在用地的多重多邊形的第一個多邊形內。

BUILDING_ID 直欄唯一識別 BUILDINGFOOTPRINTS 表格的每一列。LOT_ID 直欄識別建築物用地。FOOTPRINT 直欄儲存建築物的幾何。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS (
    lot_id integer,
    lot    db2gse.ST_MultiPolygon);
```

查詢列示在第一塊用地多邊形內的全部建築物覆蓋區的 BUILDINGFOOTPRINTS building_id 和 lot_id。ST_GeometryN 函數傳回多重多邊形陣列中的第一塊用地多邊形。

```
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 使用一個 ST_Geometry 物件並以字串傳回它的幾何類型。

語法

```
db2gse.ST_GeometryType (g db2gse.ST_Geometry)
```

回覆類型

Varchar(4000)

範例

GEOMETRYTYPE_TEST 表格包含 G1 幾何直欄。

```
CREATE TABLE GEOMETRYTYPE_TEST(g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式將每一個幾何次類別插入 G1 直欄。

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES (db2gse.ST_GeomFromText('linestring (10.01 20.01, 10.01 30.01,
10.01 40.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_Geometrytype_test values(db2gse.ST_GeomFromText('polygon
((10.02 20.01,11.92 35.64,25.02 34.15,19.15 33.94, 10.02 20.01))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multipoint (10.02
20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64),
(9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRYTYPE_TEST
VALUES(db2gse.ST_GeomFromText('multipolygon (((10.02 20.01,11.92 35.64,
25.02 34.15, 19.15 33.94,10.02 20.01)),
((51.71 21.73,73.36 27.04,71.52 32.87,
52.43 31.90,51.71 21.73)))',
db2gse.coordref()..srid(0)))
```

下列 SELECT 陳述式列示儲存在 G1 幾何直欄的每一個次類別的幾何類型。

```
SELECT db2gse.ST_GeometryType(g1) "Geometry type" FROM GEOMETRYTYPE_TEST
```

傳回下列結果集。

幾何類型

ST_Point
ST_LineString
ST_Polygon
ST_MultiPoint
ST_MultiLineString
ST_MultiPolygon

6 record(s) selected.

ST_GeomFromText

ST_GeomFromText 使用一個知名文字表示法和一個空間參照系統識別字，並傳回一個幾何物件。

語法

```
db2gse.ST_GeomFromText(geometryTaggedText Varchar(4000), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

GEOMETRY_TEST 表格包含整數 **GID** 直欄 (它唯一識別每一列) 和 **G1** 直欄 (它儲存幾何)。

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

INSERT 陳述式將資料插入 GEOMETRY_TEST 表格的 **GID** 和 **G1** 直欄。ST_GeomFromText 函數將幾何的文字表示法轉換成對應的 Spatial Extender 個體化次類別。

```
INSERT INTO GEOMETRY_TEST
VALUES(1, db2gse.ST_GeomFromText('point (10.02 20.01)',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES (2,
db2gse.ST_GeomFromText('linestring (10.01 20.01, 10.01 30.01,
10.01 40.01)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(3,
db2gse.ST_GeomFromText('polygon ((10.02 20.01,11.92 35.64,25.02 34.15,
19.15 33.94,10.02 20.01))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(4,
db2gse.ST_GeomFromText('multipoint (10.02 20.01,10.32 23.98,
11.92 25.64)', db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
VALUES(5,
db2gse.ST_GeomFromText('multilinestring ((10.02 20.01,10.32 23.98,
11.92 25.64),
(9.55 23.75,15.36 30.11))',
db2gse.coordref()..srid(0)))
```

```
INSERT INTO GEOMETRY_TEST
```

```
VALUES(6,  
      db2gse.ST_GeomFromText('multipolygon (((10.02 20.01,11.92 35.64,  
25.02 34.15, 19.15 33.94,10.02 20.01)),  
                                ((51.71 21.73,73.36 27.04,71.52 32.87,  
                                52.43 31.90,51.71 21.73)))',  
db2gse.coordref()..srid(0))
```

ST_GeomFromWKB

ST_GeomFromWKB 使用一個知名二進位表示法和空間參照系統識別字，並傳回一個幾何物件。

語法

```
db2gse.ST_GeomFromWKB(WKBGeometry Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 C 程式碼片斷包含 Spatial Extender SQL 函數所內含的 ODBC 函數，這些函數在 LOTS 表格中插入資料。

OTS 表格是以兩個直欄建立的：LOT_ID 直欄唯一定義每一塊用地，LOT 多重多邊形直欄包含每一塊用地的幾何。

```
CREATE TABLE LOTS ( lot_id integer,
                    lot db2gse.ST_MultiPolygon);
```

ST_GeomFromWKB 函數將 WKB 表示轉換成 Spatial Extender 幾何。整個 INSERT 陳述式被複製到 wkb_sql 字串。INSERT 陳述式包含參數記號，以動態接受 LOT_ID 資料與 LOT 資料。

```
/* 建立 SQL INSERT 陳述式以移入用地 id 及
   用地多邊形。問號是參數記號，
   可指出執行期將擷取的 lot_id 及
   用地值。 */
strcpy (wkb_sql,"insert into LOTS (lot_id, lot) values (?,
db2gse.ST_GeomFromWKB

(cast(? as blob(1m)), db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 將整數索引鍵值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* 將形狀連結第二個參數。 */
pcbvalue2 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);  
  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ST_InteriorRingN

以線串傳回多邊形的第 n 個內環。這些內環不是按照幾何方位排列。而是根據內部幾何驗證常式定義的規則排列。所以無法預先定義內環次序。

語法

```
ST_InteriorRingN(p ST_Polygon, n Integer)
```

回覆類型

```
db2gse.ST_LineString
```

範例

研究幾個南海島嶼上的鳥群的一位鳥類學家，知道某特定弱勢鳥類的哺育區侷限於海岸線。部份島嶼有湖泊。另一種較具侵略性鳥類獨佔這些湖泊的湖岸。這位鳥類學家知道，就每一個島嶼來說，若湖泊的周邊長度超過某臨界值，侵略性鳥類不斷繁殖而威脅到弱勢海岸鳥類的生存。因此這位鳥類學家需要島嶼的內環周長總計。

在圖34中，這些島嶼的外環代表每一個島嶼與海洋共用的生態介面。部份島嶼有湖泊，多邊形的內環代表這些湖泊。

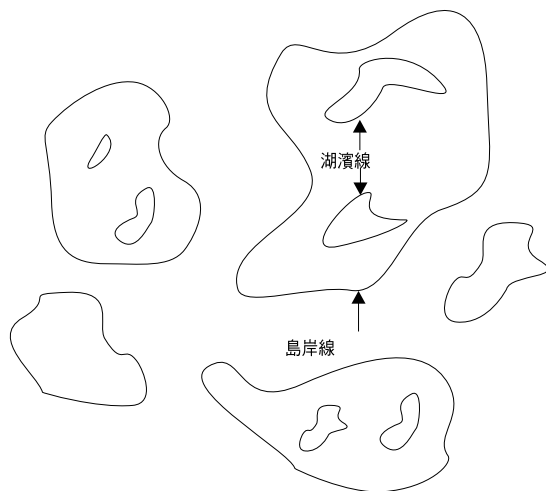


圖 34. 使用 *ST_InteriorRingN* 判斷島嶼內的湖岸長度。

ISLANDS 表格的 ID 和名稱直欄識別各個島嶼，用地多邊形直欄儲存島嶼的幾何。

```
CREATE TABLE ISLANDS (id integer,
                      name varchar(32),
                      land db2gse.ST_Polygon);
```

下列 ODBC 程式使用 ST_InteriorRingN 函數，從每一個島嶼多邊形取出內環 (湖泊) 作為線串。長度函數傳回的線串周長經過總計並連同島嶼 ID 一起顯示。

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <time.h>

#include "sg.h"
#include "sgerr.h"
#include "sqlclil.h"

/** 變更這些常數 ***/
#define USER_NAME "sdetest" /* 您的使用者名稱 */
#define USER_PASS "acid.rain" /* 您的使用者通行碼 */
#define DB_NAME "mydb" /* 要連接的資料庫 */

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;

    /* 配置記憶體給 ODBC 環境 handle henv 並起始設定應用程式。 */
    rc = SQLAllocEnv (&henv);
    if (rc != SQL_SUCCESS)
    {
        printf ("SQLAllocEnv failed with %d\n", rc);
        exit(0);
    }
}
```

```

    }

/* 配置記憶體給 henv 環境內的連接 handle。*/
rc = SQLAllocConnect (henv, &handle);
if (rc != SQL_SUCCESS)
{
    printf ("SQLAllocConnect failed with %d\n", rc);
    exit(0);
}

/* 載入 ODBC 驅動程式，並連接資料庫、使用者和通行碼所識別的資料來源。*/
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");

/* 配置記憶體給 SQL 陳述式 handle island_cursor。*/

rc = SQLAllocStmt (handle, &island_cursor);
check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* 準備並執行查詢以取得島嶼 ID 和湖泊數目 (內環)*/
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");

/* 連結島嶼表格的 ID 直欄與變數 island_id */

rc = SQLBindCol (island_cursor, 1, SQL_C_SLONG, &island_id, 0, &id_ind);
check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* 連結 numinteriorrings(land) 的結果與 num_lakes 變數。*/

rc = SQLBindCol (island_cursor, 2, SQL_C_SLONG, &num_lakes, 0, &lake_ind);
check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* 配置記憶體給 SQL 陳述式 handle lake_cursor。*/

rc = SQLAllocStmt (handle, &lake_cursor);
check_sql_err (handle, NULL, rc, "SQLAllocStmt");

/* 準備查詢以取得內環的長度。*/

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");
/* 連結 lake_number 變數成為第一個輸入參數 */

    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");

/* 連結 island_id 成為第二個輸入參數 */

    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");

/* 將 Length(db2gse.ST InteriorRingN(land, cast
    (? as integer))) 的結果連結到變數 lake_perimeter */

    rc = SQLBindCol (lake_cursor, 1, SQL_C_SLONG, &lake_perimeter, 0,
        &length_ind);
    check_sql_err (NULL, island_cursor, rc, "SQLBindCol");

/* 外部迴圈，取得島嶼 ID 和湖泊數目 (內環) */
while (SQL_SUCCESS == rc)
{
    /* 提取一個島嶼 */

    rc = SQLFetch (island_cursor);

    if (rc != SQL_NO_DATA)
    {
        check_sql_err (NULL, island_cursor, rc, "SQLFetch");

        /* 內部迴圈，針對此島嶼，取得其全部湖泊的周長 (內環) */
        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);
        }
    }
}

/* 顯示島嶼 ID 及其湖泊的總周長。*/

    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 使用兩個幾何物件並傳回交集作為幾何物件。

如果已給定 ST_Intersection 一個多邊形及線串作為輸入參數，且適用於下列條件：

- 多邊形中的一點是線串的起點，且
- 多邊形及線串沒有任何其它共同的點。

則 ST_Intersection 會傳回字串 POINT EMPTY。

語法

```
db2gse.ST_Intersection(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

消防隊隊長必須取得與可能的危險廢棄物污染半徑範圍相交的醫院、學校和療養院區域。

敏感區域儲存在使用下列 CREATE TABLE 陳述式建立的表格 SENSITIVE_AREAS。ZONE 直欄被定義成多邊形，它儲存每一個敏感區域的外框。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name       varchar(128),
                                size       float,
                                type       varchar(10),
                                zone       db2gse.ST_Polygon);
```

危險場所儲存在使用下列 CREATE TABLE 陳述式建立的 HAZARDOUS_SITES 表格。LOCATION 直欄 (定義成一個點) 儲存一個位置，這位置是每一個危險場所的地理中心。

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                                name      varchar(128),
                                location  db2gse.ST_Point);
```

緩衝區函數產生一個 5 英哩緩衝區，該緩衝區圍繞危險廢棄物場所。ST_Intersection 函數從緩衝的危險廢棄物場所多邊形與敏感地區的交集產生一些多邊形。ST_Area 函數傳回交集多邊形區域，SUM 函數為每一個危險場所彙總該區域。GROUP BY 子句指示查詢根據危險廢棄物 site_ID 聚集相交區域。


```

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;

```

在圖35中，圓形代表圍繞危險廢棄物場所的緩衝多邊形。這些緩衝區多邊形與敏感區多邊形的交集產生三個多邊形。左上角的醫院相交兩次，右下角的學校只相交一次。

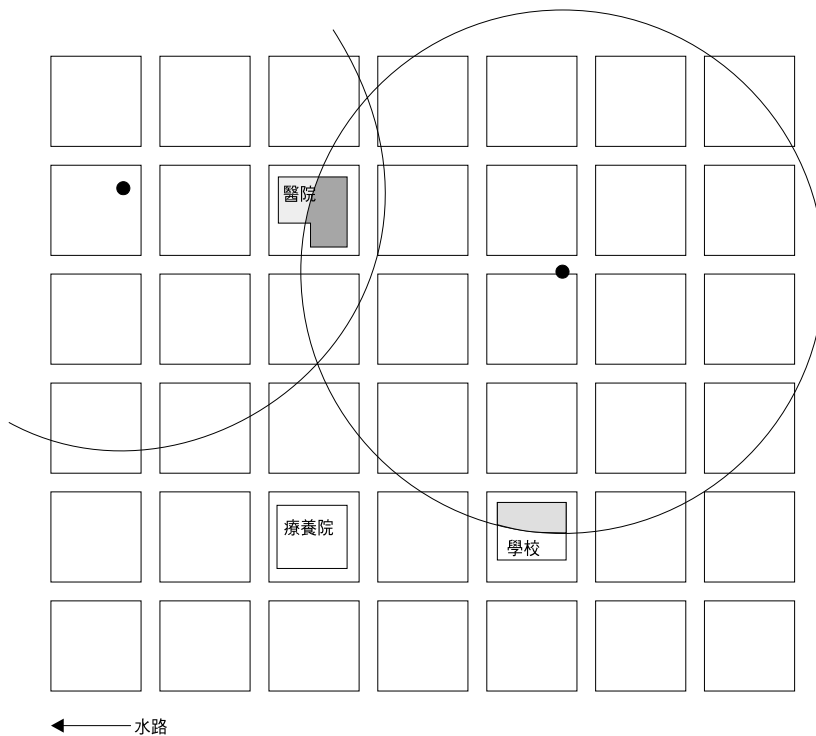


圖 35. 使用 *ST_Intersection* 判斷危險廢棄物可能影響每一棟建築物的區域大小。

ST_Intersects

ST_Intersects 使用兩個幾何並傳回 1 (TRUE) (若兩個幾何的交集沒有產生空集)。否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Intersects(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

消防隊隊長需要一個危險廢棄物場所的 5 英哩半徑範圍內的敏感區域列示。

敏感區域儲存在使用下列 CREATE TABLE 陳述式建立的表格 SENSITIVE_AREAS。ZONE 直欄被定義成多邊形，它儲存每一個敏感區域的外框。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name        varchar(128),
                                size        float,
                                type        varchar(10),
                                zone        db2gse.ST_Polygon);
```

危險場所儲存在使用下列 CREATE TABLE 陳述式建立的 HAZARDOUS_SITES 表格。LOCATION 直欄 (定義成一個點) 儲存一個位置，這位置是每一個危險場所的地理中心。

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                                name       varchar(128),
                                location   db2gse.ST_Point);
```

此查詢傳回敏感區域和敏感區域的危險場所名稱列示，與危險場所的 5 英哩緩衝區相交。

```
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 使用一條線串或多線串並傳回 1 (TRUE) (若它是封閉的)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_IsClosed(c db2gse.ST_Curve)
db2gse.ST_IsClosed(mc db2gse.ST_MultiCurve)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立 CLOSED_LINestring 表格，它有單一線串直欄。

```
CREATE TABLE CLOSED_LINestring (ln1 db2gse.ST_LineString)
```

下列 INSERT 陳述式將兩筆記錄插入 CLOSED_LINestring 表格。第一筆記錄不是封閉的線串，第二筆記錄是封閉的線串。

```
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)))
```

下列 SELECT 陳述式和對應的結果集顯示 ST_IsClosed 函數的結果。第一列傳回 0，因為線串沒有封閉，第二列傳回 1，因為線串已封閉。

```
SELECT db2gse.ST_IsClosed(ln1) "Is it closed" FROM CLOSED_LINestring
```

```
Is it closed
```

```
-----
          0
```

```
1
```

```
2 record(s) selected.
```

下列 CREATE TABLE 陳述式建立 CLOSED_MULTILINESTRING 表格，它有單一多線串直欄。

```
CREATE TABLE CLOSED_MULTILINESTRING (m1n1 db2gse.ST_MultiLineString)
```

下列 INSERT 陳述式將兩筆記錄插入 CLOSED_MULTILINESTRING，一筆多線串記錄沒有封閉，另一筆記錄已封閉。

```

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)))

```

下列 SELECT 陳述式和對應的結果集顯示 ST_IsClosed 函數的結果。
 第一列傳回 0，因為多線串沒有封閉，第二列傳回 1，因為多線串已封閉。
 若一個多線串的全部線串元素已封閉，就會封閉多線串。

```
SELECT db2gse.ST_IsClosed(mln1) "Is it closed" FROM CLOSED_MULTILINESTRING
```

```

Is it closed
-----
           0
1

```

2 record(s) selected.

ST_IsEmpty

ST_IsEmpty 使用一個幾何物件並傳回 1 (TRUE) (若它是空白)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_IsEmpty(g db2gse.ST_Geometry)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立具有兩個直欄的 EMPTY_TEST 表格。GEOTYPE 直欄儲存的次類別資料類型原本是儲存在 G1 幾何直欄。

```
CREATE TABLE EMPTY_TEST (geotype varchar(20), g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式插入幾何次類別點、線串和多邊形的兩筆記錄。一筆是空記錄，另一筆不是空記錄。

```
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)))
```

下列 SELECT 陳述式和對應的結果集顯示 GEOTYPE 直欄中的幾何類型和 ST_IsEmpty 函數結果。

```
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 使用線串，若它是環 (亦即線串是封閉且單純) 就會傳回 1 (TRUE)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_IsRing(c db2gse.ST_Curve)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立 RING_LINestring 表格，它有一個稱為 LN1 的單一線串直欄。

```
CREATE TABLE RING_LINestring (ln1 db2gse.ST_LineString)
```

下列 INSERT 陳述式將三個線串插入 LN1 直欄。第一列含有一個沒有封閉的線串，因此不是環。第二列含有一個封閉而且單純的線串，因此是環。第三列含有一個封閉但不單純的線串，因為它與自己的內部相交，所以不是環。

```
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)))
```

下列 SELECT 陳述式和對應的結果集顯示 ST_IsRing 函數結果。第一列和第三列傳回 0。這是因為線串不是環，第二列傳回 1，因為它是環。

```
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 使用一個幾何物件並傳回 1 (TRUE) (若此物件單純)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_IsSimple(g db2gse.ST_Geometry)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立 ISSIMPLE_TEST 表格，它有兩個直欄。PID 直欄 (它是 smallint) 含有每一列的唯一 ID。G1 幾何直欄儲存單純和不單純幾何範例。

```
CREATE TABLE ISSIMPLE_TEST (pid smallint, g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式將兩筆記錄插入 ISSIMPLE_TEST 表格。第一筆記錄屬單純，因為它是沒有與內部相交的線串。第二筆記錄屬不單純，因為它與內部相交。

```
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)))
```

下列 SELECT 陳述式和對應的結果集顯示 ST_IsSimple 函數結果。第一筆記錄傳回 1，因為線串屬單純，第二筆記錄傳回 0，因為線串不單純。

```
SELECT ST_IsSimple(g1)  
FROM ISSIMPLE_TEST
```

```
g1  
-----  
1  
          0
```

ST_IsValid

ST_IsValid 使用一個 ST_Geometry 並傳回 1 (TRUE) (若它有效)，否則會傳回 0 (FALSE)。插入 DB2 資料庫的幾何一定有效，因為 Spatial Extender 一定驗證它的空間資料之後才接受它。不過其它 DBMS 供應商可能沒有驗證此輸入，但是會要求應用程式這麼做。

語法

```
db2gse.ST_IsValid(g db2gse.ST_Geometry)
```

回覆類型

整數

範例

使用直欄 geotype 和 g1 建立 valid_test 表格。 geotype 直欄儲存的幾何次類別名稱原本是儲存在 g1 幾何直欄。

```
CREATE TABLE valid_test (geotype varchar(20), g1 db2gse.ST_Geometry)
```

INSERT 陳述式將範例次類別插入 valid_test 表格。

```
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))
)
```

SELECT 陳述式列示儲存在 geotype 直欄 (含有該 geotype 的維度) 的次類別名稱。

```
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 使用線串或多線串並傳回它的長度。

語法

```
db2gse.ST_Length(c db2gse.ST_Curve)
db2gse.ST_Length(mc db2gse.ST_MultiCurve)
```

回覆類型

Double

範例

一位當地生態學家研究縣內水路中的鮭魚群的遷移模式。這位生態學家想要取得流經此縣的所有河流長度。

下列 CREATE TABLE 陳述式建立 WATERWAYS 表格。ID 和 NAME 直欄識別儲存在該表格中的溪流和河川系統。WATER 直欄是多線串，因為河川與溪流系統通常是數個多線串的聚集。

```
CREATE TABLE WATERWAYS (id integer, name varchar(128),
                           water          db2gse.ST_MultiLineString);
```

下列 SELECT 陳述式使用 ST_Length 函數傳回縣內每一條水路的名稱和長度。

```
SELECT name, db2gse.ST_Length(water) "Length"
FROM WATERWAYS;
```

第246頁的圖36顯示位於縣邊界內的河川與溪流系統。

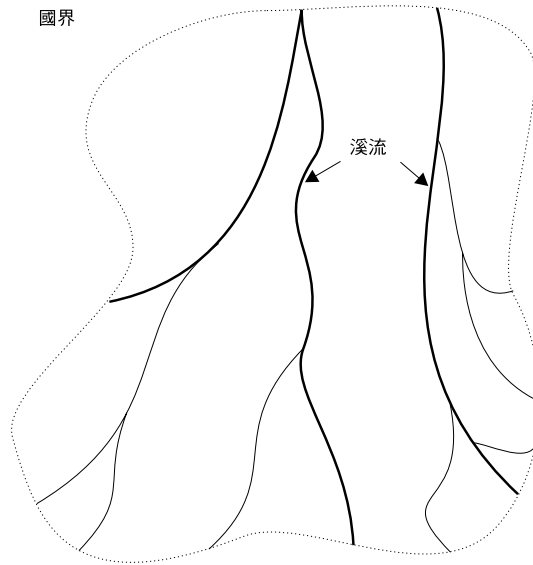


圖 36. 使用 *ST_Length* 判斷一個縣的水路總長。

ST_LineFromText

ST_LineFromText 使用線串類型的知名文字表示法和空間參照系統識別字，並傳回一條線串。

語法

```
db2gse.ST_LineFromText(lineStringTaggedText Varchar(4000), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_LineString
```

範例

下列 CREATE TABLE 陳述式建立 LINESTRING_TEST 表格，它有單一 LN1 線串直欄。

```
CREATE TABLE LINESTRING_TEST (ln1 db2gse.ST_LineString)
```

下列 INSERT 陳述式使用 ST_LineFromText 函數將線串插入 LN1 直欄。

```
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 採用線串類型的知名二進位表示法與空間參照系統 ID，然後傳回線串。

語法

```
db2gse.ST_LineFromWKB(WKBLineString Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_LineString
```

範例

下列程式碼片斷在 SEWERLINES 表格移入唯一的 ID、大小類別及水溝線的幾何。

使用三個直欄建立 SEWERLINES 表格。第一欄 SEWER_ID 唯一定義每一個水溝線。整數類型的第二個直欄 CLASS 識別水溝線類型，該類型通常與水溝線容量有關。第三欄 SEWER (線串類型) 儲存水溝線的幾何。

```
CREATE TABLE SEWERLINES (sewer_id integer,
                          class integer,
                          sewer db2gse.ST_LineString);

/* 建立 SQL insert 陳述式，移入 sewer_id、大小類別和
   水溝線。問號是參數記號，
   可指出執行期將擷取的 sewer_id、類別及
   水溝幾何值。 */
strcpy (wkb_sql,"insert into sewerlines (sewer_id,class,sewer)
values (?,?, db2gse.ST_LineFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 連結整數 sewer_id 值與第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_id, 0, &pcbvalue1);

/* 將整數類別值連結第二個參數。 */
pcbvalue2 = 0;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &sewer_class, 0, &pcbvalue2);

/* 將形狀連結第三個參數。 */
pcbvalue3 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, sewer_wkb, blob_len, &pcbvalue3);  
  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ST_MLineFromText

ST_MLineFromText 採用多線串類型的知名文字表示法及空間參照系統 ID，然後傳回多線串。

語法

```
db2gse.ST_MLineFromText(multiLineStringTaggedText String, SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiLineString
```

範例

下列 CREATE TABLE 陳述式建立 MLINESTRING_TEST 表格。MLINESTRING_TEST 有兩個直欄：一個是唯一識別列的 GID smallint 直欄，一個是 ML1 多線串直欄。

```
CREATE TABLE ST_MLINESTRING_TEST (gid smallint, ml1 db2gse.ST_MultiLineString)
```

下列 INSERT 陳述式使用 ST_MLineFromText 函數插入多線串。

```
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),  
db2gse.coordref()..srid(0))', (20.93 20.81, 21.52 40.10))',  
db2gse.coordref()..srid(0)))
```

ST_MLineFromWKB

ST_MLineFromWKB 採用多線串類型的知名二進位表示法及空間參照系統 ID，然後傳回多線串。

語法

```
db2gse.ST_MLineFromWKB(WKBMultiLineString Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiLineString
```

範例

下列程式碼片斷在 WATERWAYS 表格中移入唯一的 ID、名稱及水路多線串。

使用 ID 和 NAME 直欄建立 WATERWAYS 表格，它識別儲存在表格中的溪流與河川系統。WATER 直欄是多線串，因為河川與溪流系統通常是數個多線串的聚集。

```
CREATE TABLE WATERWAYS (id          integer,
                          name       varchar(128),
                          water       db2gse.ST_MultiLineString);

/* 建立 SQL insert 陳述式，以移入 id、名稱及
   多線串。問號是參數記號，
   可指出執行期將擷取的 id、名稱及
   水路值。 */
strcpy (shp_sql,"insert into WATERWAYS (id,name,water)
values (?,?, db2gse.ST_MLineFromWKB (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)shp_sql, SQL_NTS);

/* 將整數 ID 值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);

/* 將 VARCHAR 名稱值連結第二個參數。 */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, name_len, 0, &name, name_len, &pcbvalue2);

/* 將形狀連結第三個參數。 */
pcbvalue3 = blob_len;
```

```
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,  
    SQL_BLOB, blob_len, 0, water_shape, blob_len, &pcbvalue3);  
  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ST_MPointFromText

ST_MPointFromText 採用多點類型的知名文字表示法及空間參照系統 ID，然後傳回多點。

語法

```
db2gse.ST_MPointFromText(multiPointTaggedText Varchar(4000), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPoint
```

範例

下列 CREATE TABLE 陳述式使用一個多點直欄 MPT1 建立 MULTIPOINT_TEST 表格。

```
CREATE TABLE MULTIPOINT_TEST (mpt1 db2gse.ST_MultiPoint)
```

下列 INSERT 陳述式使用 ST_MPointFromText 直欄將多點插入 MPT1 直欄。

```
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 採用多點類型的知名二進位表示法及空間參照系統 ID，然後傳回多點。

語法

```
db2gse.ST_MPointFromWKB(WKBMultiPoint Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPoint
```

範例

下列程式碼片斷移入 SPECIES_SITINGS 表格。

SPECIES_SITINGS 由三個直欄所建立。SPECIES 和 GENUS 直欄唯一識別每一列，SITINGS 多點直欄儲存發現生物種類的地點。

```
CREATE TABLE SPECIES_SITINGS (species varchar(32),
                               genus varchar(32),
                               sitings db2gse.ST_MultiPoint);

/* 建立 SQL insert 陳述式，以移入 species、genus 及
   sitings。問號是參數記號，
   指出會在執行期擷取的 species (種)、genus (屬) 和
   sitings (位置) 值。*/
strcpy (wkb_sql,"insert into SPECIES_SITINGS (species,genus,sitings)
values (?,?, db2gse.ST_MPointFromWKB (cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。*/
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。*/
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 將 VARCHAR species 值連結第一個參數。*/
pcbvalue1 = species_len;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, species_len, 0, &species, species_len, &pcbvalue1);
/* 將 VARCHAR genus 值連結第二個參數。*/
pcbvalue2 = genus_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, genus_len, 0, &name, genus_len, &pcbvalue2);

/* 將形狀連結第三個參數。*/
pcbvalue3 = sitings_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, sitings_len, 0, sitings_wkb, sitings_len, &pcbvalue3);  
/* 執行 insert 陳述式。*/  
rc = SQLExecute (hstmt);
```

ST_MPolyFromText

ST_MPolyFromText 採用多重多邊形類型的知名文字表示法與空間參照系統 ID，然後傳回多重多邊形。

如果多重多邊形含有多個座標相同的多邊形，則此函數無法用它作為輸入。

語法

```
db2gse.ST_MPolyFromText(multiPolygonTaggedText Varchar(4000), SRID  
db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPolygon
```

範例

下列 CREATE TABLE 陳述式建立 MULTIPOLYGON_TEST 表格，它有單一多重多邊形直欄 MPL1。

```
CREATE TABLE MULTIPOLYGON_TEST (mpl1 db2gse.ST_MultiPolygon)
```

下列 INSERT 陳述式使用 ST_MPolyFromText 函數將多重多邊形插入 MPL1 直欄。

```
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 採用多重多邊形類型的知名二進位表示法及空間參照系統 ID，然後傳回多重多邊形。

語法

```
db2gse.ST_MPolyFromWKB(WKBMultiPolygon Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_MultiPolygon
```

範例

下列程式碼片斷移入 LOTS 表格。

LOTS 表格儲存 LOT_ID (它唯一識別每一塊用地) 和 LOT 多重多邊形 (它含有用地線幾何)。

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );
```

```
/* 建立 SQL insert 陳述式，以移入 lot_id 及用地。  
   問號是參數記號，可指出可指出執行期將擷取的  
   lot_id 及用地值。 */  
strcpy (wkb_sql,"insert into LOTS (lot_id,lot)  
values (?, db2gse.ST_MPolyFromWKB (cast(? as blob(1m)),  
db2gse.coordref(..srid(0)))");  
  
/* 為 SQL 陳述式 handle 配置記憶體，並結合此  
   陳述式 handle 與連接 handle。 */  
rc = SQLAllocStmt (handle, &hstmt);  
  
/* 準備要執行的 SQL 陳述式。 */  
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);  
  
/* 將 lot_id 整數值連結第一個參數。 */  
pcbvalue1 = 0;  
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,  
SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);  
  
/* 將用地形狀連結第二個參數。 */  
pcbvalue2 = lot_len;  
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,  
SQL_BLOB, lot_len, 0, lot_wkb, lot_len, &pcbvalue2);  
  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ST_NumGeometries

ST_NumGeometries 使用一個集合並傳回該集合中的幾何數目。

語法

```
db2gse.ST_NumGeometries(g db2gse.ST_GeomCollection)
```

回覆類型

整數

範例

都市工程師必須知道與每一個建築物覆蓋區相關的實際建築物數目。

建築物覆蓋區儲存在使用下列 CREATE TABLE 陳述式建立的 BUILDINGFOOTPRINTS 表格。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id      integer,
    footprint   db2gse.ST_MultiPolygon);
```

下列 SELECT 陳述式使用 ST_NumGeometries 函數列示 BUILDING_ID，該 ID 唯一識別每一棟建築物和每一個覆蓋區中的建築物數目。

```
SELECT building_id, db2gse.ST_NumGeometries (footprint) "Number of buildings"
FROM BUILDINGFOOTPRINTS;
```

ST_NumInteriorRing

ST_NumInteriorRing 使用一個多邊形並傳回它的內環數目。

語法

```
db2gse.ST_NumInteriorRing(p db2gse.ST_Polygon)
```

回覆類型

整數

範例

想要研究幾個南海島嶼上的鳥群的一位鳥類學家，知道某特定鳥類的哺育區侷限在含有淡水湖的島嶼。因此她想要知道哪些島嶼含有一個或多個湖泊。

下列 CREATE TABLE 陳述式建立 ISLANDS 表格。ISLANDS 表格的 ID 和 NAME 直欄識別每一個島嶼，LAND 多邊形直欄儲存島嶼的幾何。

```
CREATE TABLE ISLANDS (id integer, name varchar(32), land db2gse.ST_Polygon);
```

因為內環代表湖泊，所以使用 ST_NumInteriorRing 函數只列示至少有一個內環的島嶼。

```
SELECT name FROM ISLANDS WHERE db2gse.ST_NumInteriorRing(land) > 0;
```

ST_NumPoints

ST_NumPoints 使用線串並傳回它的點數目。

語法

```
db2gse.ST_NumPoints(l db2gse.ST_LineString)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立 NUMPOINTS_TEST 表格。 GEOTYPE 直欄包含儲存在 G1 幾何直欄的幾何類型。

```
CREATE TABLE NUMPOINTS_TEST (geotype varchar(12), g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式插入一個線串。

```
INSERT INTO NUMPOINTS_TEST VALUES( linestring,  
db2gse.ST_LineFromText('linestring (10.02 20.01, 23.73 21.92)',  
db2gse.coordref()..srid(0)))
```

下列 SELECT 陳述式和對應的結果集列示幾何類型和每一個幾何含有的點數目。

```
SELECT geotype, db2gse.ST_NumPoints(g1)  
FROM NUMPOINTS_TEST
```

```
GEOTYPE          Number of points  
-----  
ST_linestring    2  
1 record(s) selected.
```

ST_OrderingEquals

ST_OrderingEquals 比較兩個幾何並傳回 1 (TRUE) (若這兩個幾何相等而且座標使用相同次序)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_OrderingEquals(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

下列 CREATE TABLE 陳述式建立 LINESTRING_TEST 表格，它有兩個線串直欄 (L1 和 L2)。

```
CREATE TABLE LINESTRING_TEST (lid integer, l1 db2gse.ST_LineString, l2 db2gse.ST_LineString)
```

下列 INSERT 陳述式將兩個線串插入相等而且有相同座標排序的 L1 和 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)));
```

下列 INSERT 陳述式將兩個線串插入相等但沒有相同座標排序的 L1 和 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)));
```

如下列 SELECT 陳述式及結果集所顯示，ST_OrderingEquals 函數：

- 當提供給它作為輸入的幾何相等且其座標次序相同時，傳回 1 (TRUE)
- 當提供給它的幾何不相等，或當其中一項幾何的座標所遵循的次序與其它幾何的座標不同時，則傳回 0 (FALSE)

```
SELECT lid, db2gse.ST_OrderingEquals(l1,l2) OrderingEquals FROM linestring_test
lid      OrderingEquals
----      -
1          1
2          0
```

ST_Overlaps

ST_Overlaps 使用兩個幾何物件並傳回 1 (TRUE) (若這兩個物件的交集產生一個相同維度但不等於來源物件的幾何物件)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Overlaps(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

縣行政官需要 5 英哩半徑範圍重疊敏感區域的危險廢棄物場所列示。

下列 CREATE TABLE 陳述式建立 SENSITIVE_AREAS 表格。除了儲存機構的多邊形幾何的 ZONE 直欄以外，SENSITIVE_AREAS 表格還包含數個直欄，說明受威脅的機構。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                                name        varchar(128),
                                size        float,
                                type        varchar(10),
                                zone        db2gse.ST_Polygon);
```

HAZARDOUS_SITES 表格在 SITE_ID 及 NAME 直欄中儲存這些場所的 ID，同時每一個場所的實際地理位置會儲存在 LOCATION 點直欄中。

```
CREATE TABLE HAZARDOUS_SITES (site_id   integer,
                                name       varchar(128),
                                location   db2gse.ST_Point);
```

在下列 SELECT 陳述式中，ST_Overlaps 函數合併 SENSITIVE_AREAS 與 HAZARDOUS_SITES 表格。若 SENSITIVE_AREAS 表格中列的區域多邊形重疊 HAZARDOUS_SITES 位置點的緩衝的 5 英哩半徑範圍，對所有這些列，它傳回 1 (TRUE)。

```
SELECT hs.name
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa
WHERE db2gse.ST_Overlaps (buffer(hs.location,(5 * 5280)),sa.zone) = 1;
```

在第263頁的圖37中，醫院和學校重疊縣的兩個危險廢棄物場所的 5 英哩半徑範圍，療養院沒有重疊。

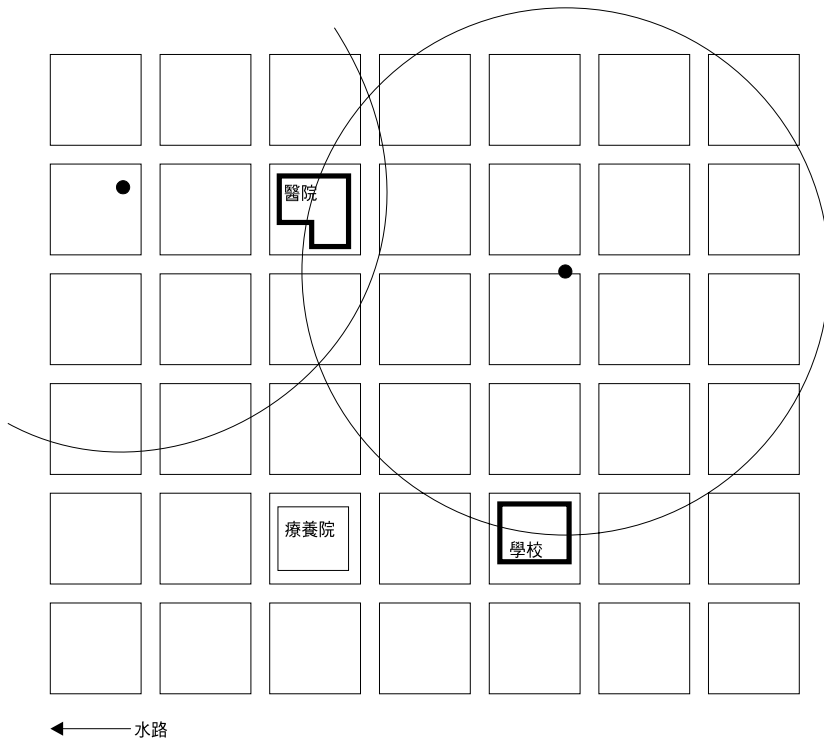


圖 37. 使用 `ST_Overlaps` 判斷至少有一部份是在危險廢棄物區域內的建築物。

ST_Perimeter

ST_Perimeter 傳回多邊形的周長。

語法

```
db2gse.ST_Perimeter(p db2gse.ST_Polygon)
```

回覆類型

Double

範例

研究海岸線鳥類的一位鳥類學家需要判斷某特定區域內的湖泊的湖濱線。湖泊會以多邊形儲存在以下列 CREATE TABLE 陳述式所建立的 WATERBODIES 表格中。

```
CREATE TABLE WATERBODIES (wbid integer,  
                             waterbody db2gse.ST_MultiPolygon);
```

在下列 SELECT 陳述式中，ST_Perimeter 函數傳回每一個湖泊的周長，SUM 函數合計周長並傳回它們的總計。

```
SELECT SUM(db2gse.ST_Perimeter(waterbody))  
FROM waterbodies;
```

ST_Point

ST_Point 傳回 ST_Point，提供 x 座標、y 座標和空間參照。

語法

```
db2gse.ST_Point(X Double, Y Double, SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列 CREATE TABLE 陳述式建立 POINT_TEST 表格，它有單一點直欄 PT1。

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

ST_Point 函數將點座標轉換成點幾何之後，INSERT 陳述式才將它插入 PT1 直欄。

```
INSERT INTO point_test VALUES(  
    db2gse.ST_Point(10.01,20.03, db2gse.coordref()..srid(0))  
)
```

ST_PointFromText

ST_PointFromText 採用點類型的知名文字表示法及空間參照系統 ID，然後傳回點。

語法

```
db2gse.ST_PointFromText(pointTaggedText Varchar(4000), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列 CREATE TABLE 陳述式建立 POINT_TEST 表格，它有單一點直欄 PT1。

```
CREATE TABLE POINT_TEST (pt1 db2gse.ST_Point)
```

INSERT 陳述式將點插入 PT1 直欄之前，ST_PointFromText 函數會將點文字座標轉換成點格式。

```
INSERT INTO POINT_TEST VALUES (  
    db2gse.ST_PointFromText ('point(10.01 20.03)',  
    db2gse.coordref()  
    ..srid(0))
```

ST_PointFromWKB

ST_PointFromWKB 採用點類型的知名二進位表示法及空間參照系統 ID，以傳回點。

語法

```
db2gse.ST_PointFromWKB(WKBPoint Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列程式碼片斷移入 HAZARDOUS_SITES 表格。

危險場所儲存在使用下列 CREATE TABLE 陳述式建立的 HAZARDOUS_SITES 表格。LOCATION 直欄 (定義成一個點) 儲存一個位置，這位置是每一個危險場所的地理中心。

```
CREATE TABLE HAZARDOUS_SITES (site_id integer,
                               name      varchar(128),
                               location  db2gse.ST_Point);

/* 建立 SQL insert 陳述式，以移入 site_id、名稱及
   位置。問號是參數記號，可指出
   執行期將擷取的 site_id、名稱及位置值。 */
strcpy (wkb_sql, "insert into HAZARDOUS_SITES (site_id, name, location)
values (?, ?, db2gse.ST_PointFromWKB(cast(? as blob(1m)),
db2gse.coordref(..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 將 site_id 整數值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &site_id, 0, &pcbvalue1);

/* 將 VARCHAR 名稱值連結第二個參數。 */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* 將位置形狀連結第三個參數。 */
pcbvalue3 = location_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
```

```
        SQL_BLOB, location_len, 0, location_wkb, location_len, &pcbvalue3);  
/* 執行 insert 陳述式。 */  
rc = SQLExecute (hstmt);
```

ST_PointN

ST_PointN 使用線串和整數指標並傳回一個點，這個點是此線串路徑中的第 n 個頂點。

語法

```
db2gse.ST_PointN(l db2gse.ST_Curve, n Integer)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列 CREATE TABLE 陳述式建立 POINTN_TEST 表格，它有兩個直欄：一個是唯一識別每一列的 GID 直欄，一個是 LN1 線串直欄。

```
CREATE TABLE POINTN_TEST (gid integer, ln1 db2gse.ST_LineString)
```

下列 INSERT 陳述式插入兩個線串值。第一個線串沒有 Z 座標或測量，第二個線串有 Z 座標和測量。

```
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)))
```

下列 SELECT 陳述式和對應的結果集列示 GID 直欄和每一個線串的第二個頂點。第一列產生一個點但沒有 Z 座標或測量，第二列產生一個含有 Z 座標和測量的點。ST_PointN 函數傳回一些含有 Z 座標或測量的點 (若這些點存在於來源線串)。

```
SELECT gid, CAST(db2gse.ST_AsText(db2gse.ST_PointN(ln1,2)) AS varchar(60)) "The 2nd vertex"
FROM POINTN_TEST
```

```
GID          The 2nd vertex
-----
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 採用多邊形或多重多邊形，並傳回 ST_Point。

語法

```
db2gse.ST_PointOnSurface(s db2gse.ST_Surface)
db2gse.ST_PointOnSurface(ms db2gse.ST_MultiSurface)
```

回覆類型

db2gse.ST_Point

範例

都市工程師需要建立每一個建築物覆蓋區的標籤點。

建築物覆蓋區儲存在使用下列 CREATE TABLE 陳述式建立的 BUILDINGFOOTPRINTS 表格。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

ST_PointOnSurface 函數產生一個點，該點一定在建築物覆蓋區表面。

```
SELECT db2gse.ST_PointOnSurface(footprint) FROM BUILDINGFOOTPRINTS;
```

ST_PolyFromText

ST_PolyFromText 採用多邊形類型的知名文字表示法及空間參照系統 ID，然後傳回多邊形。

語法

```
db2gse.ST_PolyFromText(polygonTaggedText Varchar(4000), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Polygon
```

範例

下列 CREATE TABLE 陳述式建立具有單一多邊形直欄的 POLYGON_TEST 表格。

```
CREATE TABLE POLYGON_TEST (p11 db2gse.ST_Polygon)
```

下列 INSERT 陳述式使用 ST_PolyFromText 函數將多邊形插入多邊形直欄。

```
INSERT INTO POLYGON_TEST VALUES (db2gse.ST_PolyFromText(  
'polygon((10.01 20.03,10.52 40.11,30.29  
41.56, 31.78 10.74,10.01 20.03))',  
db2gse.coordref(..srid(0)))
```

ST_PolyFromWKB

ST_PolyFromWKB 採用多邊形類型的知名二進位表示法及空間參照系統 ID，然後傳回多邊形。

語法

```
db2gse.ST_PolyFromWKB(WKBPolygon Blob(1M), SRID db2gse.coordref)
```

回覆類型

```
db2gse.ST_Polygon
```

範例

下列程式碼片斷移入 SENSITIVE_AREAS 表格。

除了儲存機構的多邊形幾何的 zone 直欄以外，SENSITIVE_AREAS 表格還包含一些直欄，說明受威脅的機構。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);

/* 建立 SQL insert 陳述式，以移入 id、name、size、type 及
   zone。問號是參數記號，可指出執行期將擷取的
   id、name、size、type 及 zone 值。 */
strcpy (shp_wkb,"insert into SENSITIVE_AREAS (id, name, size, type, zone)
values (?, ?, ?, ?, db2gse.ST_PolyFromWKB (cast(? as blob(1m)),
db2gse.coordref()..srid(0)))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */
rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */
rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 將 id 整數值連結第一個參數。 */
pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_INTEGER,
SQL_INTEGER, 0, 0, &id, 0, &pcbvalue1);

/* 將 VARCHAR 名稱值連結第二個參數。 */
pcbvalue2 = name_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_CHAR,
SQL_CHAR, 0, 0, name, 0, &pcbvalue2);

/* 將 site 浮點連結第三個參數。 */
pcbvalue3 = 0;
```

```

rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_FLOAT,
    SQL_REAL, 0, 0, &size, 0, &pcbvalue3);

/* 將 varchar 類型連結第四個參數。 */
pcbvalue4 = type_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_CHAR,
    SQL_VARCHAR, type_len, 0, type, type_len, &pcbvalue4);

/* 將多邊形區域連結第五個參數。 */
pcbvalue5 = zone_len;
rc = SQLBindParameter (hstmt, 3, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, zone_len, 0, zone_wkb, zone_len, &pcbvalue5);

/* 執行 insert 陳述式。 */
rc = SQLExecute (hstmt);

```

ST_Polygon

ST_Polygon 會從 ST_LineString 中產生 ST_Polygon。

語法

```
db2gse.ST_Polygon(l db2gse.ST_LineString)
```

回覆類型

```
db2gse.ST_Polygon
```

範例

下列 CREATE TABLE 陳述式建立 POLYGON_TEST 表格，它有單一直欄 P1。

```
CREATE TABLE POLYGON_TEST (p1 db2gse.ST_polygon)
```

下列 INSERT 陳述式將一個環 (是封閉和單純的線串) 轉換成一個多邊形，然後使用 ST_Polygon 函數中的 ST_LineFromText 函數將該多邊形插入 P1 直欄。

```
INSERT INTO POLYGON_TEST VALUES (  
db2gse.ST_Polygon(db2gse.ST_LineFromText('linestring(10.01 20.03,20.94  
21.34,35.93 10.04,10.01 20.03)', db2gse.coordref()..srid(0)))  
)
```


ST_Relate

ST_Relate 比較兩個幾何並傳回 1 (TRUE) (若這兩個幾何符合 DE-9IM 型樣矩陣字串)；否則會傳回 0 (FALSE)。關於 DE-9IM 型樣矩陣的資訊，請參閱第144頁的『述詞函數』。

語法

```
db2gse.ST_Relate(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry, patternMatrix CHAR(9))
```

回覆類型

整數

範例

DE-9IM 型樣矩陣是比較幾何的裝置。這類矩陣有幾種類型。例如，若有兩個幾何相等，等於型樣矩陣會通知您。

在本範例中，顯示在表57的等於型樣矩陣，它是由左向右和由上向下讀取到字串(『T**F**FFF*』)。

表 57. 等於型樣矩陣

		內部	b 界限	外部
a	內部	T	*	F
	界限	*	*	F
	外部	F	F	*

下一步，使用下列 CREATE TABLE 陳述式建立 RELATE_TEST 表格。

```
CREATE TABLE RELATE_TEST (rid integer, g1 db2gse.ST_Geometry,  
g2 db2gse.ST_Geometry, g3 db2gse.ST_Geometry);
```

下列 INSERT 陳述式將範例次類別插入 RELATE_TEST 表格。

```
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)  
)
```

下列 `SELECT` 陳述式和對應的結果集列示儲存在 `GEOTYPE` 直欄 (具有該 `geotype` 的維度) 的次類別名稱。

```
SELECT rid, relate(g1,g2, 'T**FFF*') equals FROM relate_test
```

RID	equals
1	1

1 record(s) selected.

ST_SRID

ST_SRID 採用幾何物件並傳回其空間參照系統 ID。

語法

```
db2gse.ST_SRID(g1 db2gse.ST_Geometry)
```

回覆類型

整數

範例

ST_SRID 函數會傳回與 ST_Geometry 值相關聯的空間參照系統的 ID。

例如，一個幾何類型是用於 CREATE TABLE 陳述式：

```
CREATE TABLE SRID_TEST(g1 db2gse.ST_Geometry)
```

在下列 INSERT 陳述式中，位於座標 10.01,50.76 的一個點幾何會插入幾何直欄 G1。ST_PointFromText 函數建立點幾何時，它指定 srid 值 1。

```
INSERT INTO SRID_TEST  
VALUES (db2gse.ST_PointFromText('point(10.01 50.76)',  
db2gse.coordref().srid(0)))
```

ST_SRID 函數會傳回剛剛輸入的幾何之空間參照系統 ID，如下列 SELECT 陳述式及相對應結果集的說明。

```
SELECT db2gse.ST_SRID(g1) FROM SRID_TEST
```

```
g1
```

```
-----  
0
```

ST_StartPoint

ST_StartPoint 使用一個線串並傳回一個點，該點是此線串的第一個點。

語法

```
db2gse.ST_StartPoint(c db2gse.ST_Curve)
```

回覆類型

```
db2gse.ST_Point
```

範例

下列 CREATE TABLE 陳述式建立 STARTPOINT_TEST 表格。STARTPOINT_TEST 有兩個直欄：GID 整數直欄 (它唯一識別表格的列) 和 LN1 線串直欄。

```
CREATE TABLE STARTPOINT_TEST (gid integer, ln1 db2gse.ST_LineString)
```

下列 INSERT 陳述式將線串插入 LN1 直欄。第一個線串沒有 Z 座標或測量，第二個線串有 Z 座標和測量。

```
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)))
```

下列 SELECT 陳述式和對應的結果集顯示 ST_StartPoint 函數如何取出每一線串的第一個點。ST_AsText 函數將此點轉換成它的文字格式。列示中的第一個點沒有 Z 座標或測量，第二個點有 Z 座標和測量，因為來源線串有這兩者。

```
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 使用兩個幾何物件並傳回一個幾何物件，該物件是來源物件的對稱差集。

ST_SymmetricDiff 函數會傳回具有相同維度的兩個交集幾何間的對稱差異（空間的 Boolean 邏輯 XOR）。如果這些幾何是相等的，則 ST_SymmetricDiff 會傳回空幾何。如果它們不相等，則其中之一或兩者的一部份會超出交集區域。ST_SymmetricDiff 會以集合方式傳回非交集部份；例如，以多重多邊形。

如果已給定 ST_SymmetricDiff 不同維度的幾何作為輸入，則會傳回 NULL。

語法

```
db2gse.ST_SymmetricDiff(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

縣行政官必須判斷敏感區域與 5 英哩危險場所半徑範圍沒有相交的區域。

下列 CREATE TABLE 陳述式建立 SENSITIVE_AREAS 表格，它包含一些說明受威脅機構的直欄。SENSITIVE_AREAS 表格也包含 ZONE 直欄，它儲存機構的多邊形幾何。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

下列 CREATE TABLE 陳述式會建立 HAZARDOUS_SITES 表格，在 SITE_ID 及 NAME 直欄中儲存場所的 ID，同時會在 LOCATION 點直欄中儲存每一個場所的實際地理位置。

```
CREATE TABLE HAZARDOUS_SITES (site_id  integer,
                               name      varchar(128),
                               location  point);
```

ST_Buffer 函數產生圍繞危險廢棄物場所的 5 英哩緩衝區。ST_SymmetricDiff 函數從緩衝的危險廢棄物場所多邊形與敏感區域的交集產生多邊形。ST_Area 函數傳回每一個危險場所的相交多邊形區域。

```
SELECT sa.name, hs.name,
       db2gse.ST_Area(db2gse.ST_SymmetricDiff (db2gse.ST_Buffer(hs.location,(5 * 5280)),
FROM HAZARDOUS_SITES hs, SENSITIVE_AREAS sa
```

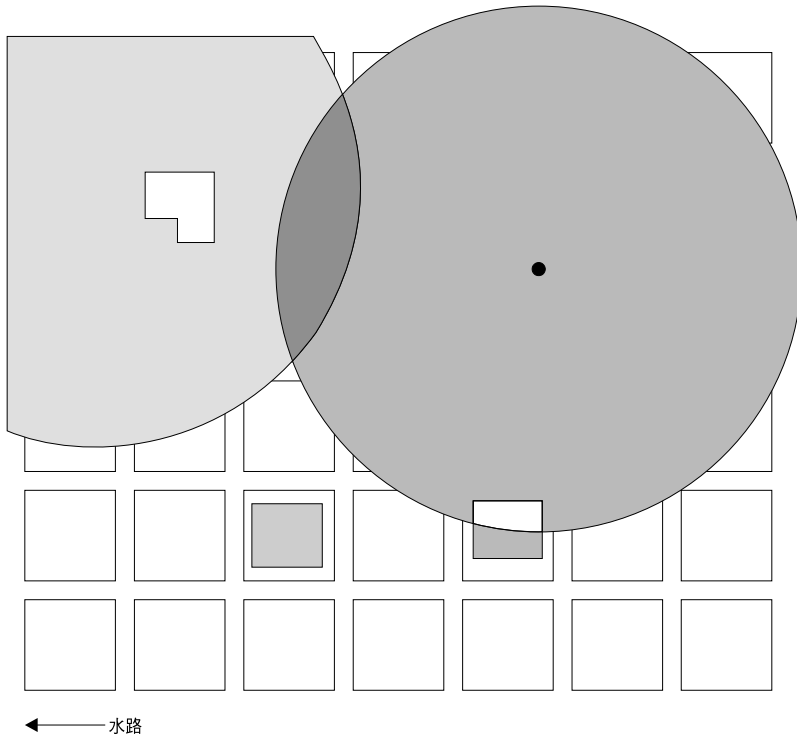


圖 38. 使用 *ST_SymmetricDiff* 判斷不含敏感區域 (有人居住的建築物) 的危險廢棄物區域。

在圖38中，危險廢棄物場所與敏感區域的對稱差集會減少相交區域。

ST_Touches

若兩個幾何相交兩個幾何的內部沒有共同點，ST_Touches 就會傳回 1 (TRUE)；否則會傳回 0 (FALSE)。至少一個幾何必須是線串、多邊形、多線串或多重多邊形。

語法

```
db2gse.ST_Touches(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

GIS 技術人員需要提供終點與另一條水溝線交集的全部水溝線列示。

下列 CREATE TABLE 陳述式建立 SEWERLINES 表格，該表格有三個直欄。第一欄 SEWER_ID 唯一定義每一個水溝線。整數類型的第二個直欄 CLASS 識別水溝線類型，該類型通常與水溝線容量有關。第三欄 SEWER (線串類型) 儲存水溝線的幾何。

```
CREATE TABLE SEWERLINES (sewer_id integer, class integer, sewer db2gse.ST_LineString);
```

下列 SELECT 陳述式傳回彼此接觸的 SEWER_IDS 排序列示。

```
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 會將幾何關聯到目前幾何關聯的空間參照系統以外的空間參照系統。

語法

```
db2gse.ST_Transform(g db2gse.ST_Geometry, SRID db2gse.coordref)
```

回覆類型

db2gse.ST_Geometry

範例

下列 CREATE TABLE 陳述式建立 TRANSFORM_TEST 表格，該表格有兩個線串直欄 (L1 和 L2)。

```
CREATE TABLE TRANSFORM_TEST (tid integer, l1 db2gse.ST_LineString, l2 db2gse.ST_LineString)
```

下列 INSERT 陳述式將線串插入具有 SRID 102 的 l1。

```
INSERT INTO TRANSFORM_TEST VALUES (1, db2gse.ST_LineFromText('linestring
(10.01 40.43, 92.32 29.89)',
db2gse.coordref()..srid(102)),NULL)
```

ST_Transform 函數從指定給 SRID 102 的座標參照將 l1 線串轉換成指定給 SRID 105 的座標參照。下列 UPDATE 陳述式在直欄 l2 儲存轉換的線串。

```
UPDATE TRANSFORM_TEST SET l2 = db2gse.ST_Transform(l1, db2gse.coordref()..srid(105))
```

如果指定的幾何是落在新空間參照系統下的座標系統外，則 ST_Transform 會以 NULL 傳回幾何。

例如，請思考其 X 座標為 10.01 且 Y 座標為 20.02 的 ST_Point 幾何。假設此幾何已被指定給具有下列參數的空間參照系統：

falsex	0
falsey	0
xyunits	1

然後，假設您呼叫 ST_Transform 將 ST_Point 幾何的空間參照系統變更為具有下列參數的空間參照系統：

falsex	100
falsey	100
xyunits	1

ST_Transform 會以 NULL 傳回幾何，因為座標 (10.01, 20.02) 落在新空間參照系統下的座標系統外。

ST_Union

ST_Union 使用兩個幾何物件並傳回一個幾何物件，該物件是來源物件的聯集。

語法

```
db2gse.ST_Union(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 CREATE TABLE 陳述式建立 SENSITIVE_AREAS 表格，它包含一些說明受威脅機構的直欄。 SENSITIVE_AREAS 表格也包含 ZONE 直欄，它儲存機構的多邊形幾何。

```
CREATE TABLE SENSITIVE_AREAS (id          integer,
                               name        varchar(128),
                               size        float,
                               type        varchar(10),
                               zone        db2gse.ST_Polygon);
```

下列 CREATE TABLE 陳述式會建立 HAZARDOUS_SITES 表格，在 SITE_ID 及 NAME 直欄中儲存場所的 ID。每一個場所的實際地理位置儲存在 LOCATION 點直欄。

```
CREATE TABLE HAZARDOUS_SITES (site_id integer, name varchar(128),
                               location db2gse.ST_Point);
```

下列 SELECT 陳述式使用 ST_Buffer 函數產生圍繞危險廢棄物場所的 5 英哩緩衝區。 ST_Union 函數從緩衝的危險廢棄物場所多邊形與敏感區域的聯集產生多邊形。 ST_Area 函數傳回多邊形區域的聯集。

```
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 使用兩個幾何物件並傳回 1 (TRUE) (若第二個物件包含第一個物件)；否則會傳回 0 (FALSE)。

語法

```
db2gse.ST_Within(g1 db2gse.ST_Geometry, g2 db2gse.ST_Geometry)
```

回覆類型

整數

範例

在下例中會建立兩個表格。第一個表格 BUILDINGFOOTPRINTS 含有都市的建築物覆蓋區。第二個表格 LOTS 含有都市的用地。都市工程師想要確定全部建築物覆蓋區都在它們的用地內。

在這兩個表格中，多重多邊形資料類型儲存建築物覆蓋區和用地的幾何。資料庫設計者為這兩個形狀選取多重多邊形，因為用地可以被自然地形 (如河流隔開)，建築物覆蓋區通常可以由一些建築物組成。

```
CREATE TABLE BUILDINGFOOTPRINTS (
    building_id integer,
    lot_id       integer,
    footprint    db2gse.ST_MultiPolygon);
```

```
CREATE TABLE LOTS ( lot_id integer, lot db2gse.ST_MultiPolygon );
```

使用下列 SELECT 陳述式時，都市工程師首先選取不完全位於一塊用地的建築物。

```
SELECT building_id
   FROM BUILDINGFOOTPRINTS, LOTS
  WHERE db2gse.ST_Within(footprint,lot) ≠ 1;
```

雖然第一個查詢會提供覆蓋區在用地多邊形外面的全部 BUILDING_ID 列示，但它不會判斷是否有將正確 lot_id 指定給其餘覆蓋區。第二個 SELECT 陳述式則對 BUILDINGFOOTPRINTS 表格的 LOT_ID 直欄執行資料完整性檢查。

```
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 建構 ST_Geometry 值，該值提供它的知名二進位表示。自動使用 SRID 值 0。

語法

```
db2gse.ST_WKBTToSQL(WKBGeometry Blob(1M))
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 CREATE TABLE 陳述式建立 LOTS 表格，該表格有兩個直欄： LOT_ID 直欄 (它唯一識別每一塊用地) 和 LOT 多重多邊形直欄 (它含有每一塊用地的幾何)。

```
CREATE TABLE lots (lot_id integer,
                   lot      db2gse.ST_MultiPolygon);
```

下列 C 程式碼片斷包含 Spatial Extender SQL 函數所內含的 ODBC 函數，這些函數在 LOTS 表格中插入資料。

ST_WKBTToSQL 函數將 WKB 表示轉換成 Spatial Extender 幾何。整個 INSERT 陳述式被複製到 wkb_sql 字串。INSERT 陳述式含有一些參數記號以動態接受 LOT_ID 資料和 LOT 資料。

```
/* 建立 SQL INSERT 陳述式以移入用地 id 及
   用地多邊形。問號是參數記號，
   可指出執行期將擷取的 lot_id 及
   用地值。 */
strcpy (wkb_sql,"insert into lots (lot_id, lot)
        values(?, db2gse.ST_WKBTToSQL(cast(? as blob(1m))))");

/* 為 SQL 陳述式 handle 配置記憶體，並結合此
   陳述式 handle 與連接 handle。 */

rc = SQLAllocStmt (handle, &hstmt);

/* 準備要執行的 SQL 陳述式。 */

rc = SQLPrepare (hstmt, (unsigned char *)wkb_sql, SQL_NTS);

/* 將整數索引鍵值連結第一個參數。 */

pcbvalue1 = 0;
rc = SQLBindParameter (hstmt, 1, SQL_PARAM_INPUT, SQL_C_SLONG,
                      SQL_INTEGER, 0, 0, &lot_id, 0, &pcbvalue1);

/* 將形狀連結第二個參數。 */
```

```
pcbvalue2 = blob_len;
rc = SQLBindParameter (hstmt, 2, SQL_PARAM_INPUT, SQL_C_BINARY,
    SQL_BLOB, blob_len, 0, shape_blob, blob_len, &pcbvalue2);

/* 執行 insert 陳述式。 */

rc = SQLExecute (hstmt);
```

ST_WKTTToSQL

ST_WKTTToSQL 建構 ST_Geometry 值，該值提供它的知名文字表示法。自動使用 SRID 值 0。

語法

```
db2gse.ST_WKTTToSQL(geometryTaggedText Varchar(4000))
```

回覆類型

```
db2gse.ST_Geometry
```

範例

下列 CREATE TABLE 陳述式建立 GEOMETRY_TEST 表格，該表格包含兩個直欄：整數類型的 GID 直欄（它唯一識別每一列）和 G1 直欄（它儲存幾何）。

```
CREATE TABLE GEOMETRY_TEST (gid smallint, g1 db2gse.ST_Geometry)
```

下列 INSERT 陳述式將資料插入 GEOMETRY_TEST 表格的 GID 和 G1 直欄。ST_WKTTToSQL 函數將幾何的文字表示法轉換成對應的 Spatial Extender 個體化次類別。

```
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 使用一個點並傳回該點的 X 座標。

語法

ST_X(p ST_Point)

回覆類型

Double

範例

下列 CREATE TABLE 陳述式建立 X_TEST 表格，該表格有兩個直欄：GID 直欄 (它唯一識別橫列) 和 PT1 點直欄。

```
CREATE TABLE X_TEST (gid integer, pt1 db2gse.ST_Point)
```

下列 INSERT 陳述式插入兩列。一列是一個無 Z 座標或測量的點。另一個直欄有 Z 座標和測量。

```
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)))
```

下列 SELECT 陳述式和對應的結果集列示 GID 直欄和點的倍整數 X 座標。

```
SELECT gid, db2gse.ST_X(pt1) "The X coordinate" FROM X_TEST
```

GID	The X coordinate
1	+1.0020000000000000E+001
2	+1.0020000000000000E+001

2 record(s) selected.

ST_Y

ST_Y 使用一個點並傳回該點的 Y 座標。

語法

```
db2gse.ST_Y(p db2gse.ST_Point)
```

回覆類型

Double

範例

下列 CREATE TABLE 陳述式建立 Y_TEST 表格，該表格有兩個直欄：GID 直欄（它唯一識別橫列）和 PT1 點直欄。

```
CREATE TABLE Y_TEST (gid integer, pt1 db2gse.ST_Point)
```

INSERT 陳述式插入兩列。一列是一個無 Z 座標或測量的點。另一個直欄有 Z 座標和測量。

```
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)))
```

下列 SELECT 陳述式和對應的結果集列示 GID 直欄和點的倍整數 Y 座標。

```
SELECT gid, db2gse.ST_Y(pt1) "The Y coordinate" FROM Y_TEST
```

GID	The Y coordinate
1	+2.0010000000000000E+001
2	+2.0010000000000000E+001

2 record(s) selected.

Z

Z 使用一個點並傳回該點的 Z 座標。

語法

```
Z(p db2gse.ST_Point)
```

回覆類型

Double

範例

下列 CREATE TABLE 陳述式建立 Z_TEST 表格，該表格有兩個直欄：GID 直欄 (唯一識別橫列) 和 PT1 點直欄。

```
CREATE TABLE Z_TEST (gid integer, pt1 db2gse.ST_Point)
```

下列 INSERT 陳述式插入兩列。一列是一個無 Z 座標或測量的點。另一個直欄有 Z 座標和測量。

```
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)))
```

下列 SELECT 陳述式和對應的結果集列示 GID 直欄和點的倍整數 Z 座標。第一列是 NULL，因為點

```
SELECT gid, db2gse.Z(pt1) "The Z coordinate" FROM Z_TEST
```

GID	The Z coordinate
1	-
2	+5.0000000000000000E+000

2 record(s) selected.

第15章 座標系統

本章提供的參考資料是關於空間參照系統 (SRS) 和用來解譯空間資料的座標值。

- 『座標系統的概觀』
- 第295頁的『支援的長度單位』
- 第295頁的『支援的角度單位』
- 第296頁的『支援的旋轉橢圓體』
- 第297頁的『支援的大地測量基準點』
- 第299頁的『支援的中央子午線』
- 第300頁的『支援的地圖投影』
- 第301頁的『支援的圓錐投影』
- 第301頁的『支援的方位或平面投影』
- 第301頁的『支援的地圖投影參數』

座標系統的概觀

空間參照系統的知名文字表示法，以標準的文字表示法來提供空間參照系統資訊。知名文字表示法的定義，是做效 Petrotechnical Open Software Corporation/European Professional Surveyors Group (POSC/EPSG) 座標系統資料模型。

空間參照系統是一種地理的 (經緯度)、投影的 (X,Y) 或以地球為中心的 (X,Y,Z) 座標系統。座標系統由數個物件所組成。每一個物件有一個大寫的關鍵字 (例如，DATUM 或 UNIT)，後面接著以括弧括住並以逗號區隔的物件參數。有些物件內另外包含其它物件，所以結果是巢狀結構。

註： 施行時可使用標準括弧 () 來代替方括弧 []，但要能夠讀取這兩種括弧。

使用方括弧的座標系統字串表示法的 EBNF (Extended Backus Naur Form) 定義如下 (請參閱上述的方括弧用法)：

```
<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>
```

若資料以投影座標表示，則資料集的座標系統由 PROJCS 關鍵字來識別 (若以地理座標表示，則由 GEOGCS 識別，若以地心座標表示，則由 GEOCCS 來識別)。PROJCS 關鍵字後面接著投影座標系統的所有定義「元素」。任何物件的第一個元素一律為名稱。投影座標系統名稱後面接著數個物件：地理座標系統、地圖投影、一或多個參數、以及長度測量單位。所有投影座標系統皆以地理座標系統為基礎，所以本節首先說明投影座標系統的相關元素。例如，NAD83 資料項的 UTM 區域 10N 定義如下：

```
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]]
```

名稱和數個物件依序定義下列的地理座標系統物件：資料項、中央子午線、及角度測量單位。

```
<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 是以公尺為測量單位，必須為 > 0。)  
<inverse flattening> = <number>  
<prime meridian> = PRIMEM["<name>", <longitude>]  
<longitude> = <number>
```

NAD83 的 UTM 區域的座標系統字串：

```
GEOGCS["GCS_North_American_1983",  
DATUM["D_North_American_1983",  
SPHEROID["GRS_1980",6378137,298.257222101]],  
PRIMEM["Greenwich",0],  
UNIT["Degree",0.0174532925199433]]
```

UNIT 物件可代表角度或長度測量單位：

```
<angular unit> = <unit>  
<linear unit> = <unit>  
<unit> = UNIT["<name>", <conversion factor>]  
<conversion factor> = <number>
```

轉換因數指定每單位的公尺數 (應用於長度單位) 或角度數，必須大於零。

UTM Zone 10N 的完整字串表示法如下：

```

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]]

```

地心座標系統和地理座標系統非常類似：

<geocentric cs> = GEOCCS["<name>", <datum>, <prime meridian>, <linear unit>]

支援的長度單位

表 58. 支援的長度單位

單位	轉換因數
公尺	1.0
英尺 (國際)	0.3048
美制英尺	12/39.37
已修改的美制英尺	12.0004584/39.37
Clarke 英尺	12/39.370432
印度制英尺	12/39.370141
令	7.92/39.370432
令 (Benoit)	7.92/39.370113
令 (Sears)	7.92/39.370147
測鏈 (Benoit)	792/39.370113
測鏈 (Sears)	792/39.370147
碼 (印度制)	36/39.370141
碼 (Sears)	36/39.370147
尋	1.8288
海涅	1852.0

支援的角度單位

表 59. 支援的角度單位

單位	轉換因數
徑	1.0
小數度	$\pi/180$

表 59. 支援的角度單位 (繼續)

單位	轉換因數
小數分	$(\pi/180)/60$
小數秒	$(\pi/180)/36000$
Gon	$\pi/200$
Grad	$\pi/200$

支援的旋轉橢圓體

表 60. 支援的旋轉橢圓體

名稱	半長軸	扁率
Airy	6377563.396	299.3249646
已修改的 Airy	6377340.189	299.3249646
Australian	6378160	298.25
Bessel	6377397.155	299.1528128
已修改的 Bessel	6377492.018	299.1528128
Bessel (Namibia)	6377483.865	299.1528128
Clarke 1866	6378206.4	294.9786982
Clarke 1866 (Michigan)	6378693.704	294.978684677
Clarke 1880	6378249.145	293.465
Clarke 1880 (Arc)	6378249.145	293.466307656
Clarke 1880 (Benoit)	6378300.79	293.466234571
Clarke 1880 (IGN)	6378249.2	293.46602
Clarke 1880 (RGS)	6378249.145	293.465
Clarke 1880 (SGA)	6378249.2	293.46598
Everest 1830	6377276.345	300.8017
Everest 1975	6377301.243	300.8017
Everest (Sarawak 和 Sabah)	6377298.556	300.8017
已修改的 Everest 1948	6377304.063	300.8017
Fischer 1960	6378166	298.3
Fischer 1968	6378150	298.3
已修改的 Fischer (1960)	6378155	298.3
GEM10C	6378137	298.257222101
GRS 1980	6378137	298.257222101

表 60. 支援的旋轉橢圓體 (繼續)

名稱	半長軸	扁率
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
已修改的 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
球體 (半徑 = 1.0)	1	0
球體 (半徑 = 6371000 m)	6371000	0
球體 (半徑 = 6370997 m)	6370997	0
Struve 1860	6378297	294.73
Walbeck	6376896	302.78
War Office	6378300.583	296
WGS 1960	6378165	298.3
WGS 1966	6378145	298.25
WGS 1972	6378135	298.26
WGS 1984	6378137	298.257223563

支援的大地測量基準點

表 61. 支援的大地測量基準點

支援的大地測量基準點	
Adindan	里斯本
Afgooye	Loma Quintana

表 61. 支援的大地測量基準點 (繼續)

支援的大地測量基準點	
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
波哥大	Nord de Guerre
Bukit Rimpah	NGO 1948
Camacupa	Nord Sahara 1959
Campo Inchauspe	NSWC 9Z-2
Cape	Nouvelle Triangulation Francaise
Carthage	New Zealand Geodetic Datum 1949
Chua	OS (SN) 1980
Conakry 1905	OSGB 1936
Corrego Alegre	OSGB 1970 (SN)
Cote d'Ivoire	Padang 1884
Datum 73	Palestine 1923
Deir ez Zor	Pointe Noire
Deutsche Hauptdreiecksnetz	Provisional South American Datum 1956
Douala	Pulkovo 1942

表 61. 支援的大地測量基準點 (繼續)

支援的大地測量基準點	
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
Leigon	Yoff
Liberia 1964	Zanderij

支援的中央子午線

表 62. 支援的中央子午線

位置	座標
格林威治	0° 0' 0"

表 62. 支援的中央子午線 (繼續)

位置	座標
伯恩	7° 26' 22.5" E
波哥大	74° 4' 51.3" W
布魯塞爾	4° 22' 4.71" E
Ferro	17° 40' 0" W
雅加達	106° 48' 27.79" E
里斯本	9° 7' 54.862" W
馬德里	3° 41' 16.58" W
巴黎	2° 20' 14.025" E
羅馬	12° 27' 8.4" E
斯德哥爾摩	18° 3' 29" E

支援的地圖投影

表 63. 支援的地圖投影

圓柱投影	偽圓柱投影
Behrmann	Craster 拋物線
Cassini	Eckert I
圓柱等積	Eckert II
等矩形	Eckert III
Gall 立體圖	Eckert IV
Gauss-Kruger	Eckert V
麥卡托	Eckert VI
米勒圓柱	McBryde-Thomas 平面極線四次式
斜面	麥卡托 (Hotine) 摩爾威特
Plate-Carée	Robinson
時間	正弦曲線 (Sansom-Flamsteed)
橫軸麥卡托	Winkel I

支援的圓錐投影

表 64. 支援的圓錐投影

支援的圓錐投影	
Albers 圓錐等積	Chamberlin 三度
Bipolar 斜面等圓錐	兩點等距
Bonne	Hammer-Aitoff 等積
等距圓錐	Van der Grinten I
蘭伯特等角圓錐	雜項
多圓錐	Alaska 系列 E
簡單圓錐	Alaska Grid (由 Snyder 修改的立體圖)

支援的方位或平面投影

- 方位等距
- 一般垂直近邊透視
- 日晷儀
- 蘭伯特方位等積
- 正交
- 極線立體圖
- 立體圖

支援的地圖投影參數

表 65. 支援的地圖投影參數

參數	說明
central_meridian	做為 x 座標原點的經線。
scale_factor	通常用來減少地圖投影的扭曲程度。
standard_parallel_1	通常不會扭曲的緯線。亦表示「實際比例的緯線」。
standard_parallel_2	通常不會扭曲的經線。
longitude_of_center	定義地圖投影中心點的經線。
latitude_of_center	定義地圖投影中心點的緯線。
longitude_of_origin	做為 x 座標原點的經線。
latitude_of_origin	做為 y 座標原點的緯線。

表 65. 支援的地圖投影參數 (繼續)

參數	說明
false_easting	加到 x 座標。用來提供正值。
false_northing	加到 y 座標。用來提供正值。
azimuth	定義斜面投影中心線的西北角。
longitude_of_point_1	地圖投影所需第一點的經度。
latitude_of_point_1	地圖投影所需第一點的緯度。
longitude_of_point_2	地圖投影所需第二點的經度。
latitude_of_point_2	地圖投影所需第二點的緯度。
longitude_of_point_3	地圖投影所需第三點的經度。
latitude_of_point_3	地圖投影所需第三點的緯度。
landsat_number	Landsat 衛星的編號。
path_number	特定衛星的軌道編號。
perspective_point_height	地圖投影透視點地面以上的高度。
fipszone	State Plane Coordinate System 區域編號。
zone	UTM 區域編號。

第16章 空間資料的檔案格式

本章說明 Spatial Extender 知名表示法。將這些表示法形容成知名，因為它們是由 Open GIS Consortium (OGC) 所定義，而且不是 Spatial Extender 特有的。對於瞭解匯入和匯出空間資料來說，有三種空間值很重要：

- OGC 知名文字 (WKT) 表示法
- OGC 知名二進位 (WKB) 表示法
- ESRI 形狀表示法

OGC 知名文字表示法

Spatial Extender 有一些函數從文字說明產生幾何：

ST_GeomFromText

根據任何幾何類型的文字表示法來建立幾何。

ST_PointFromText

從點文字表示法建立點。

ST_LineFromText

從線串文字表示法建立線串。

ST_PolyFromText

從多邊形文字表示法建立多邊形。

ST_MPointFromText

從多點文字表示法建立多點。

ST_MLineFromText

從多線串文字表示法建立多線串。

ST_MPolyFromText

從多重多邊形文字表示法建立多重多邊形。

文字表示法是一個 ASCII 文字格式字串，它允許以 ASCII 文字形式交換幾何。您可在第三代或第四代語言 (3GL 或 4GL) 程式中使用這些函數，因為它們不需要特殊程式結構的定義。ST_AsText 函數將現存幾何轉換成文字表示法。

每一種幾何類型都有一個知名文字表示法，您可使用該表示法建構該類型的新案例以及將現存案例轉換成英數顯示的文字格式。

一個幾何的知名文字表示法定義如下：表示法 `{}`* 表示大括弧內的 0 或更多符號重複；大括弧不出現在輸出符號列示中。

```

<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 > }* )

```

基本函數語法：

```
function (<text description>,<SRID db2gse.coordref>)
```

SRID (空間參照識別字) 和 SPATIAL_REFERENCES 表格的主要鍵識別幾何的空間參照系統，該系統儲存在 SPATIAL_REFERENCES 表格。將一個幾何插入空間直欄之前，它的 SRID 必須符合空間直欄的 SRID。

由單引號括住的 3 個基本元件組成文字說明，例如：

```
<'幾何類型'> ['座標類型'] ['座標列示']
```

其中：

幾何類型

是下列其中一項：點、線串、多邊形、多點、多線串或多重多邊形。

座標類型

指定幾何是否有 Z 座標或測量。若幾何既無 Z 座標亦無測量，則本引數請保留空白。否則，請為含有 Z 座標的幾何將座標類型設為 Z，為含有測量的幾何設定 M，為含有這兩者的幾何設定 ZM。

座標列示

定義幾何的頂點。以逗點區隔座標列示而且以括弧括住。具有多重元件的幾何需要幾組括弧括住元件。若幾何是空白，那麼 EMPTY 關鍵字會取代座標。

表66顯示全部可能的文字表示法範例列示。

表 66. 幾何類型和它們的文字表示法

幾何類型	文字說明	註解
點	point empty	空白點
點	point z empty	具有 z 座標的空白點
點	point m empty	具有測量的空白點

表 66. 幾何類型和它們的文字表示法 (繼續)

幾何類型	文字說明	註解
點	point zm empty	具有 z 座標和測量的空白點
點	point (10.05 10.28)	點
點	point z (10.05 10.28 2.51)	具有 z 座標的點
點	point m (10.05 10.28 4.72)	具有測量的點
點	point zm (10.05 10.28 2.51 4.72)	具有 z 座標和測量的點
線串	linestring empty	空白線串
線串	linestring z empty	具有 z 座標的空白線串
線串	linestring m empty	具有測量的空白線串
線串	linestring zm empty	具有 z 座標和測量的空白線串
線串	linestring (10.05 10.28 , 20.95 20.89)	線串
線串	linestring z (10.05 10.28 3.09, 20.95 31.98 4.72, 21.98 29.80 3.51)	具有 z 座標的線串
線串	linestring m (10.05 10.28 5.84, 20.95 31.98 9.01, 21.98 29.80 12.84)	具有測量的線串
線串	linestring zm ()	具有 z 座標和測量的線串
多邊形	polygon empty	空白多邊形
多邊形	polygon z empty	具有 z 座標的空白多邊形
多邊形	polygon m empty	具有測量的空白多邊形
多邊形	polygon zm empty	具有 z 座標和測量的空白多邊形
多邊形	polygon ((10 10, 10 20, 20 20, 20 15, 10 10))	多邊形
多邊形	polygon z (())	具有 z 座標的多邊形
多邊形	polygon m (())	具有測量的多邊形
多邊形	polygon zm (())	具有 z 座標和測量的多邊形
多點	multipoint empty	空白多點
多點	multipoint z empty	具有 z 座標的空白多點
多點	multipoint m empty	具有測量的空白多點

表 66. 幾何類型和它們的文字表示法 (繼續)

幾何類型	文字說明	註解
多點	multiptoint zm empty	具有 z 座標和測量的空白多點
多點	multiptoint empty	空白多點
多點	multiptoint (10 10, 20 20)	具有兩點的多點
多點	multiptoint z (10 10 2, 20 20 3)	具有 z 座標的多點
多點	multiptoint m (10 10 4, 20 20 5)	具有測量的多點
多點	multiptoint zm (10 10 2 4, 20 20 3 5)	具有 z 座標和測量的多點
多線串	multilinestring empty	空白多線串
多線串	multilinestring z empty	具有 z 座標的空白多線串
多線串	multilinestring m empty	具有測量的空白多線串
多線串	multilinestring zm empty	具有 z 座標和測量的空白多線串
多線串	multilinestring (())	多線串
多線串	multilinestring z (())	具有 z 座標的多線串
多線串	multilinestring m (())	具有測量的多線串
多線串	multilinestring zm (())	具有 z 座標和測量的多線串
多重多邊形	multiptolygon empty	空白多重多邊形
多重多邊形	multiptolygon z empty	具有 z 座標的空白多重多邊形
多重多邊形	multiptolygon m empty	具有測量的空白多重多邊形
多重多邊形	multiptolygon z	具有 z 座標和測量的空白多重多邊形
多重多邊形	multiptolygon ((()))	多重多邊形
多重多邊形	multiptolygon z ((()))	具有 z 座標的多重多邊形
多重多邊形	multiptolygon 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)))	具有測量的多重多邊形
多重多邊形	multiptolygon zm ((()))	具有 z 座標和測量的多重多邊形

OGC 知名二進位 (WKB) 表示法

Spatial Extender 有一些函數從二進位表示法產生幾何：

ST_GeomFromWKB

從任何幾何類型的 WKB 表示法建立幾何。

ST_PointFromWKB

從點 WKB 表示法建立點。

ST_LineFromWKB

從線串 WKB 表示法建立線串。

ST_PolyFromWKB

從多邊形 WKB 表示法建立多邊形。

ST_MPointFromWKB

從多點 WKB 表示法建立多點。

ST_MLineFromWKB

從多線串 WKB 表示法建立多線串。

ST_MPolyFromWKB

從多重多邊形 WKB 表示法建立多重多邊形。

知名二進位表示法為一連串的位元組。它允許幾何以二進位形式在 ODBC 從屬站與 SQL 資料庫之間交換。因為這些幾何函數需要 C 程式設計語言結構定義才能對映二進位表示法，所以它們是專門在第三代語言 (3GL) 程式內使用。它們不適合第四代語言 (4GL) 環境。ST_AsBinary 函數將現存幾何值轉換成知名二進位表示法。

透過一連串數值類型來序列化幾何案例，從而取得幾何的知名二進位表示法。這些類型是從這個設定 (無正負號整數、倍整數) 產生，然後每一個數值類型序列化成為一連串位元組。這些類型是使用數值類型的兩個定義完善的標準二進位表示法 (NDR, XDR) 的其中一個來序列化。位於序列化位元組前面的一位元組標籤，說明用於幾何位元組串流的特定二進位編碼 (NDR 或 XDR)。這兩個幾何編碼的唯一差異是位元組順序差異：XDR 編碼是 Big Endian；NDR 編碼是 Little Endian。

數值類型定義

無正負號整數是 32 位元 (4 位元組) 資料類型，它在範圍 [1, 4294967295] 中編碼非負數整數。

倍整數是 64 位元 (8 位元組) 倍準度資料類型，它使用 IEEE 754 倍準度格式編碼倍準數。

XDR 和 NDR 共用這些定義。

數值類型的 XDR (Big Endian) 編碼

無正負號整數的 XDR 表示是 Big Endian (最大有效位元組是第一個位元組)。

倍整數的 XDR 表示是 Big Endian (正負號位元是第一個位元)。

數值類型的 NDR (Little Endian) 編碼

無正負號整數的 NDR 表示 Little Endian (最小有效位元組是第一個位元組)。

倍整數的 NDR 表示是 Little Endian (正負號位元是最後一個位元組)。

在 NDR 與 XDR 之間轉換

在無正負號整數和倍整數的 NDR 與 XDR 資料類型之間轉換是很簡單的作業。該作業會反轉位元組串流中的每一個無正負號整數或倍整數內的位元組順序。

WKBBGeometry 位元組串流說明

本節說明幾何的知名二進位表示法。基本建置區塊是點的位元組串流，它由兩個倍整數組成。使用已定義的幾何的位元組串流建置其它幾何的位元組串流。

```
// 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,
    wkbNDR = 1
};
WKBPoint {
```

// Big Endian
// Little Endian

```

    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;
    }
};

```

下圖顯示 NDR 表示。

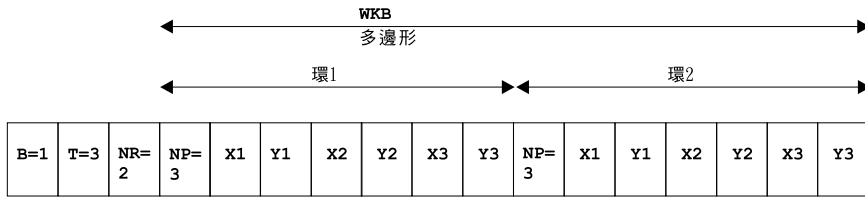


圖 39. 使用 NDR 格式的代表. 具有 2 線性 (NR=2) 的類型多邊形 (T=3) 的 (B=1), 每一個環有 3 點 (NP=3)。

WKB 表示法的主張

幾何的知名二進位表示法代表 Geometry Object Model 和 OpenGIS Abstract Specification 說明的幾何類型案例。

這些主張對於環、多邊形和多重多邊形隱含下列涵意：

線性環 環屬單純和封閉，這表示線性環無法自己相交。

多邊形 在多邊形界限中沒有兩個線性環可以彼此交叉。多邊形界限中的線性環頂多可以在單一點相交，但只作為切線。

多重多邊形

兩個多邊形的內部，它們是多重多邊形無法相交的元素。身為多重多邊形元素的兩個多邊形界限只能接觸有限點數。

ESRI 形狀表示法

Spatial Extender 有一些函數從 ESRI 形狀表示法產生幾何。除了開啓的 GIS 知名二進位表示法支援的二維表示法以外，ESRI 形狀表示法也支援選用性 Z 座標和測量。下列函數從 ESRI 形狀產生幾何：

GeometryFromShape

從任何幾何類型的形狀表示法建立幾何。

PointFromShape

從點形狀表示法建立點。

LineFromShape

從線串形狀表示法建立線串。

PolyFromShape

從多邊形形狀表示法建立多邊形。

MPointFromShape

從多點形狀表示法建立多點。

MLineFromShape

從多線串形狀表示法建立多線串。

MPolyFromShape

從多重多邊形形狀表示法建立多重多邊形。

這些函數的一般語法都相同。第一個引數是輸入成二進位大型物件 (BLOB) 資料類型的形狀表示法。第二個引數是指定給幾何的空間參照識別字整數。GeometryFromShape 函數有下列語法：

```
db2gse.GeometryFromShape(ShapeGeometry Blob(1M), cr db2gse.coordref)
```

因為這些形狀函數需要 C 程式設計語言結構定義才能對映二進位表示法，所以它們是專門在 3GL 程式內使用而不適合 4GL 環境。AsShape 函數將幾何值轉換成 ESRI 形狀表示法。

0 的形狀類型表示一個空形狀，沒有此形狀的幾何資料。

值	形狀類型
0	Null 形狀
1	Point
3*	折線
5	多邊形
8	多點
11	PointZ
13	PolyLineZ
15	PolygonZ
18	MultiPointZ
21	PointM
23	PolyLineM
25	PolygonM
28	MultiPointM

註：* 保留前面沒有指定的形狀類型 (2、4、6 等等) 供將來使用。

XY 空間中的形狀類型

點

一個點由一對倍準度座標 (按照 X、Y 順序) 組成。

表 67. 點位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	1	整數	1	Little
位元組 4	X	X	Double	1	Little
位元組 12	Y	Y	Double	1	Little

多點

一個多點由點集合組成。按照 Xmin、Ymin、Xmax、Ymax 順序儲存外框。

表 68. 多點位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	8	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumPoints	NumPoints	整數	1	Little
位元組 40	點	點	Point	NumPoints	Little

折線

折線是由一個或多個線段組成的一組有排序頂點。線段是兩個點或更多點的連續順序。點不一定彼此相連。線段不一定彼此相交。

因為本規格不禁止具有相同座標的連續點，所以形狀檔讀取器必須處理這種狀況。另一方面，它不接受可能產生的退化、0 長度線段。

折線的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的折線的外框。

NumParts

折線的線段數目。

NumPoints

全部線段的總點數。

Parts 長度 NumParts 陣列。每一條折線把它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存折線的每一個線段的點。線段 2 的點在線段 1 的點後面並依此類推。線段陣列保留每一個線段的起點的陣列指標。點陣列的線段之間沒有定界符號。

表 69. 折線位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	3	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little

註: $X = 44 + 4 * \text{NumParts}$.

多邊形

多邊形是由一個或多個環組成。環是 4 個或更多點的連接順序，這些點形成封閉不相交迴圈。多邊形可含有多個外環。環的頂點順序或方向指出環的哪一邊是多邊形的內部。按照頂點順序沿著環移動的觀察者，其右邊的鄰域即多邊形內側的鄰域。環的頂點（定義多邊形中的孔）是按照逆時鐘方向排列。因此單一環狀多邊形的頂點一定按照順時針方向排列。多邊形的環稱為線段。

因為本規格不禁止具有相同座標的連續點，所以形狀檔讀取器必須處理這種狀況。另一方面，它不接受可能產生的退化、0 長度或 0 區域線段。

多邊形的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的多邊形的外框。

NumParts

多邊形中的環數目。

NumPoints

全部環的總點數。

Parts 長度 NumParts 陣列。就每一個環而言，它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存多邊形中的每一個環的點。Ring 2 的點在 Ring 1 的點後面並依此類推。線段陣列保留每一個環的起點的陣列指標。點陣列的環之間沒有定界符號。

關於多邊形形狀的重要事項：

- 環是封閉的 (環的第一個頂點和最後一個頂點必須是同一個頂點)。
- 點陣列中的環順序如何不重要。
- 儲存在形狀檔中的多邊形必須乾淨。乾淨的多邊形具有下列特點：
 - 本身不相交。這表示屬於一個環的線段無法與屬於另一個環的線段相交。多邊形的環可以在頂點彼此接觸但不沿著線段。共線線段是相交線段。
 - 多邊形的內側在定義它的線的正確邊。按照頂點順序沿著環移動的觀察者，其右邊的鄰域是多邊形內側。因此單一環狀多邊形的頂點一定按照順時針方向排列。定義這些多邊形中的孔的環按照逆時鐘方向排列。

定義多邊形中的孔的環也按照順時針方向排列時會發生 "Dirty" 多邊形，這些多邊形會重疊內部。

範例多邊形案例:

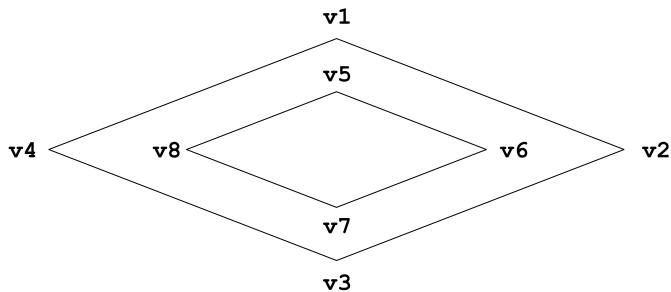


圖 40. 具有一個孔和 8 個頂點的多邊形

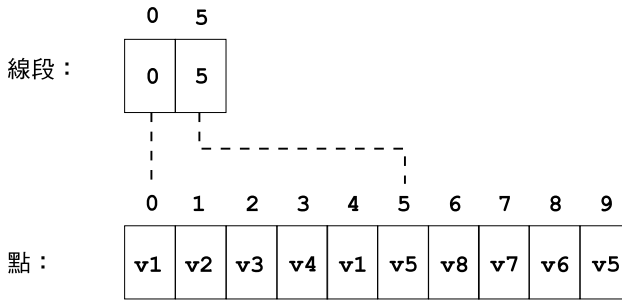


圖 41. 多邊形位元組串流內容。NumParts 等於 2 而且 NumPoints 等於 10。請注意：圓環 (孔) 多邊形的點順序反轉。

表 70. 多邊形位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	5	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little

註: $X = 44 + 4 * \text{NumParts}$.

XY 空間中的測量形狀類型

PointM

一個 PointM 由一對倍準度座標 (按照 X、Y 順序) 加上測量 M 組成。

表 71. PointM 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	21	整數	1	Little
位元組 4	X	X	Double	1	Little
位元組 12	Y	Y	Double	1	Little
位元組 20	M	M	Double	1	Little

MultiPointM

MultiPointM 的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的 MultiPointM 的外框。

NumPoints

點數目。

Points

長度 NumPoints 的點陣列。

NumMs

跟隨的測量數字。若本欄位後面沒有測量，那麼 NumMs 只能有兩個值 0；或等於 NumPoints (若有測量)。

M Range

按照 Mmin、Mmax 順序儲存的 MultiPointM 的最小和最大長度。

M Array

長度 NumPoints 的測量陣列。

表 72. MultiPointM 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	28	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumPoints	NumPoints	整數	1	Little
位元組 40	點	點	Point	NumPoints	Little
位元組 X	NumMs	NumMs	整數	1	Little
位元組 X+4*	Mmin	Mmin	Double	1	Little
位元組 X+12*	Mmax	Mmax	Double	1	Little
位元組 X+20*	Marray	Marray	Double	NumPoints	Little

註:

1. $X = 40 + (16 * NumPoints)$
2. * 選用性

PolyLineM

一個形狀檔的 PolyLineM 含有一個或多個線段。線段是兩個點或更多點的連續順序。線段不一定彼此相連。線段不一定彼此相交。

PolyLineM 的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的 PolyLineM 的外框。

NumParts

PolyLineM 中的線段數目。

NumPoints

全部線段的總點數。

Parts 長度 NumParts 陣列。就線段而言，它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存 PolyLineM 中的每一個線段的點。線段 2 的點在線段 1 的點後面並依此類推。線段陣列保留每一個線段的起點的陣列指標。點陣列的線段之間沒有定界符號。

NumMs

跟隨的測量數字。若本欄位後面沒有測量，那麼 NumMs 只能有兩個值 0；或等於 NumPoints (若有測量)。

M Range

按照 Mmin、Mmax 順序儲存的 PolyLineM 的最小和最大測量。

M Array

長度 NumPoints 陣列。以端對端方式儲存 PolyLineM 中的每一個線段的測量。線段 2 的測量在線段 1 的測量後面並依此類推。線段陣列保留每一個線段的起點的陣列指標。測量陣列的線段之間沒有定界符號。

表 73. PolyLineM 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	13	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little
位元組 Y	NumMs	NumMs	整數	1	Little
位元組 Y+4*	Mmin	Mmin	Double	1	Little
位元組 Y+12*	Mmax	Mmax	Double	1	Little
位元組 Y+20*	Marray	Marray	Double	NumPoints	Little

註:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$.
2. * 選用性

PolygonM

PolygonM 由一些環組成。環是封閉、不自我相交的迴圈。請注意是在 XY 空間計算交集，不是在 XYM 空間計算交集。PolygonM 可含有多個外環。PolygonM 的環稱為線段。

PolygonM 的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的 PolygonM 的外框。

NumParts

PolygonM 中的環數目。

NumPoints

全部環的總點數。

Parts 長度 NumParts 陣列。就每一個環而言，它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存 PolygonM 中的每一個環的點。Ring 2 的點在 Ring 1 的點後面並依此類推。線段陣列保留每一個環的起點的陣列指標。點陣列的環之間沒有定界符號。

NumMs

跟隨的測量數字。若本欄位後面沒有測量，那麼 NumMs 只能有兩個 0 值，或等於 NumPoints (若有測量)。

M Range

按照 Mmin、Mmax 順序儲存的 PolygonM 的最小和最大測量。

M Array

長度 NumPoints 陣列。以端對端方式儲存 PolygonM 中的每一個環的測量。Ring 2 的測量單位在 Ring 1 的測量單位後面並依此類推。線段陣列保留每一個環的起始測量的陣列指標。測量陣列的環之間沒有定界符號。

關於 PolygonM 形狀的重要事項：

- 環是封閉的 (環的第一個頂點和最後一個頂點必須是同一個頂點)。
- 點陣列中的環順序如何不重要。

表 74. PolygonM 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	15	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little
位元組 Y	NumMs	NumMs	整數	1	Little
位元組 Y+4*	Mmin	Mmin	Double	1	Little
位元組 Y+12*	Mmax	Mmax	Double	1	Little
位元組 Y+20*	Marray	Marray	Double	NumPoints	Little

註:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$.
2. * 選用性

XYZ 空間中的形狀類型

PointZ

一個 PointZ 由三倍準度座標、倍準度座標 (按照 X、Y、Z 順序) 加上一個測量組成。

表 75. PointZ 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	11	整數	1	Little
位元組 4	X	X	Double	1	Little
位元組 12	Y	Y	Double	1	Little
位元組 20	Z	Z	Double	1	Little
位元組 28	測量	M	Double	1	Little

MultiPointZ

一個 MultiPointZ 代表一組 PointZs，如下：

- 按照 Xmin、Ymin、Xmax、Ymax 順序儲存外框。

- 按照 Zmin、Zmax 順序儲存界限 Z 範圍。按照 Mmin、Mmax 順序儲存界限 M 範圍。

表 76. MultiPointZ 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	18	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumPoints	NumPoints	整數	1	Little
位元組 40	點	點	Point	NumPoints	Little
位元組 X	Zmin	Zmin	Double	1	Little
位元組 X+8	Zmax	Zmax	Double	1	Little
位元組 X+16	Zarray	Zarray	Double	NumPoints	Little
位元組 Y	NumMs	NumMs	整數	1	Little
位元組 Y+4*	Mmin	Mmin	Double	1	Little
位元組 Y+12*	Mmax	Mmax	Double	1	Little
位元組 Y+20*	Marray	Marray	Double	NumPoints	Little

註:

1. $X = 40 + (16 * \text{NumPoints})$; $Y = X + 16 + (8 * \text{NumPoints})$
2. * 選用性

PolyLineZ

PolyLineZ 由一個或多個線段組成。線段是兩個點或更多點的連續順序。線段不一定彼此相連。線段不一定彼此相交。

PolyLineZ 的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的 PolyLineZ 的外框。

NumParts

PolyLineZ 中的線段數目。

NumPoints

全部線段的總點數。

Parts 長度 NumParts 陣列。就線段而言，它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存 PolyLineZ 中的每一個線段的

點。線段 2 的點在線段 1 的點後面並依此類推。線段陣列保留每一個線段的起點的陣列指標。點陣列的線段之間沒有定界符號。

Z Range

按照 Zmin、Zmax 順序儲存的 PolyLineZ 的最小和最大 Z 值。

Z Array

長度 NumPoints 陣列。以端對端方式儲存 PolyLineZ 中的每一個線段的 Z 值。線段 2 的 Z 值在線段 1 的 Z 值後面並依此類推。線段陣列保留每一個線段的起點的陣列指標。Z 陣列的線段之間沒有定界符號。

NumMs

跟隨的測量數字。若本欄位後面沒有測量，那麼 NumMs 只能有兩個值 0；或等於 NumPoints (若有測量)。

M Range

按照 Mmin、Mmax 順序儲存的 PolyLineZ 的最小和最大測量。

M Array

長度 NumPoints 陣列。以端對端方式儲存 PolyLineZ 中的每一個線段的測量。線段 2 的測量在線段 1 的測量後面並依此類推。線段陣列保留每一個線段的起始測量的陣列指標。測量陣列的線段之間沒有定界符號。

表 77. PolyLineZ 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	13	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little
位元組 Y	Zmin	Zmin	Double	1	Little
位元組 Y+8	Zmax	Zmax	Double	1	Little
位元組 Y+16	Zarray	Zarray	Double	NumPoints	Little
位元組 Z	NumMs	NumMs	整數	1	Little
位元組 Z+4*	Mmin	Mmin	Double	1	Little
位元組 Z+12*	Mmax	Mmax	Double	1	Little
位元組 Z+20*	Marray	Marray	Double	NumPoints	Little

註:

1. $X = 44 + (4 * \text{NumParts})$, $Y = X + (16 * \text{NumPoints})$, $Z = Y + 16 + (8 * \text{NumPoints})$
2. * 選用性

PolygonZ

PolygonZ 由一些環組成。環是封閉、不自我相交的迴圈。PolygonZ 可含有多個外環。PolygonZ 的環稱為線段。

PolygonZ 的欄位：

Box 按照 Xmin、Ymin、Xmax、Ymax 順序儲存的 PolygonZ 的外框。

NumParts

PolygonZ 中的環數目。

NumPoints

全部環的總點數。

Parts 長度 NumParts 陣列。就每一個環而言，它的第一個點的指標儲存在點陣列中。陣列指標相對於 0。

Points

長度 NumPoints 陣列。以端對端方式儲存 PolygonZ 中的每一個環的點。Ring 2 的點在 Ring 1 的點後面並依此類推。線段陣列保留每一個環的起點的陣列指標。點陣列的環之間沒有定界符號。

Z Range

按照 Zmin、Zmax 順序儲存的弦的最小和最大 Z 值。

Z Array

長度 NumPoints 陣列。以端對端方式儲存 PolygonZ 中的每一個環的 Z 值。Ring 2 的 Z 值在 Ring 1 的 Z 值後面並依此類推。線段陣列保留每一個環的起始 Z 值的陣列指標。Z 值陣列的環之間沒有定界符號。

NumMs

跟隨的測量數字。若本欄位後面沒有測量，那麼 NumMs 只能有兩個值 0；或等於 NumPoints (若有測量)。

M Range

按照 Mmin、Mmax 順序儲存的 PolygonZ 的最小和最大測量。

M Array

長度 NumPoints 陣列。以端對端方式儲存 PolygonZ 中的每一個環的測

量。Ring 2 的測量單位在 Ring 1 的測量單位後面並依此類推。線段陣列保留每一個環的起始測量的陣列指標。測量陣列的環之間沒有定界符號。

關於 PolygonZ 形狀的重要事項：

- 環是封閉的 (環的第一個頂點和最後一個頂點必須是同一個頂點)。
- 點陣列中的環順序如何不重要。

表 78. PolygonZ 位元組串流內容

位置	欄位	值	類型	數字	順序
位元組 0	形狀類型	15	整數	1	Little
位元組 4	方框	方框	Double	4	Little
位元組 36	NumParts	NumParts	整數	1	Little
位元組 40	NumPoints	NumPoints	整數	1	Little
位元組 44	線段	線段	整數	NumParts	Little
位元組 X	點	點	Point	NumPoints	Little
位元組 Y	Zmin	Zmin	Double	1	Little
位元組 Y+8	Zmax	Zmax	Double	1	Little
位元組 Y+16	Zarray	Zarray	Double	NumPoints	Little
位元組 Z	NumMs	NumMs	整數	1	Little
位元組 Z+4*	Mmin	Mmin	Double	1	Little
位元組 Z+12*	Mmax	Mmax	Double	1	Little
位元組 Z+20*	Marray	Marray	Double	NumPoints	Little

第3篇 後記

注意事項

而在其它國家中，IBM 不見得有提供本書中所提的各項產品、服務或功能。要知道在您所在地區是否可用到這些產品與服務時，請向當地的 IBM 服務代表查詢。本書在提及 IBM 的產品、程式或服務時，不表示或暗示只能使用 IBM 的產品、程式或服務。只要未侵犯 IBM 的智慧財產權，任何功能相當的產品、程式或服務都可以取代 IBM 的產品、程式或服務。不過，其他非 IBM 產品、程式、或服務在運作上的評價與驗證，其責任屬於使用者。

本文件中包含著 IBM 所擁有之專利或暫准專利。使用者不因取得本書而享有本書內容之專利權。您可以用書面方式來查詢授權，來函請寄到：

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

若要查詢有關二位元組 (DBCS) 資訊的授權事宜，請聯絡當地國家的 IBM 智慧財產部門，或者用書面信函寄到：

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

下列段落不適用於英國，或與該國之法律條款抵觸，即不適用： IBM 以『交附時之現況』供應本書，而不提供任何明示或默示之保證，包括但不限於如默示保證之適售性或符合客戶之特殊使用目的；某些地區不允許在某些交易行為中，不作明示或暗示的保證，在此情況下，本聲明亦不適用。

本資訊可能會有技術上或排版印刷上的訛誤。因此，IBM 公司會定期修訂，並將修訂後的內容納入新版中。同時，IBM 會隨時改進並 (或) 變動本書中所提及的產品及 (或) 程式，而不另行通知。

本書對於非 IBM 網站的援引只是為了方便而提供，並不對這些網站作任何認可。這些網站上的內容並非本 IBM 產品內容的一部份，用戶使用該網站時應自行承擔風險。

當您提供資訊給 IBM 時，您即授權予 IBM 以其認為適當的方式來使用或分送資訊，而不必對您負起任何責任。

本程式之獲授權者若希望取得相關資料，以便使用下列資訊者可洽詢 IBM。其下列資訊指的是：(1) 獨立建立的程式與其他程式 (包括此程式) 之間更換資訊的方式 (2) 相互使用已交換之資訊方法，若有任何問題請聯絡：

IBM Canada Limited
Office of the Lab Director
1150 Eglinton Ave. East
North York, Ontario
M3C 1H7
CANADA

上述資料之取得有其特殊要件，在某些情況下必須付費方得使用。

IBM 基於雙方之「IBM 客戶合約」、「IBM 國際程式授權合約」或任何同等合約中的條款約定，提供本書中所說的授權程式與其所有適用的授權材料。

此間所含之任何效能資料，皆是得自控制的環境之下；因此不同作業環境之下所得的結果，可能會有很大差異。部份測量可能是在開發中的系統上執行，因此不保證可以從一般的系統獲致相同的結果。甚至有部份的測量，是利用插補法而得的估計值，其實際結果可能會有不同。本書的使用者應根據其特有的環境，驗證出適用的資料。

本書所提及之非 IBM 產品資訊，係一由產品的供應商，或其出版的聲明或其他公開管道取得。IBM 並未測試過這些產品，也無法確認這些非 IBM 產品的執行效能、相容性、或任何對產品的其他主張是否完全無誤。如果您對非 IBM 產品的性能有任何的疑問，請逕向該產品的供應商查詢。

有關 IBM 未來動向的任何陳述，僅代表 IBM 的目標而已，並可能於未事先聲明的情況下有所變動或撤回。

本書中含有日常商業活動所用的資料及報告範例。爲了盡可能詳細，範例中涵蓋了個人、公司、品牌及產品的名稱。所有名稱純屬虛構，如有雷同純屬巧合。

著作權授權：

本書包含原始語言的範例應用程式，用以說明各種作業平台上的程式設計技術。您可以基於研發、使用、銷售或散布符合作業平台應用程式介面的應用程式等目的，以任何形式複製、修改及散布這些範例程式而不必向 IBM 付費。這些範例應用程式並未經過徹底的測試，因此，IBM 不提供明示或默示其可靠性、有用性或符合特定效用的保證。

這些範例程式或是任何衍生著作的每一份拷貝或任何部份，都必須具有下列的著作權聲明：

© (your company name) (year). Portions of this code are derived from IBM Corp. Sample Programs. © Copyright IBM Corp. _enter the year or years_. All rights reserved.

商標

下列術語 (以星號 (*) 標示) 是 IBM 公司在美國、其它國家或兩者的商標。

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

下列專有名詞是其它公司的商標或註冊商標：

Microsoft、Windows、Windows NT 是微軟公司的商標或註冊商標。

Java 以及所有與 Java 有關的商標與標章，以及 Solaris 是 Sun Microsystems, Inc. 在美國、其它國家或兩者的商標。

Tivoli 與 NetView 是 Tivoli Systems Inc. 在美國、其它國家或兩者的商標。

UNIX 是 X/Open Company Limited 在美國、其它國家或兩者的註冊商標，須經該公司授權始可使用。

其它公司、產品或服務名稱 (以兩顆星號 (*) 標示) 可能是其它公司的商標或服務標誌。

索引

索引順序以中文字，英文字，及特殊符號之次序排列。

〔三劃〕

大地測量基準點 297

〔四劃〕

中央子午線 299

內部 131, 135

方位投影 301

比例係數

指定 35, 37

〔五劃〕

包封 125

外部 131, 135

平面投影 301

〔六劃〕

地心座標系統 295

地理特徵

以資料代表 4

相關資料類型 40

說明 3

地理資訊系統 (GIS)

使用 10

建立 8

說明 3

地理編碼程式

以批次模式執行

討論 46

「執行地理編碼程式」視窗

48

範例程式 63, 64

db2gse.gse_run_gc 98

批次 46

地理編碼程式 (繼續)

非預設地理編碼程式

使用 db2gse.gse_register_gc 來

登記 90

使用 db2gse.gse_unregister_gc 來

取消登記 100

討論 45

討論 45

停用自動地理編碼

「執行地理編碼程式」視窗

49

範例程式 64

停用自動地理編碼程式

db2gse.gse_disable_autogc 71

啓用自動地理編碼

「建立空間層」視窗 41

討論 39, 46

範例程式 64

db2gse.gse_enable_autogc 75

預設地理編碼程式 45

精準度 12

說明 6

增量 46

DB2GSE.SPATIAL_GEOCODER

型錄概略表 120

地圖投影 300

地圖投影參數 301

多重多邊形 133, 142

多線串 132, 141

多點 132, 141

多點位元組串流內容 313

多邊形 132, 140

多邊形位元組串流內容 316

安裝 Spatial Extender

軟硬體需求 16

驗證 23

自動地理編碼 46

〔七劃〕

作業的實務範例 10

形狀

在 XY 空間中 313

形狀 (繼續)

在 XYZ 空間中 320

批次地理編碼 46

折線位元組串流內容 314

投影

方位 301

平面 301

地圖

參數 301

類型 300

圓錐 301

角度單位 295

〔八劃〕

來源資料 5

空間函數

用來開發空間索引 59

按執行的作業分類 57

述詞 58

類型

比較幾何的函數 144

述詞函數 144

產生幾何的函數 155

資料交換 160

與可引證的幾何相關 137

與幾何內容相關 133

顯示幾何之間關係的函數 144

AsShape 163, 168

EnvelopesIntersect 149, 169

GeometryFromShape 162

Is3d 134, 173

IsMeasured 134, 174

LineFromShape 162, 175

LocateAlong 159, 177

LocateBetween 159, 179

M 138, 181

MLine FromShape 182

MLineFromShape 163

MPointFromShape 163, 184

MPolyFromShape 163, 186

PointFromShape 137, 162, 187

空間函數 (繼續)

PolyFromShape 163, 189
ShapeToSQL 162, 191
ST_Area 140, 143
ST_AsBinary 162, 195
ST_AsText 161, 196
ST_Boundary 135, 197
ST_Buffer 158, 199
ST_Centroid 140, 143, 201
ST_Contains 154, 202
ST_ConvexHull 160
ST_Convexhull 204
ST_CoordDim 135, 206
ST_Crosses 151, 208
ST_Difference 156, 210
ST_Dimension 136, 211
ST_Disjoint 147, 213
ST_Distance 155, 215
ST_Endpoint 139, 216
ST_Envelope 136, 217
ST_Equals 145, 219
ST_ExteriorRing 140, 220
ST_GeometryN 141, 222
ST_GeometryType 133, 223
ST_GeomFromText 160, 225, 303
ST_GeomFromWKB 161, 227, 308
ST_InteriorRingN 140, 229
ST_Intersection 155, 234
ST_Intersects 148, 236
ST_IsClosed 139, 141, 237
ST_IsEmpty 136, 239
ST_IsRing 139, 241
ST_IsSimple 135, 242
ST_IsValid 133, 243
ST_Length 139, 141, 245
ST_LineFromText 160, 247, 303
ST_LineFromWKB 161, 248, 308
ST_MLineFromText 161, 250, 303
ST_MLineFromWKB 162, 251, 308
ST_MPointFromText 161, 253, 303
ST_MPointFromWKB 162, 254, 308

空間函數 (繼續)

ST_MPolyFromText 161, 256, 303
ST_MPolyFromWKB 162, 257, 308
ST_NumGeometries 141, 258
ST_NumInteriorRing 140, 259
ST_NumPoints 139, 260
ST_OrderingEquals 146, 261
ST_Overlaps 150, 262
ST_Perimeter 141, 264
ST_Point 138, 265
ST_PointFromText 138, 266, 303
ST_PointFromWKB 138, 161, 267, 308
ST_PointN 139, 269
ST_PointOnSurface 141, 270
ST_PolyFromText 161, 271, 303
ST_PolyFromWKB 162, 272, 308
ST_Polygon 160, 274
ST_Relate 155, 275
ST_SRID 277
ST_StartPoint 138, 278
ST_SymmetricDiff 157, 279
ST_Touches 149, 281
ST_Transform 282
ST_Union 157, 284
ST_Within 152, 285
ST_WKBToSQL 161, 286
ST_WKTToSQL 160, 288
ST_X 138, 289
ST_Y 138, 290
Z 138
.ST_Area 193

空間直欄 45

空間索引 123

如何產生它們 125
使用 129
建立
 決定格線大小 56, 130
 建立「空間索引」視窗 55
 範例程式 63
 db2gse.gse_enable_idx 79
格線索引 55
開發 59

空間參照系統

建立

「建立空間參照系統」視窗 36

討論 33

範例程式 62

db2gse.gse_enable_sref 81

指定參數

比例係數 35, 37

假 M 35, 38

假 X 35, 37

假 Y 35, 37

假 Z 35, 38

偏移係數 35, 37

M 單位 36, 38

XY 單位 35, 37

Z 單位 36, 38

捨棄

範例程式 62

db2gse.gse_disable_sref 74

說明 9

DB2GSE.SPATIAL_REF_SYS 型錄

概略表 121

空間資料

的性質 5

從其它空間資料取得

 衍生資料的空間函數 155

 討論 6

從屬性資料取得 6

匯入

討論 7, 49

「匯入空間資料」視窗 50, 52

範例程式 63

db2gse.gse_import_sde 85

db2gse.gse_import_shape 87

匯出

討論 49

「匯出空間資料」視窗 53

範例程式 65

db2gse.gse_export_shape 83

檔案格式

ESRI 形狀表示法 162, 311

WKB (知名二進位) 表示法 161, 308

WKT (知名文字) 表示法 160, 303

空間資料類型 39
 與幾何的對應 133
 說明 39
空間資訊
 說明 3
 擷取及分析
 使用的介面 8, 57
 使用的空間函數類型 57
 使用空間述詞函數 58
 開發空間索引 59
 範例程式 65
長度單位 295

〔九劃〕

型樣矩陣 145
型錄概略表
 DB2GSE.COORD_REF_SYS 119
 DB2GSE.GEOMETRY_
 COLUMNS 120
 DB2GSE.SPATIAL_
 GEOCODER 120
 DB2GSE.SPATIAL_REF_SYS 121
建立「空間索引」視窗 55
 「建立空間參照」視窗 37
 「建立空間參照系統」視窗 36
 「建立空間層」視窗
 將表格直欄登記為「層」 42
 將概略表直欄登記為「層」 44
查詢
 使用的空間函數類型 57
 使用空間述詞函數 58
 提交的介面 8, 57
 開發空間索引 59
 範例程式 65
為空間作業啓用資料庫
 討論 31
 說明 9
 「DB2 控制中心」功能表選項
 32
界限 131, 135

〔十劃〕

座標
 說明 5
 X 座標
 幾何內容 134

座標 (繼續)
 X 座標 (繼續)
 說明 33
 Y 座標
 幾何內容 134
 說明 33
 Z 座標
 幾何內容 134
 說明 33
座標系統 293
 從...衍生空間參照系統 33
 說明 5, 33
 DB2GSE.COORD_REF_SYS 型錄
 概略表 119
格式排列 160
格線索引 55
訊息 103

〔十一劃〕

假 M
 指定 35, 38
假 X
 指定 35, 37
假 Y
 指定 35, 37
假 Z
 指定 35, 38
偏移係數
 指定 35, 37
參考訊息 103
參考資料 31
 「執行地理編碼程式」視窗 48, 49
旋轉橢圓體 296

〔十二劃〕

幾何
 內容
 內部 131, 135
 包封 125
 外部 131, 135
 界限 131, 135
 測量 134
 維度 136
 類別 133
 X 座標 134

幾何 (繼續)
 內容 (繼續)
 Y 座標 134
 Z 座標 134
 多重多邊形 133, 142
 多線串 132, 141
 多點 132, 141
 多邊形 132, 140
 空間索引格線 125
 討論 131
 與空間資料類型的對應 133
 線串 132, 138
 點 132, 137
測量
 幾何內容 134
 說明 33, 134

〔十三劃〕

「匯入空間資料」視窗 50, 52
「匯出空間資料」視窗 53, 54
圓錐投影 301
資料庫
 為空間作業啓用
 討論 31
 範例程式 62
 「DB2 控制中心」功能表選項
 32
 停用空間作業的支援
 範例程式 62
 db2gse.gse_disable_db 73
 啓用空間作業
 db2gse.gse_enable_db 78
資料項目 5
預設地理編碼程式 45

〔十四劃〕

疑難排解秘訣
 使用 runGseDemo 24
 範例程式 24
磁碟空間需求 17
精準度
 地理編碼 12
 地理編碼程式 46
 為空間參照系統保留 34
維度 136

〔十五劃〕

增量地理編碼 46
層

使用 db2gse.gse_unregister_layer 來
取消登記 101

將表格直欄登記成爲
「建立空間層」視窗 41

範例程式 63

db2gse.gse_register_layer 92

將概略表直欄登記成爲

「建立空間層」視窗 44

範例程式 64

db2gse.gse_register_layer 92

說明 10

DB2GSE.GEOMETRY_
COLUMNS 型錄概略表 120

範例程式

說明 61

線串 132, 138

線性環 311

〔十六劃〕

錯誤訊息 103

〔十七劃〕

儲存程序

db2gse.gse_disable_autogc 71

db2gse.gse_disable_db 73

db2gse.gse_disable_sref 74

db2gse.gse_enable_autogc 75

db2gse.gse_enable_db 78

db2gse.gse_enable_idx 79

db2gse.gse_enable_sref 81

db2gse.gse_export_shape 83

db2gse.gse_import_sde 85

db2gse.gse_import_shape 87

db2gse.gse_register_gc 90

db2gse.gse_register_layer 92

db2gse.gse_run_gc 98

db2gse.gse_unregister_gc 100

db2gse.gse_unregister_layer 101

應用程式

撰寫原則 61

儲存程序 69

點 132, 137

點位元組串流內容 313

〔十九劃〕

類別 133

〔二十劃〕

觸發函式

用來呼叫地理編碼程式 39, 46

停用自動地理編碼程式

db2gse.gse_disable_autogc 71

啓用自動地理編碼

db2gse.gse_enable_autogc 75

警告訊息 103

〔二十一劃〕

屬性資料 4

A

AIX

儲存參考資料的位置 31

儲存常數的巨集定義的位置 69

ArcExplorer

作爲介面使用 8, 57

ArcExplorer Java 版本 3.0

下載 26

AsShape 134, 163, 168

B

B 樹狀結構索引 124

D

DB2 控制中心

建立「空間索引」視窗 55

「建立空間參照」視窗 37

「建立空間參照系統」視窗 36

「建立空間層」視窗

將表格直欄登記爲「層」 42

將概略表直欄登記爲「層」

44

DB2 控制中心 (繼續)

「執行地理編碼程式」視窗 48, 49

從...呼叫 Spatial Extender 28

「匯入空間資料」視窗 50, 52

「匯出空間資料」視窗 53, 54

DB2GSE.COORD_REF_SYS 119

DB2GSE.GEOMETRY_COLUMNS 120

db2gse.gse_disable_autogc 71

db2gse.gse_disable_db 73

db2gse.gse_disable_sref 74

db2gse.gse_enable_autogc 75

db2gse.gse_enable_db 78

db2gse.gse_enable_idx 79

db2gse.gse_enable_sref 81

db2gse.gse_export_shape 83

db2gse.gse_import_sde 85

db2gse.gse_import_shape 87

db2gse.gse_register_gc 90

db2gse.gse_register_layer 92

db2gse.gse_run_gc 98

db2gse.gse_unregister_gc 100

db2gse.gse_unregister_layer 101

DB2GSE.SPATIAL_GEOCODER 120

DB2GSE.SPATIAL_REF_SYS 121

E

EBNF (Extended Backus Naur) 293

EnvelopesIntersect 149, 169

ESRI 形狀表示法

相關的空間函數 162

討論 311

G

GEOGCS 關鍵字 294

GeometryFromShape 162, 171

I

Is3d 134, 173

IsMeasured 134, 174

J

Java 2 Runtime Environment (JRE)
v1.2.2 26

L

LineFromShape 162, 175
LocateAlong 159, 177
LocateBetween 159, 179

M

M 138, 181
M 單位
指定 36, 38
MLine FromShape 182
MLineFromShape 163
MPointFromShape 163, 184
MPolyFromShape 163, 186
MultiPointM 位元組串流內容 316,
317
MultiPointZ 位元組串流內容 320,
321

N

NDR 編碼 308, 309

P

PointFromShape 137, 162, 187
PointM 位元組串流內容 316
PointZ 位元組串流內容 320
PolyFromShape 163, 189
PolygonM 位元組串流內容 320
PolygonZ 位元組串流內容 324
PolyLineM 位元組串流內容 318
PolyLineZ 位元組串流內容 322
POSC/EPSB 座標系統模型 293
PROJCS 關鍵字 294

S

ShapeToSQL 162, 191
Spatial Extender
介面 7

Spatial Extender (繼續)

目的 3
安裝
軟體需求 16
作業, 摘要
由儲存程序執行 70
概觀 8
實務範例 10
範例程式 61
空間函數 165
型錄概略表 119
架構 15
從 DB2 控制中心呼叫 28
資源

空間作業 31
參考資料 31
摘要 31
範例程式
說明 61
錯誤、警告和參考訊息 103
儲存程序 69
應用程式
撰寫原則 61
儲存程序 69

Spatial Extender 的介面 7

SRID (空間參照系統識別字) 305

ST_Area 140, 143, 193
ST_AsBinary 162, 195
ST_AsText 161, 196
ST_Boundary 135, 197
ST_Buffer 158, 199
ST_Centroid 140, 143, 201
ST_Contains 154, 202
ST_ConvexHull 160
ST_Convexhull 204
ST_CoordDim 135, 206
ST_Crosses 151, 208
ST_Difference 156, 210
ST_Dimension 136, 211
ST_Disjoint 147, 213
ST_Distance 155, 215
ST_Endpoint 139, 216
ST_Envelope 136, 217
ST_Equals 145, 219
ST_ExteriorRing 140, 220
ST_GeometryN 141, 222

ST_GeometryType 133, 223
ST_GeomFromText 160, 225, 303
ST_GeomFromWKB 161, 227, 308
ST_InteriorRingN 140, 229
ST_Intersection 155, 234
ST_Intersects 148, 236
ST_IsClosed 139, 141, 237
ST_IsEmpty 136, 239
ST_IsRing 139, 241
ST_IsSimple 135, 242
ST_IsValid 133, 243
ST_Length 139, 141, 245
ST_LineFromText 160, 247, 303
ST_LineFromWKB 161, 248, 308
ST_MLineFromText 161, 250, 303
ST_MLineFromWKB 162, 251, 308
ST_MPointFromText 161, 253, 303
ST_MPointFromWKB 162, 254, 308
ST_MPolyFromText 161, 256, 303
ST_MPolyFromWKB 162, 257, 308
ST_NumGeometries 141, 258
ST_NumInteriorRing 140, 259
ST_NumPoints 139, 260
ST_OrderingEquals 146, 261
ST_Overlaps 150, 262
ST_Perimeter 141, 264
ST_Point 138, 265
ST_PointFromText 138, 266, 303
ST_PointFromWKB 138, 161, 267,
308
ST_PointN 139, 269
ST_PointOnSurface 141, 270
ST_PolyFromText 161, 271, 303
ST_PolyFromWKB 162, 272, 308
ST_Polygon 160, 274
ST_Relate 155, 275
ST_SRID 277
ST_StartPoint 138, 278
ST_SymmetricDiff 157, 279
ST_Touches 149, 281
ST_Transform 282
ST_Union 157, 284
ST_Within 152, 285
ST_WKBToSQL 161, 286
ST_WKTToSQL 160, 288
ST_X 138, 289

ST_Y 138, 290

U

UNIT 關鍵字 294

W

Windows NT

儲存參考資料的位置 31

儲存常數的巨集定義的位置 69

WKB (知名二進位) 表示法

相關的空間函數 161

討論 308

WKBGeometry 309

WKBGeometry 位元組串流 309

WKT (知名文字) 表示法

相關的空間函數 160

討論 303

X

X 座標

幾何內容 134

說明 33

XDR 編碼 308, 309

XY 空間中的測量形狀類型 316

XY 單位

指定 35, 37

Y

Y 座標

幾何內容 134

說明 33

Z

Z 138

Z 座標

幾何內容 134

說明 33

Z 單位

指定 36, 38

連絡 IBM

當您有技術上的問題時，請在洽詢「DB2 客戶支援中心」之前，仔細閱讀並執行疑難排解指南所建議的動作。該指南會告訴您必須預先準備的資訊，協助「DB2 客戶支援中心」提供更完善的服務。

若要取得 DB2 Universal Database 產品的相關資訊，或是訂購該系列產品，請洽詢當地 IBM 分公司的業務代表，或是 IBM 授權的軟體經銷商。

美國地區居民請撥下列電話：

- 1-800-237-5511，客戶支援中心
- 1-888-426-4343，取得可用服務選項的資訊

產品資訊

美國地區居民請撥下列電話：

- 1-800-IBM-CALL (1-800-426-2255) 或 1-800-3IBM-OS2 (1-800-342-6672)，訂購產品或取得一般資訊。
- 1-800-879-2755，訂購出版品。

<http://www.ibm.com/software/data/>

DB2 全球資訊網 (WWW) 網頁提供最新的 DB2 資訊，包括新聞、產品說明、教育課程等。

<http://www.ibm.com/software/data/db2/library/>

DB2 Product and Service Technical Library 提供常見問題、修正程式、書籍及最新的 DB2 技術資訊。

註：本資訊僅提供英文版。

<http://www.elink.ibm.com/pbl/pbl/>

International Publications 訂購網站會提供書籍的訂購資訊。

<http://www.ibm.com/education/certify/>

IBM 網站中的 Professional Certification Program 會提供包括 DB2 在內之各種 IBM 產品的認證測試資訊。

<ftp://software.ibm.com>

以匿名方式登入。您可以在目錄 /ps/products/db2 中找到 DB2 及其它產品的相關示範程式、修訂程式、資訊及工具。

comp.databases.ibm-db2, bit.listserv.db2-l

使用者可以利用這些 Internet 新聞群組討論 DB2 產品的使用經驗。

在 Compuserve 上：GO IBMDB2

輸入此項指令，即可存 IBM DB2 Family 論壇。這些論壇支援所有 DB2 產品。

關於美國以外地區如何連絡 IBM 的資訊，請參考 *IBM Software Support Handbook* 的附錄 A。若要存取本文件，請造訪下列網頁：<http://www.ibm.com/support/>，然後選取接近網頁底端的 IBM Software Support Handbook 鏈結。

註：在某些國家，IBM 授權經銷商應連絡其經銷商支援組織，而不是 IBM 支援中心。

讀者意見表

為使本書盡善盡美，本公司極需您寶貴的意見；懇請您使用過後，撥冗填寫下表，惠予指教。

請於下表適當空格內，填入記號（√）；我們會在下一版中，作適當修訂，謝謝您的合作！

評估項目	評 估 意 見	備 註
正 確 性	內容說明與實際程序是否符合 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	參考書目是否正確 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
一 致 性	文句用語及風格，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	實際畫面訊息與本書所提之畫面訊息是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
完 整 性	是否遺漏您想知道的項目 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字句、章節是否有遺漏 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
術語使用	術語之使用是否恰當 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	術語之使用，前後是否一致 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
可 讀 性	文句用語是否通順 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	有否不知所云之處 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
內容說明	內容說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	例題說明是否詳盡 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
排版方式	本書的形狀大小，版面安排是否方便使用 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	字體大小，顏色編排，是否有助於閱讀 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
目錄索引	目錄內容之編排，是否便於查考 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	索引語錄之排定，是否便於查考 <input type="checkbox"/> 是 <input type="checkbox"/> 否	
	※評估意見為 "否" 者，請於備註欄說明。	

其他：（篇幅不夠時，請另紙說明。）

上述改正意見，一經採用，本公司有合法之使用及發佈權利，特此聲明。

折疊線

台北市115 南港區三重路十九之十一號五棟九樓

臺灣國際商業機器股份有限公司
大中華研發中心 軟體國際部 啟

廣告回信
臺灣北區郵政管理局 登記
北台字第0587號

(免貼郵票)

寄件人 姓名：
地址：

寄

折疊線



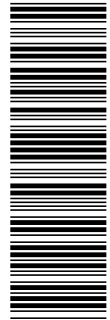
Part Number: CT7M2TC

Printed in Singapore

SC40-0527-01



(1P) P/N: CT7M2TC



Spine information:



**IBM[®] DB2[®] Spatial
Extender**

DB2 Spatial Extender 使用與參考手冊 版本 7