

Tibero

Spatial 참조 안내서

Tibero 6

TMAXTibero

Copyright © 2025 TmaxTibero Co., Ltd. All Rights Reserved.

Copyright Notice

Copyright © 2025 TmaxTibero Co., Ltd. All Rights Reserved.

대한민국 경기도 성남시 분당구 정자일로 45, 티맥스소프트타워 우)13613

Website

<http://www.tmaxtibero.com>

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

Restricted Rights Legend

All TmaxTibero Software (Tibero®) and documents are protected by copyright laws and international convention. TmaxTibero software and documents are made available under the terms of the TmaxTibero License Agreement and this document may only be distributed or copied in accordance with the terms of this agreement. No part of this document may be transmitted, copied, deployed, or reproduced in any form or by any means, electronic, mechanical, or optical, without the prior written consent of TmaxTibero Co., Ltd. Nothing in this software document and agreement constitutes a transfer of intellectual property rights regardless of whether or not such rights are registered) or any rights to TmaxTibero trademarks, logos, or any other brand features.

This document is for information purposes only. The company assumes no direct or indirect responsibilities for the contents of this document, and does not guarantee that the information contained in this document satisfies certain legal or commercial conditions. The information contained in this document is subject to change without prior notice due to product upgrades or updates. The company assumes no liability for any errors in this document.

이 소프트웨어(Tibero®) 사용설명서의 내용과 프로그램은 저작권법과 국제 조약에 의해서 보호받고 있습니다. 사용설명서의 내용과 여기에 설명된 프로그램은 TmaxTibero Co., Ltd.와의 사용권 계약 하에서만 사용이 가능하며, 사용설명서는 사용권 계약의 범위 내에서만 배포 또는 복제할 수 있습니다. 이 사용설명서의 전부 또는 일부분을 TmaxTibero의 사전 서면 동의 없이 전자, 기계, 녹음 등의 수단을 사용하여 전송, 복제, 배포, 2차적 저작물작성 등의 행위를 하여서는 안 됩니다.

이 소프트웨어 사용설명서와 프로그램의 사용권 계약은 어떠한 경우에도 사용설명서 및 프로그램과 관련된 지적 재산권(등록 여부를 불문)을 양도하는 것으로 해석되지 아니하며, 브랜드나 로고, 상표 등을 사용할 권한을 부여하지 않습니다. 사용설명서는 오로지 정보의 제공만을 목적으로 하고, 이로 인한 계약상의 직접적 또는 간접적 책임을 지지 아니하며, 사용설명서 상의 내용은 법적 또는 상업적인 특정한 조건을 만족시키는 것을 보장하지는 않습니다. 사용설명서의 내용은 제품의 업그레이드나 수정에 따라 그 내용이 예고 없이 변경될 수 있으며, 내용상의 오류가 없음을 보장하지 아니합니다.

Trademarks

Tibero® is a registered trademark of TmaxTibero Co., Ltd. Other products, titles or services may be registered trademarks of their respective companies.

Tibero®는 TmaxTibero Co., Ltd.의 등록 상표입니다. 기타 모든 제품들과 회사 이름은 각각 해당 소유주의 상표로서 참조용으로만 사용됩니다.

Open Source Software Notice

Some modules or files of this product are subject to the terms of the following licenses. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

Detailed Information related to the license can be found in the following directory : \${INSTALL_PATH}/license/oss_licenses

본 제품의 일부 파일 또는 모듈은 다음의 라이선스를 준수합니다. : OpenSSL, RSA Data Security, Inc., Apache Foundation, Jean-loup Gailly and Mark Adler, Paul Hsieh's hash

관련 상세한 정보는 제품의 다음의 디렉터리에 기재된 사항을 참고해 주십시오. : \${INSTALL_PATH}/license/oss_licenses

안내서 정보

안내서 제목: Tibero Spatial 참조 안내서

발행일: 2025-09-30

소프트웨어 버전: Tibero 6.7.4

안내서 버전: v6.7.4

내용 목차

안내서에 대하여	ix
제1장 Spatial 소개	1
1.1. 개요	1
1.2. GEOMETRY 객체	1
제2장 Spatial 구조	3
2.1. 테이블	3
2.1.1. GEOMETRY_COLUMNS_BASE	3
2.1.2. SPATIAL_REF_SYS_BASE	4
2.2. 뷰	4
2.2.1. ALL_GEOMETRY_COLUMNS	4
2.2.2. SPATIAL_REF_SYS	5
2.2.3. USER_GEOMETRY_COLUMNS	5
2.3. 프러시저	6
2.3.1. REGISTER_GEOMETRY_COLUMNS	6
2.3.2. REGISTER_SPATIAL_REF_SYS	7
2.3.3. UNREGISTER_GEOMETRY_COLUMNS	7
2.3.4. UNREGISTER_SPATIAL_REF_SYS	8
제3장 공간 인덱스	9
3.1. 공간 인덱스 생성	9
3.1.1. 공간 인덱스 제약 사항	10
3.2. 공간 인덱스 제거	10
제4장 Spatial 함수	11
4.1. ST_AREA	11
4.2. ST_ASBINARY	11
4.3. ST_ASTEXT	12
4.4. ST_BOUNDARY	13
4.5. ST_BUFFER	14
4.6. ST_CENTROID	14
4.7. ST_CONTAINS	15
4.8. ST_CONVEXHULL	16
4.9. ST_CROSSES	17
4.10. ST_DIFFERENCE	17
4.11. ST_DIMENSION	18
4.12. ST_DISJOINT	19
4.13. ST_DISTANCE	20
4.14. ST_ENDPOINT	21
4.15. ST_ENVELOPE	21
4.16. ST_EQUALS	22
4.17. ST_EXTERIORRING	23

4.18.	ST_GEOMCOLLFROMTEXT	24
4.19.	ST_GEOMCOLLFROMWKB	24
4.20.	ST_GEOMFROMTEXT	25
4.21.	ST_GEOMFROMWKB	26
4.22.	ST_GEOMETRYN	27
4.23.	ST_GEOMETRYTYPE	27
4.24.	ST_INTERIORRINGN	28
4.25.	ST_INTERSECTION	29
4.26.	ST_INTERSECTS	30
4.27.	ST_ISCLOSED	31
4.28.	ST_ISEMPTY	31
4.29.	ST_ISRING	32
4.30.	ST_ISSIMPLE	32
4.31.	ST_ISVALID	33
4.32.	ST_LENGTH	34
4.33.	ST_LINEFROMTEXT	34
4.34.	ST_LINEFROMWKB	35
4.35.	ST_MAXX	36
4.36.	ST_MAXY	37
4.37.	ST_MINX	37
4.38.	ST_MINY	38
4.39.	ST_MLINEFROMTEXT	39
4.40.	ST_MLINEFROMWKB	40
4.41.	ST_MPOINTFROMTEXT	40
4.42.	ST_MPOINTFROMWKB	41
4.43.	ST_MPOLYFROMTEXT	42
4.44.	ST_MPOLYFROMWKB	42
4.45.	ST_NUMGEOMETRIES	43
4.46.	ST_NUMINTERIORRING	44
4.47.	ST_NUMPOINTS	44
4.48.	ST_OVERLAPS	45
4.49.	ST_POINTFROMTEXT	46
4.50.	ST_POINTFROMWKB	47
4.51.	ST_POINTN	47
4.52.	ST_POINTONSURFACE	48
4.53.	ST_POLYFROMTEXT	49
4.54.	ST_POLYFROMWKB	49
4.55.	ST_RELATE	50
4.56.	ST_SRID	51
4.57.	ST_STARTPOINT	52
4.58.	ST_SYMDIFFERENCE	52
4.59.	ST_TOUCHES	53
4.60.	ST_UNION	54

4.61.	ST_WITHIN	55
4.62.	ST_X	56
4.63.	ST_Y	56
제5장	Spatial 유틸리티	59
5.1.	gisLoader	59
색인		61

안내서에 대하여

안내서의 대상

본 안내서는 Tibero[®](이하 Tibero)를 사용하여 Spatial Reference System을 생성하고 유지하는 관리자와 이를 이용하여 프로그램 작성하는 개발자를 기술한다.

안내서의 전제 조건

본 안내서는 Tibero의 Spatial 기능을 사용하고 유지하는 데 필요한 SQL 문장을 설명하는 안내서이다. 따라서 본 안내서를 원활히 이해하기 위해서는 다음과 같은 사항을 미리 알고 있어야 한다.

- 데이터베이스의 이해
- RDBMS의 이해
- SQL의 이해

안내서의 제한 조건

본 안내서는 Tibero를 실무에 적용하거나 운용하는 데 필요한 모든 사항을 포함하고 있지 않다. 따라서 설치, 환경설정 등 운용 및 관리에 대해서는 각 제품 안내서를 참고하기 바란다.

참고

Tibero의 설치 및 환경설정에 관한 내용은 "Tibero 설치 안내서"를 참고한다.

안내서 구성

Tibero Spatial 참조 안내서는 총 5개의 장으로 구성되어 있다.

각 장의 주요 내용은 다음과 같다.

- 제1장: Spatial 소개

기본 개념에 대해 기술한다.

- 제2장: Spatial 구조

Tibero Spatial과 관련된 스키마 객체에 대하여 기술한다.

- 제3장: 공간 인덱스

Tibero Spatial에서 제공하는 공간 인덱스에 대해 기술한다.

- 제4장: Spatial 함수

Tibero Spatial에서 제공하는 함수들에 대해 기술한다.

- 제5장: Spatial 유틸리티

Tibero Spatial에서 제공하는 유틸리티인 `gisLoader`의 사용법에 대해서 기술한다.

안내서 규약

표기	의미
<<AaBbCc123>>	프로그램 소스 코드의 파일명
<Ctrl>+C	Ctrl과 C를 동시에 누름
[Button]	GUI의 버튼 또는 메뉴 이름
진하게	강조
" "(따옴표)	다른 관련 안내서 또는 안내서 내의 다른 장 및 절 언급
'입력항목'	화면 UI에서 입력 항목에 대한 설명
하이퍼링크	메일 계정, 웹 사이트
>	메뉴의 진행 순서
+----	하위 디렉터리 또는 파일 있음
----	하위 디렉터리 또는 파일 없음
<u>참고</u>	참고 또는 주의사항
<u>주의</u>	주의할 사항
[그림 1.1]	그림 이름
[예 1.1]	예제 이름
AaBbCc123	Java 코드, XML 문서
[command argument]	옵션 파라미터
< xyz >	'<'와 '>' 사이의 내용이 실제 값으로 변경됨
	선택 사항. 예) A B: A나 B 중 하나
...	파라미터 등이 반복되어서 나옴
\${ }	환경변수

안내서 변경이력

[illegible]

시스템 사용 환경

	요구 사항
Platform	HP-UX 11i (IA64)
	Solaris (Solaris 10)
	AIX (PPC 5L/AIX 5.3)
	GNU (X86, 64, IA64)
	Linux kernel 3.10 이상
	Windows(x86) 64bit
Hardware	최소 2.5GB 하드디스크 공간
	4GB 이상 메모리 공간
Compiler	PSM (C99 지원 필요)
	tbESQL/C (C99 지원 필요)

관련 안내서

안내서	설명
Tibero 설치 안내서	설치에 필요한 시스템 요구사항과 설치 및 제거 방법을 기술한 안내서이다.
Tibero 관리자 안내서	Tibero의 동작과 주요 기능의 원활한 수행을 보장하기 위해 DBA가 알아야 할 관리 방법을 논리적 또는 물리적 측면에서 설명하고, 관리를 지원하는 각종 도구를 기술한 안내서이다.
Tibero 애플리케이션 개발자 안내서	각종 애플리케이션 라이브러리를 이용하여 애플리케이션 프로그램을 개발하는 방법을 기술한 안내서이다.
Tibero tbCLI 안내서	Call Level Interface인 tbCLI의 개념과 구성요소, 프로그램 구조를 소개하고 tbCLI 프로그램을 작성하는 데 필요한 사항을 기술한 안내서이다.
Tibero External Procedure 안내서	External Procedure를 소개하고 이를 생성하고 사용하는 방법을 기술한 안내서이다.
Tibero tbESQL/C 안내서	C 프로그래밍 언어를 사용해 데이터베이스 작업을 수행하는 각종 애플리케이션 프로그램을 작성하는 방법을 기술한 안내서이다.
Tibero tbPSM 안내서	저장 프러시저 모듈인 tbPSM의 개념과 문법, 구성요소를 소개하고, 프로그램을 작성하는 데 필요한 제어 구조, 복합 타입, 서브 프로그램, 패키지 및 SQL 문장을 실행하고 에러를 처리하는 방법을 기술한 안내서이다.
Tibero tbPSM 참조 안내서	저장 프러시저 모듈인 tbPSM의 패키지를 소개하고, 이러한 패키지에 포함된 각 프러시저와 함수의 프로토타입, 파라미터, 예제 등을 기술한 참조 안내서이다.
Tibero 유틸리티 안내서	데이터베이스와 관련된 작업을 수행하기 위해 필요한 유틸리티의 설치 및 환경설정, 사용 방법을 기술한 안내서이다.
Tibero TAS 안내서	Tibero Active Storage(TAS)를 사용해서 Tibero의 파일을 관리하고자 하는 관리자를 대상으로 기술한 안내서이다.
Tibero 에러 안내서	Tibero를 사용하는 도중에 발생할 수 있는 각종 에러의 원인과 해결 방법을 기술한 안내서이다.
Tibero Spatial 참조 안내서	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한 안내서이다.

안내서	설명
Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성하고 사용하는 방법을 기술하는 안내서이다.

제1장 Spatial 소개

본 장에서는 Tiberio Spatial의 기본적인 개념을 설명한다.

1.1. 개요

Tiberio Spatial은 GIS, 회로도 등의 공간 데이터를 저장하고 이를 활용하기 위한 기능을 제공한다. SQL-MM 표준에 따른 여러 공간 질의를 지원하며, 성능 향상을 위한 공간 인덱싱 또한 지원한다.

1.2. GEOMETRY 객체

Tiberio Spatial에서는 SQL-MM 표준에 따른 7가지 타입의 GEOMETRY 객체와 Empty Type을 지원한다.

다음은 Tiberio Spatial에서 지원하는 GEOMETRY 타입이다.

이름	설명
POINT	0차원, 1개의 점으로 구성된 객체이다.
LINESTRING	2개 이상의 점으로 구성된 객체이다.
POLYGON	1개의 외부 링과 0개 이상의 내부 링으로 구성된 객체이다. 링은 시작점과 끝점이 같고, 내부에 교차점이 없는 라인 스트링을 의미한다. 내부 링끼리의 교집합은 POINT이거나 없어야 한다.
MULTIPOINT	0차원, 1개 이상의 점으로 구성된 객체이다.
MULTILINESTRING	1개 이상의 라인 스트링으로 구성된 객체이다.
MULTIPOLYGON	1개 이상의 폴리곤으로 구성된 객체이다.
GEOMETRYCOLLECTION	1개 이상의 GEOMETRY 객체로 구성된 객체이다.
EMPTY	비어있는 GEOMETRY 객체를 나타낸다. 사용자가 입력할 수 없으며 결과값으로 나타난다.

제2장 Spatial 구조

본 장에서는 Tiberio Spatial과 관련된 스키마 객체에 대하여 설명한다.

2.1. 테이블

다음은 관련 테이블 목록이다.

테이블	설명
GEOMETRY_COLUMNS_BASE	데이터베이스에서 사용되는 GEOMETRY COLUMN의 메타데이터를 저장하는 테이블이다.
SPATIAL_REF_SYS_BASE	데이터베이스에서 사용된 Spatial Reference System의 메타데이터를 저장하는 테이블이다.

2.1.1. GEOMETRY_COLUMNS_BASE

데이터베이스에서 사용된 Spatial Reference System의 메타데이터를 저장하는 테이블이다.

- 컬럼

컬럼	데이터 타입	설명
F_TABLE_SCHEMA	VARCHAR(30)	GEOMETRY 컬럼이 속한 TABLE의 소유자이다.
F_TABLE_NAME	VARCHAR(30)	GEOMETRY 컬럼이 속한 TABLE의 이름이다.
F_GEOMETRY_COLUMN	VARCHAR(30)	GEOMETRY 컬럼의 이름이다.
COORD_DIMENSION	INTEGER	Spatial Reference System의 DIMENSION이다.
SRID	INTEGER	Spatial Reference System의 데이터베이스 내에서의 ID로 SPATIAL_REF_SYS_BASE 테이블의 외부 키이다.
F_GEOMETRY_TYPE	VARCHAR(30)	GEOMETRY 컬럼에 있는 GEOMETRY 객체의 TYPE이다.
F_MBR	VARCHAR(256)	GEOMETRY 컬럼에 있는 GEOMETRY 객체들의 MBR(Minimum Bounding Rectangle)이다.

- 참조

[ALL_GEOMETRY_COLUMNS](#), [USER_GEOMETRY_COLUMNS](#)

2.1.2. SPATIAL_REF_SYS_BASE

데이터베이스에서 사용된 Spatial Reference System의 메타데이터를 저장하는 테이블이다.

- 컬럼

컬럼	데이터 타입	설명
SRID	INTEGER	Spatial Reference System의 데이터베이스 내에서의 ID이다.
AUTH_NAME	VARCHAR(256)	Spatial Reference System에서 사용된 표준의 이름이다.
AUTH_SRID	NUMBER	표준에 의해 정의된 Spatial Reference System의 ID이다.
SRTEXT	VARCHAR(2000)	Spatial Reference System의 Well-Known Text 표현이다.

- 참조

[SPATIAL_REF_SYS](#)

2.2. 뷰

본 절에서는 Spatial Reference System의 메타데이터 테이블을 저장하는 테이블과 연결된 뷰에 대하여 설명한다.

다음은 관련 뷰 목록이다.

뷰	설명
ALL_GEOMETRY_COLUMNS	GEOMETRY_COLUMNS_BASE에 등록되어 있는 모든 GEOMETRY COLUMN을 나열한다.
SPATIAL_REF_SYS	데이터베이스에서 사용된 Spatial Reference System 정보를 나열한다.
USER_GEOMETRY_COLUMNS	GEOMETRY_COLUMNS_BASE에 등록되어 있는 현재 사용자의 GEOMETRY COLUMN을 나열한다.

2.2.1. ALL_GEOMETRY_COLUMNS

ALL_GEOMETRY_COLUMNS는 GEOMETRY_COLUMNS_BASE에 등록되어 있는 모든 GEOMETRY COLUMN을 나열하는 뷰이다.

- 컬럼

GEOMETRY_COLUMNS_BASE 테이블과 같다.

- 참조

2.2.2. SPATIAL_REF_SYS

SPATIAL_REF_SYS는 데이터베이스에서 사용된 Spatial Reference System 정보를 나열하는 뷰이다.

- 컬럼

SPATIAL_REF_SYS_BASE 테이블과 같다.

- 참조

[SPATIAL_REF_SYS_BASE](#)

2.2.3. USER_GEOMETRY_COLUMNS

USER_GEOMETRY_COLUMNS은 GEOMETRY_COLUMNS_BASE에 등록되어 있는 현재 사용자의 GEOMETRY COLUMN을 나열하는 뷰이다.

Instead of Trigger가 구현되어 USER_GEOMETRY_COLUMNS를 통하여 GEOMETRY_COLUMNS_BASE 테이블에 대한 입력/수정/삭제가 가능하다.

트리거	설명
GEOMETRY_COLUMNS_DEL	USER_GEOMETRY_COLUMNS 뷰에 로우를 삭제할 경우 GEOMETRY_COLUMNS_BASE 테이블에서 현재 사용자의 로우를 삭제한다.
GEOMETRY_COLUMNS_INS	USER_GEOMETRY_COLUMNS 뷰에 로우를 입력할 경우 GEOMETRY_COLUMNS_BASE 테이블에서 해당 로우가 현재 사용자 정보와 함께 입력된다.
GEOMETRY_COLUMNS_UPT	USER_GEOMETRY_COLUMNS 뷰에 로우를 수정할 경우 GEOMETRY_COLUMNS_BASE 테이블에서 현재 사용자의 로우를 수정한다.

- 컬럼

ALL_GEOMETRY_COLUMNS과 같다. 단, F_TABLE_SCHEMA 컬럼은 예외다.

- 참조

[GEOMETRY_COLUMNS_BASE](#), [ALL_GEOMETRY_COLUMNS](#)

2.3. 프러시저

본 절에서는 Spatial Reference System의 메타데이터 테이블을 관리하기 위한 프러시저를 사용하는 방법을 설명한다.

다음은 Spatial에서 제공하는 프러시저 목록이다.

프러시저	설명
REGISTER_GEOMETRY_COLUMNS	GEOMETRY_COLUMNS_BASE 테이블에 Geometry Column의 메타데이터를 등록한다.
REGISTER_SPATIAL_REF_SYS	SPATIAL_REF_SYS_BASE 테이블에 Spatial Reference System 메타데이터를 등록한다.
UNREGISTER_GEOMETRY_COLUMNS	GEOMETRY_COLUMNS_BASE 테이블에 등록한 Geometry Column의 메타데이터를 삭제한다.
UNREGISTER_SPATIAL_REF_SYS	SPATIAL_REF_SYS_BASE 테이블에 등록한 Spatial Reference System 메타데이터를 삭제한다.

2.3.1. REGISTER_GEOMETRY_COLUMNS

GEOMETRY_COLUMNS_BASE 테이블에 Geometry Column의 메타데이터를 등록하는 프러시저이다.

REGISTER_GEOMETRY_COLUMNS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
SYSGIS.REGISTER_GEOMETRY_COLUMNS
(
    TABLE_NAME      IN   VARCHAR2,
    GEOMETRY_COLUMN  IN   VARCHAR2,
    COORD_DIMENSION  IN   INTEGER,
    SRID              IN   INTEGER
);
```

- 파라미터

파라미터	설명
TABLE_NAME	GEOMETRY 컬럼이 속한 TABLE의 이름이다.
GEOMETRY_COLUMN	GEOMETRY 컬럼의 이름이다.
COORD_DIMENSION	Spatial Reference System의 DIMENSION이다.
SRID	Spatial Reference System의 데이터베이스 내에서의 ID로 SPATIAL_REF_SYS 테이블의 외부 키이다.

2.3.2. REGISTER_SPATIAL_REF_SYS

SPATIAL_REF_SYS_BASE 테이블에 Spatial Reference System 메타데이터를 등록하는 프러시저이다.

REGISTER_SPATIAL_REF_SYS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
SYSGIS.REGISTER_SPATIAL_REF_SYS
(
    SRID      IN INTEGER,
    AUTH_NAME IN VARCHAR2,
    AUTH_SRID IN INTEGER,
    SRTEXT    IN VARCHAR2
);
```

- 파라미터

파라미터	설명
SRID	Spatial Reference System의 데이터베이스 내에서의 ID이다.
AUTH_NAME	Spatial Reference System에서 사용된 표준의 이름이다.
AUTH_SRID	표준에 의해 정의된 Spatial Reference System의 ID이다.
SRTEXT	Spatial Reference System의 Well-Known Text 표현이다.

2.3.3. UNREGISTER_GEOMETRY_COLUMNS

GEOMETRY_COLUMNS_BASE 테이블에 등록한 Geometry Column의 메타데이터를 삭제하는 프러시저이다.

UNREGISTER_GEOMETRY_COLUMNS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
SYSGIS.UNREGISTER_GEOMETRY_COLUMNS
(
    TABLE_NAME      IN VARCHAR2,
    GEOMETRY_COLUMN IN VARCHAR2
);
```

- 파라미터

파라미터	설명
TABLE_NAME	GEOMETRY 컬럼이 속한 TABLE의 이름이다.
GEOMETRY_COLUMN	GEOMETRY 컬럼의 이름이다.

2.3.4. UNREGISTER_SPATIAL_REF_SYS

SPATIAL_REF_SYS_BASE 테이블에 등록된 Spatial Reference System 메타데이터를 삭제하는 프러시저이다.

UNREGISTER_SPATIAL_REF_SYS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
SYSGIS.UNREGISTER_SPATIAL_REF_SYS  
(  
    AUTH_NAME IN VARCHAR2  
);
```

- 파라미터

파라미터	설명
AUTH_NAME	삭제 할 Spatial Reference System에서 사용된 표준의 이름이다.

제3장 공간 인덱스

Tibero Spatial에서는 공간 질의 성능 향상을 위해 RTREE 방식으로 구현된 공간 인덱스를 제공한다.

3.1. 공간 인덱스 생성

공간 인덱스 생성 방법과 제약에 대해서 설명한다.

- 사용법

```
CREATE INDEX index_name on [schema_name.]
      table_name ON col_name RTREE
```

이름	설명
<index_name>	생성할 공간인덱스의 이름이다.
schema_name	인덱스를 생성할 대상 테이블의 소유자이다.
table_name	인덱스를 생성할 대상 테이블을 설정한다.
col_name	인덱스를 생성할 대상을 설정한다. 컬럼은 GEOMETRY TYPE이어야 한다.
RTREE	RTREE 인덱스로 생성함을 의미하는 예약어를 설정한다.

- 예제

```
SQL>CREATE TABLE GIS (ID INTEGER PRIMARY KEY, GEOM GEOMETRY);
Table 'GIS' created.
SQL>CREATE INDEX RT_IDX_GIS ON GIS(GEOM) RTREE;
Index 'RT_IDX_GIS' created.

INSERT INTO GIS VALUES (101, ST_GEOMFROMTEXT('POINT(1 1)'));
INSERT INTO GIS VALUES (102, ST_GEOMFROMTEXT('MULTIPOINT(1 1, 2 2)'));
INSERT INTO GIS VALUES (103, ST_GEOMFROMTEXT('LINESTRING(1 1, 2 2)'));
INSERT INTO GIS VALUES (104, ST_GEOMFROMTEXT('MULTILINESTRING((1 1, 2 2),
(3 3, 4 5))'));
INSERT INTO GIS VALUES (105, ST_GEOMFROMTEXT('POLYGON((1 1, 2 1, 2 2, 1 2,
1 1))'));
INSERT INTO GIS VALUES (106, ST_GEOMFROMTEXT('POLYGON((0 0, 0 12, 12 12, 12 0,0 0),
(6 10, 6 11, 9 11, 9 10,6 10), (6 3, 6 6, 9 6, 9 3,6 3))'));
INSERT INTO GIS VALUES (107, ST_GEOMFROMTEXT('MULTIPOLYGON(((1 1, 2 1, 2 2, 1 2, 1 1)),
((3 3, 3 5, 5 5, 5 3, 3 3))'));
INSERT INTO GIS VALUES (108, ST_GEOMFROMTEXT('GEOMETRYCOLLECTION(POINT(1 1),
LINESTRING(2 2, 3 3))'));
```

```
INSERT INTO GIS VALUES (109, ST_BOUNDARY(ST_GEOMFROMTEXT('POINT(10 10)')));
INSERT INTO GIS VALUES (110, ST_GEOMFROMTEXT('GEOMETRYCOLLECTION(POINT(1 1),
LINESTRING(2 2, 3 3))'));
COMMIT;
```

3.1.1. 공간 인덱스 제약 사항

공간 인덱스는 다음과 같은 경우에 제약을 가진다.

- 여러 컬럼에 대해 하나의 인덱스를 걸 수 없다.
- 공간 인덱스는 파티션될 수 없다.
- 공간 인덱스는 GEOMETRY TYPE이 아닌 컬럼에 대하여 생성할 수 없다.
- SQL의 WHERE 절에 다음의 함수를 구현하였을 때만 공간 인덱스의 사용이 가능하다.
 - ST_EQUALS
 - ST_TOUCHES
 - ST_WITHIN
 - ST_OVERLAPS
 - ST_CROSSES
 - ST_INTERSECTS
 - ST_CONTAINS

3.2. 공간 인덱스 제거

공간 인덱스의 제거는 다른 인덱스와 동일하다.

- 사용법

```
DROP INDEX index_name
```

이름	설명
<i>index_name</i>	삭제할 공간 인덱스의 이름이다.

제4장 Spatial 함수

본 장에서는 Tiberio Spatial에서 제공되는 함수에 대해 설명한다.

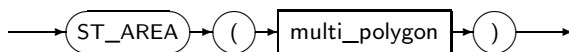
4.1. ST_AREA

ST_AREA은 멀티 폴리곤의 면적을 반환하는 함수이다.

ST_AREA의 세부 내용은 다음과 같다.

- 문법

st_area



- 구성요소

구성요소	설명
multi_polygon	멀티 폴리곤 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_AREA 함수를 사용하는 예이다.

```
SQL> SELECT ST_AREA(GEOM), ST_ASTEXT(GEOM)FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTIPOLYGON';
```

```
ST_AREA(GEOM) ST_ASTEXT(GEOM)
```

```
-----
5 MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
175 MULTIPOLYGON(((10 10,10 20,20 20,20 15,10 10)),((60 60,70 70,80 60,60 60)))
```

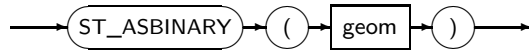
4.2. ST_ASBINARY

ST_ASBINARY는 GEOMETRY 객체를 WKB(Well Known Binary)형태로 반환하는 함수이다.

ST_ASBINARY의 세부 내용은 다음과 같다.

- 문법

st_asbinary



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ASBINARY 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM), ST_ASBINARY(GEOM) FROM GIS;
```

ST_ASTEXT(GEOM)	ST_ASBINARY(GEOM)
POINT(1 1)	010100000000000000000000F03F000000000000F03F
MULTIPOINT(1 1,2 2)	01040000000200000001010000000000000000F0...
LINESTRING(1 1,2 2)	01020000000200000000000000000000F03F00000000...
POLYGON((1 1,2 1,2 2,1 2,1 1))	01030000000100000005000000000000000000F03F...
EMPTY	0108000000
POINT(10 10)	0101000000000000000000002440000000000002440

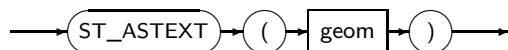
4.3. ST_ASTEXT

ST_ASTEXT는 GEOMETRY 객체를 WKT(Well Known Text) 형태로 반환하는 함수이다.

ST_ASTEXT의 세부 내용은 다음과 같다.

- 문법

st_astext



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ASTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM) FROM GIS

ST_ASTEXT(GEOM)
-----
POINT(1 1)
MULTIPOINT(1 1,2 2)
LINESTRING(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))
POLYGON((1 1,2 1,2 2,1 2,1 1))
MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
EMPTY
```

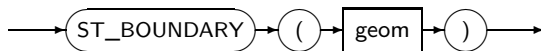
4.4. ST_BOUNDARY

ST_BOUNDARY는 주어진 **GEOMETRY** 객체의 경계를 반환하는 함수이다.

ST_BOUNDARY의 세부 내용은 다음과 같다.

- 문법

st_boundary



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 **ST_BOUNDARY** 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM),ST_ASTEXT(ST_BOUNDARY(GEOM)) FROM GIS;

ST_ASTEXT(GEOM)                                ST_ASTEXT(ST_BOUNDARY(GEOM))
-----
POINT(1 1)                                     EMPTY
MULTIPOINT(1 1,2 2)                           EMPTY
LINESTRING(1 1,2 2)                           MULTIPOINT(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))MULTIPOINT(1 1,2 2,3 3,4 5)
POLYGON((1 1,2 1,2 2,1 2,1 1))                LINESTRING(1 1,2 1,2 2,1 2,1 1)
```

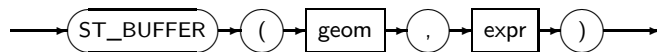
4.5. ST_BUFFER

ST_BUFFER는 주어진 GEOMETRY 객체로부터 일정 거리 안에 있는 모든 점을 표현하는 GEOMETRY 객체를 반환하는 함수이다.

ST_BUFFER의 세부 내용은 다음과 같다.

- 문법

st_buffer



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
expr	임의의 연산식이며 주어진 GEOMETRY 객체로부터의 거리를 의미한다.

- 예제

다음은 ST_BUFFER 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM), ST_ASTEXT(ST_BUFFER(GEOM, 10))  
       FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POINT';
```

```
ST_ASTEXT(GEOM) ST_ASTEXT(ST_BUFFER(GEOM, 10))  
-----
```

```
POINT(1 1)      POLYGON((11 1, 10 -0, 10 -2, ... , 6 9, 8 8, 9 6, 10 4, 10 2, 11 1))
```

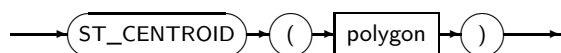
4.6. ST_CENTROID

ST_CENTROID는 주어진 GEOMETRY 객체의 중심을 반환하는 함수이다.

ST_CENTROID의 세부 내용은 다음과 같다.

- 문법

st_centroid



- 구성요소

구성요소	설명
polygon	폴리곤 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_CENTROID 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM),ST_ASTEXT(ST_CENTROID(GEOM))
      FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON';
```

ST_ASTEXT(GEOM)	ST_ASTEXT(ST_CENTROID(GEOM))
POLYGON((1 1,2 1,2 2,1 2,1 1))	POINT(1.5 1.5)
POLYGON((10 10,10 20,20 20,20 15,10 10))	POINT(14.4 16.1)

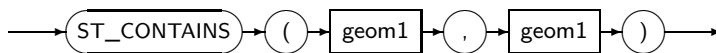
4.7. ST_CONTAINS

ST_CONTAINS는 GEOMETRY 객체 2가 GEOMETRY 객체 1의 외부에 존재하지 않고, 객체 1의 내부에 하나 이상의 포인트가 존재할 때만 1을 반환하는 함수이다. 두 GEOMETRY 객체 모두 GEOMETRY 객체가 아니어야 한다.

ST_CONTAINS의 세부 내용은 다음과 같다.

- 문법

st_contains



- 구성요소

구성요소	설명
geom1	포함하는 GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	포함되는 GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_CONTAINS 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM) FROM GIS A,GIS B
      WHERE ST_CONTAINS(A.GEOM,B.GEOM) >0
      AND ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
-------------------	-------------------

```

-----
MULTIPOINT(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
POLYGON((0 0,0 12,12 12 12,12 0,0 0)...)
-----
POINT(1 1)
LINESTRING(1 1,2 2)
LINESTRING(1 1,2 2)
POINT(1 1)
MULTIPOINT(1 1,2 2)
LINESTRING(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))
POLYGON((1 1,2 1,2 2,1 2,1 1))
MULTIPOLYGON(((1 1,2 1,2 2,1 2, ...

```

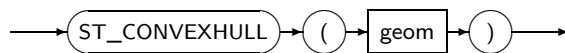
4.8. ST_CONVEXHULL

ST_CONVEXHULL은 주어진 GEOMETRY의 convex hull을 반환하는 함수이다. convex hull은 GEOMETRY 객체를 포함하는 제일 작은 볼록한 폐곡선을 의미한다.

ST_CONVEXHULL의 세부 내용은 다음과 같다.

- 문법

st_convexhull



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_CONVEXHULL 함수를 사용하는 예이다.

```

SQL> SELECT ST_ASTEXT(GEOM),ST_ASTEXT(ST_CONVEXHULL(GEOM)) FROM GIS;

ST_ASTEXT(GEOM)                                ST_ASTEXT(ST_CONVEXHULL(GEOM))
-----
POINT(1 1)                                     POINT(1 1)
MULTIPOINT(1 1,2 2)                           LINESTRING(1 1,2 2)
LINESTRING(1 1,2 2)                           LINESTRING(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))           POLYGON((1 1,4 5,3 3,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))                 POLYGON((1 1,1 2,2 2,2 1,1 1))
POLYGON((0 0,0 12,12 12 12,12 0,0 0),...)     POLYGON((0 0,0 12,12 12 12,12 0,0 0))

```

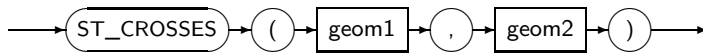
4.9. ST_CROSSES

ST_CROSSES는 주어진 두 GEOMETRY 객체가 교차하면 1을 반환하는 함수이다. 포함관계에 있거나, 경계가 닿아 있는 경우는 해당하지 않는다.

ST_CROSSES의 세부 내용은 다음과 같다.

- 문법

st_crosses



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_CROSSES 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM) FROM GIS A,GIS B
      WHERE ST_CROSSES(A.GEOM,B.GEOM) >0
      AND ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
MULTILINESTRING((1 1,2 2),(3 3,4 5))	POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))	MULTILINESTRING((1 1,2 2),(3 3,4 5))
POLYGON((0 0,0 12,12 12,12 0,0 0),...	MULTIPOINT(10 10,20 20)
POLYGON((0 0,0 12,12 12,12 0,0 0),...	LINESTRING(10 10,20 20,30 40)
MULTILINESTRING((10 10,20 20),(15 ...	MULTIPOINT(10 10,20 20)

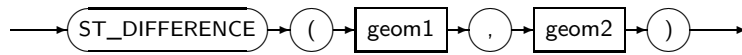
4.10. ST_DIFFERENCE

ST_DIFFERENCE는 GEOMETRY 객체 2와 겹치지 않는 객체 1의 부분을 반환하는 함수이다.

ST_DIFFERENCE의 세부 내용은 다음과 같다.

- 문법

st_difference



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_DIFFERENCE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM), ST_ASTEXT(B.GEOM),
          ST_ASTEXT(ST_DIFFERENCE(A.GEOM,B.GEOM)) FROM GIS A,GIS B
WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_DIMENSION(A.GEOM) = ST_DIMENSION(B.GEOM);
```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)	ST_DIFFERENCE(...
-----	-----	-----
POINT(1 1)	MULTIPOINT(1 1,2 2)	EMPTY
LINESTRING(1 1,2 2)	MULTILINESTRING((1 1,2 2),(3 3,4 5))	EMPTY

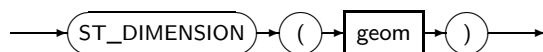
4.11. ST_DIMENSION

ST_DIMENSION은 GEOMETRY 객체의 차원을 반환하는 함수이다.

ST_DIMENSION의 세부 내용은 다음과 같다.

- 문법

st_dimension



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_DIMENSION 함수를 사용하는 예이다.

```
SQL> SELECT ST_DIMENSION(GEOM), ST_ASTEXT(geom) FROM GIS;

ST_DIMENSION(GEOM)  ST_ASTEXT(GEOM)
-----
0  POINT(1 1)
0  MULTIPOINT(1 1,2 2)
1  LINESTRING(1 1,2 2)
1  MULTILINESTRING((1 1,2 2),(3 3,4 5))
2  POLYGON((1 1,2 1,2 2,1 2,1 1))
2  POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,6 11,9 11,9 10,6 10))
2  MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
1  GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
-1  EMPTY
```

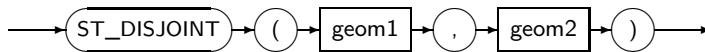
4.12. ST_DISJOINT

ST_DISJOINT는 두 GEOMETRY 객체가 어떤 영역도 공유하지 않을 때 1을 반환하는 함수이다.

ST_DISJOINT의 세부 내용은 다음과 같다.

- 문법

st_disjoint



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_DISJOINT 함수를 사용하는 예이다.

```
SQL> SELECT ST_DISJOINT(A.GEOM,B.GEOM), ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
FROM GIS A,GIS B
WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

DISJOINT	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
0	POINT(1 1)	MULTIPOINT(1 1,2 2)
0	POINT(1 1)	LINESTRING(1 1,2 2)

1	POINT(1 1)	EMPTY
1	POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
0	POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11, ...
0	MULTIPOINT(1 1,2 2)	POINT(1 1)

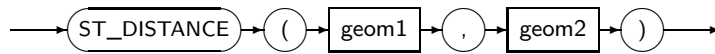
4.13. ST_DISTANCE

ST_DISTANCE는 두 GEOMETRY 객체사이의 최단 거리를 반환하는 함수이다.

ST_DISTANCE의 세부 내용은 다음과 같다.

- 문법

st_distance



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_DISTANCE 함수를 사용하는 예이다.

```

SQL> SELECT ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM),ST_DISTANCE(A.GEOM,B.GEOM)
      FROM GIS A,GIS B
      WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_DIMENSION(a.Geom) = ST_DIMENSION(b.geom);

```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)	ST_DISTANCE
MULTIPOINT(1 1,2 2)	POINT(1 1)	0
POINT(10 10)	POINT(1 1)	12.72792
MULTIPOINT(10 10,20 20)	POINT(1 1)	12.72792
POINT(1 1)	MULTIPOINT(1 1,2 2)	0
POINT(10 10)	MULTIPOINT(1 1,2 2)	11.31371
LINestring(10 10,20 20,30 40)	LINestring(1 1,2 2)	11.31371

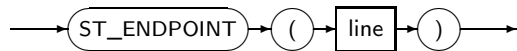
4.14. ST_ENDPOINT

ST_ENDPOINT는 라인 스트링 GEOMETRY 객체의 마지막 포인트를 반환하는 함수이다.

ST_ENDPOINT의 세부 내용은 다음과 같다.

- 문법

st_endpoint



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ENDPOINT 함수를 사용하는 예이다.

```
SQL> SELECT ST_astext(geom),ST_ASTEXT(ST_ENDPOINT(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';
```

ST_ASTEXT(GEOM)	ST_ASTEXT(ST_ENDPOINT(GEOM))
LINESTRING(1 1,2 2)	POINT(2 2)
LINESTRING(10 10,20 20,30 40)	POINT(30 40)

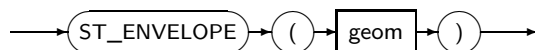
4.15. ST_ENVELOPE

ST_ENVELOPE는 주어진 GEOMETRY 객체를 둘러싸는 사각형을 GEOMETRY 형태로 반환하는 함수이다.

ST_ENVELOPE의 세부 내용은 다음과 같다.

- 문법

st_envelope



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ENVELOPE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_ENVELOPE(GEOM)) FROM GIS;

ST_ASTEXT(ST_ENVELOPE(GEOM))
-----
POLYGON((1 1,1 1,1 1,1 1,1 1))
POLYGON((1 1,1 2,2 2,2 1,1 1))
POLYGON((1 1,1 2,2 2,2 1,1 1))
POLYGON((1 1,1 5,4 5,4 1,1 1))
POLYGON((1 1,1 2,2 2,2 1,1 1))
POLYGON((0 0,0 12,12 12,12 0,0 0))
POLYGON((1 1,1 5,5 5,5 1,1 1))
POLYGON((1 1,1 3,3 3,3 1,1 1))
EMPTY
```

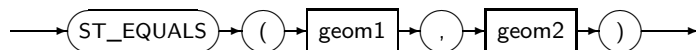
4.16. ST_EQUALS

ST_EQUALS는 두 GEOMETRY 객체가 같은 GEOMETRY를 표현하면 1을 반환하는 함수이다.

ST_EQUALS의 세부 내용은 다음과 같다.

- 문법

st_equals



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_EQUALS 함수를 사용하는 예이다.

```
SQL> SELECT ST_EQUALS(A.GEOM,B.GEOM), ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
FROM GIS A,GIS B
```

```

WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';

ST_EQUALS ST_ASTEXT(A.GEOM) ST_ASTEXT(B.GEOM)
-----
0 POINT(1 1) MULTIPOINT(1 1,2 2)
0 POINT(1 1) LINESTRING(1 1,2 2)
0 MULTIPOINT(1 1,2 2) POINT(1 1)
0 EMPTY EMPTY

```

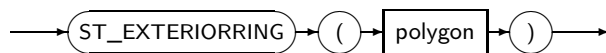
4.17. ST_EXTERIORRING

ST_EXTERIORRING은 폴리곤 객체의 외부 링을 반환하는 함수이다.

ST_EXTERIORRING의 세부 내용은 다음과 같다.

- 문법

st_exteriorring



- 구성요소

구성요소	설명
polygon	폴리곤 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_EXTERIORRING 함수를 사용하는 예이다.

```

SQL> SELECT ST_ASTEXT(ST_EXTERIORRING(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON';

ST_ASTEXT(ST_EXTERIORRING(GEOM))
-----
LINESTRING(1 1,2 1,2 2,1 2,1 1)
LINESTRING(0 0,0 12,12 12,12 0,0 0)
LINESTRING(10 10,10 20,20 20,20 15,10 10)
LINESTRING(1 1,1 13,13 13,13 1,1 1)

```

4.18. ST_GEOMCOLLFROMTEXT

ST_GEOMCOLLFROMTEXT는 주어진 WKT를 컬렉션 객체로 반환하는 함수이다.

ST_GEOMCOLLFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_geomcollfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 컬렉션 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_GEOMCOLLFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_GEOMCOLLFROMTEXT('GEOMETRYCOLLECTION(POINT(1 1),
    LINESTRING(2 2,3 3))') FROM DUAL;

ST_GEOMCOLLFROMTEXT('GEOMETRYCOLLECTION(POINT(11),LINESTRING(22,33))')
-----
0000000000000F03F000000000000F03F0000000000000840000000000000840070000...
```

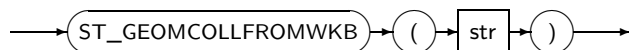
4.19. ST_GEOMCOLLFROMWKB

ST_GEOMCOLLFROMWKB는 주어진 WKB를 컬렉션 객체로 반환하는 함수이다.

ST_GEOMCOLLFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_geomcollfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 컬렉션 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_GEOMCOLLFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_GEOMCOLLFROMWKB(ST_ASBINARY(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'GEOMETRYCOLLECTION';

ST_GEOMCOLLFROMWKB(ST_ASBINARY(GEOM))
-----
0000000000000F03F000000000000F03F000000000000084000000000000008400700000002...
00000000000024400000000000002440000000000003E400000000000003E400700000003...
```

4.20. ST_GEOMFROMTEXT

ST_GEOMFROMTEXT는 주어진 WKT를 GEOMETRY 객체로 반환하는 함수이다.

ST_GEOMFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_geomfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_GEOMFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_GEOMFROMTEXT('MULTIPOINT(1 1,2 2)') FROM DUAL;

ST_GEOMFROMTEXT('MULTIPOINT(11,22)')
-----
0000000000000F03F000000000000F03F0000000000000400000000000000400400000002...
```

4.21. ST_GEOMFROMWKB

ST_GEOMFROMWKB는 주어진 WKB를 GEOMETRY 객체로 반환하는 함수이다.

ST_GEOMFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_geomfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하기 위한 WKB 형식이 저장되어 있는 바이너리 타입 (BLOB 또는 RAW) 값이다.

- 예제

다음은 ST_GEOMFROMWKB 함수를 사용하는 예이다.

```
SQL>SELECT ST_GEOMFROMWKB(ST_ASBINARY(GEOM))
      FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) NOT LIKE 'EMPTY';
```

```
ST_GEOMFROMWKB(ST_ASBINARY(GEOM))
```

```
-----
000000000000F03F000000000000F03F000000000000F03F000000000000F03F010000000000...
000000000000F03F000000000000F03F00000000000004000000000000004004000000020...
000000000000F03F000000000000F03F00000000000004000000000000004002000000020...
000000000000F03F000000000000F03F000000000000104000000000000144005000000020...
000000000000F03F000000000000F03F00000000000004000000000000004003000000010...
0000000000000000000000000000000000000000000000000000000028400000000000284003000000030...
000000000000F03F000000000000F03F000000000000144000000000000144006000000020...
000000000000F03F000000000000F03F000000000000084000000000000084007000000020...
0000000000002440000000000000244000000000000244000000000000244001000000000...
0000000000002440000000000000244000000000000344000000000000344004000000020...
00000000000024400000000000002440000000000003E4000000000000444002000000030...
00000000000024400000000000002440000000000003E4000000000000344005000000020...
0000000000002440000000000000244000000000000344000000000000344003000000010...
000000000000F03F000000000000F03F0000000000002A40000000000002A4003000000030...
0000000000002440000000000000244000000000000544000000000080514006000000020...
00000000000024400000000000002440000000000003E40000000000003E4007000000030...
```

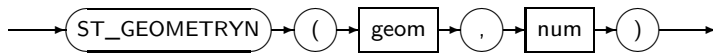
4.22. ST_GEOMETRYN

ST_GEOMETRYN은 GEOMETRY 컬렉션, 멀티 포인트, 멀티 라인 스트링, 멀티 폴리곤 객체에서 N번째 GEOMETRY 객체를 반환하는 함수이다.

ST_GEOMETRYN의 세부 내용은 다음과 같다.

- 문법

st_geometryn



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
num	N번째 객체를 지정한다.

- 예제

다음은 ST_GEOMETRYN 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMETRYN(GEOM,2)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) IN
      ('MULTIPOLYGON', 'MULTILINESTRING', 'MULTIPOINT', 'GEOMETRYCOLLECTION');
```

```
ST_ASTEXT(ST_GEOMETRYN(GEOM,2))
-----
POINT(2 2)
LINESTRING(3 3,4 5)
POLYGON((3 3,3 5,5 5,5 3,3 3))
LINESTRING(2 2,3 3)
POINT(20 20)
LINESTRING(15 15,30 15)
POLYGON((60 60,70 70,80 60,60 60))
POINT(30 30)
```

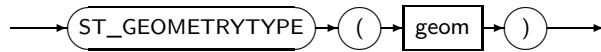
4.23. ST_GEOMETRYTYPE

ST_GEOMETRYTYPE은 GEOMETRY 객체의 타입을 반환하는 함수이다.

ST_GEOMETRYTYPE의 세부 내용은 다음과 같다.

- 문법

st_geometrytype



- 구성 요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_GEOMETRYTYPE 함수를 사용하는 예이다.

```
SQL> SELECT ST_GEOMETRYTYPE(GEOM) FROM GIS;
```

```
ST_GEOMETRYTYPE(GEOM)
```

```
-----
```

```
POINT  
MULTIPOINT  
LINESTRING  
MULTILINESTRING  
POLYGON  
POLYGON  
MULTIPOLYGON  
GEOMETRYCOLLECTION  
EMPTY
```

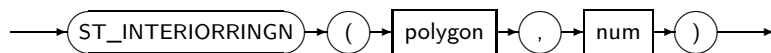
4.24. ST_INTERIORRINGN

ST_INTERIORRINGN은 폴리곤 객체의 N번째 내부 링을 반환하는 함수이다.

ST_INTERIORRINGN의 세부 내용은 다음과 같다.

- 문법

st_interiorringn



- 구성 요소

구성요소	설명
polygon	폴리곤 객체를 나타내는 GEOMETRY TYPE이어야 한다.
num	N번째 내부 링을 지정한다.

- 예제

다음은 ST_INTERIORRINGN 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_INTERIORRINGN(GEOM,1)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON'
      AND ST_NUMINTERIORRING(GEOM) > 1

ST_ASTEXT(ST_INTERIORRINGN(GEOM,1))
-----
LINESTRING(6 10,6 11,9 11,9 10,6 10)
LINESTRING(7 11,7 12,10 12,10 11,7 11)
```

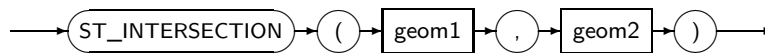
4.25. ST_INTERSECTION

ST_INTERSECTION은 GEOMETRY 객체 1과 GEOMETRY 객체 2가 공유하는 부분을 나타내는 GEOMETRY 객체를 반환하는 함수이다.

ST_INTERSECTION의 세부 내용은 다음과 같다.

- 문법

st_intersection



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_INTERSECTION 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM), ST_ASTEXT(B.GEOM),
      ST_ASTEXT(ST_INTERSECTION(A.GEOM,B.GEOM))
      FROM GIS A, GIS B
      WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_DIMENSION(A.GEOM) = ST_DIMENSION(B.GEOM);

ST_ASTEXT(A.GEOM)      ST_ASTEXT(B.GEOM)      ST_INTERSECTION
-----
POINT(1 1)             MULTIPOINT(1 1,2 2)      POINT(1 1)
```

```

LINESTRING(1 1,2 2)      MULTILINESTRING((1 1,2 2),(3 3,4 5))LINESTRING(1 1,2 2)
POLYGON((1 1,2 1,2 2,...POLYGON((0 0,0 12,12 12,12 0,0 0))) POLYGON((1 1,...

```

4.26. ST_INTERSECTS

ST_INTERSECTS는 두 GEOMETRY 객체가 공유하는 부분이 있을 때 1을 반환하는 함수이다. GEOMETRY 인자로 GEOMETRYCOLLECTION 타입이 올 경우에는 런타임 에러를 발생시킨다.

ST_INTERSECTS의 세부 내용은 다음과 같다.

- 문법

st_intersects



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_INTERSECTS 함수를 사용하는 예이다.

```

SQL> SELECT ST_INTERSECTS(A.GEOM,B.GEOM),
           ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
FROM GIS A,GIS B
WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';

```

	ST_INTERSECTS	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
1		POINT(1 1)	MULTIPOINT(1 1,2 2)
1		POINT(1 1)	LINESTRING(1 1,2 2)
0		POINT(1 1)	EMPTY
0		POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
1		POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1), ...
1		MULTIPOINT(1 1,2 2)	POINT(1 1)

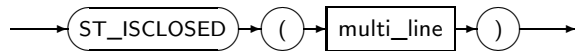
4.27. ST_ISCLOSED

ST_ISCLOSED는 라인 스트링 객체의 첫 포인트와 마지막 포인트가 동일할 때 1을 반환하는 함수이다.

ST_ISCLOSED의 세부 내용은 다음과 같다.

- 문법

st_isclosed



- 구성요소

구성요소	설명
multi_line	멀티 라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ISCLOSED 함수를 사용하는 예이다.

```
SQL> SELECT ST_ISCLOSED(GEOM), ST_ASTEXT(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTILINESTRING'

ST_ISCLOSED(GEOM) ST_ASTEXT(GEOM)
-----
0                MULTILINESTRING((1 1,2 2),(3 3,4 5))
0                MULTILINESTRING((10 10,20 20),(15 15,30 15))
```

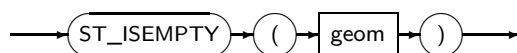
4.28. ST_ISEMPTY

ST_ISEMPTY는 주어진 GEOMETRY 객체가 EMPTY 타입일 때 1을 반환하는 함수이다.

ST_ISEMPTY의 세부 내용은 다음과 같다.

- 문법

st_isempty



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ISEMPY 함수를 사용하는 예이다.

```
SQL> SELECT ST_ISEMPY(GEOM), ST_ASTEXT(GEOM) FROM GIS

ST_ISEMPY(GEOM) ST_ASTEXT(GEOM)
-----
0                POINT(1 1)
1                EMPTY
```

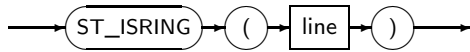
4.29. ST_ISRING

ST_ISRING은 주어진 라인 스트링이 닫혀있고(첫 포인트와 마지막 포인트가 동일) 심플하면 1을 반환하는 함수이다.

ST_ISRING의 세부 내용은 다음과 같다.

- 문법

st_isring



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ISRING 함수를 사용하는 예이다.

```
SQL> SELECT ST_ISRING(GEOM), ST_ASTEXT(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';

ST_ISRING(GEOM) ST_ASTEXT(GEOM)
-----
0                LINESTRING(1 1,2 2)
0                LINESTRING(10 10,20 20,30 40)
```

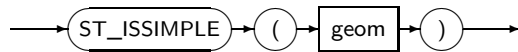
4.30. ST_ISSIMPLE

ST_ISSIMPLE은 GEOMETRY 객체가 자체 교차점이나, 자체 접촉점을 가지지 않으면 1을 반환하는 함수이다.

ST_ISSIMPLE의 세부 내용은 다음과 같다.

- 문법

st_issimple



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ISSIMPLE 함수를 사용하는 예이다.

```

SQL> SELECT ST_ISSIMPLE(GEOM), ST_ASTEXT(GEOM) FROM GIS;

ST_ISSIMPLE(GEOM) ST_ASTEXT(GEOM)
-----
1                MULTIPOINT(10 10,20 20)
1                LINESTRING(10 10,20 20,30 40)
0                MULTILINESTRING((10 10,20 20),(15 15,30 15))
  
```

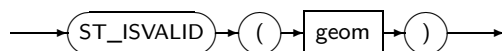
4.31. ST_ISVALID

ST_ISVALID는 GEOMETRY 객체가 공간적으로 유효한지(well-formed 인지)를 검사하는 함수이다. 공간적으로 유효하면 1, 아니면 0을 반환한다. EMPTY 공간 객체에 대해서는 1을 반환한다.

ST_ISVALID의 세부 내용은 다음과 같다.

- 문법

st_isvalid



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_ISVALID 함수를 사용하는 예이다.

```
SQL> SELECT ST_ISVALID(ST_GEOMFROMTEXT('POLYGON((0 0, 1 1, 0 1, 1 0, 0 0))')) BAD,
        ST_ISVALID(ST_GEOMFROMTEXT('POLYGON((0 0, 0 1, 1 1, 1 0, 0 0))')) GOOD
FROM DUAL;

        BAD        GOOD
-----
          0          1
```

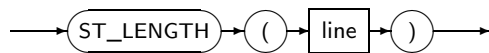
4.32. ST_LENGTH

ST_LENGTH는 라인 스트링 객체의 길이를 반환하는 함수이다.

ST_LENGTH의 세부 내용은 다음과 같다.

- 문법

st_length



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_LENGTH 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM), ST_LENGTH(GEOM) FROM GIS
        WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';

ST_ASTEXT(GEOM)                ST_LENGTH(GEOM)
-----
LINESTRING(1 1,2 2)            1.4142135623731
LINESTRING(10 10,20 20,30 40)  23 36.502815398729
```

4.33. ST_LINEFROMTEXT

ST_LINEFROMTEXT는 주어진 WKT를 라인 스트링 객체로 반환하는 함수이다.

ST_LINEFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_linefromtext



- 구성요소

구성요소	설명
str	GEOMETRY 라인 스트링 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_LINEFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_LINEFROMTEXT('LINESTRING(1 1,2 2)') FROM DUAL;

ST_LINEFROMTEXT('LINESTRING(11,22)')
-----
00000000000000F03F000000000000F03F000000000000004000000000000000400200000000200...
```

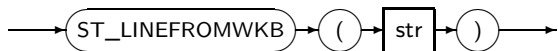
4.34. ST_LINEFROMWKB

ST_LINEFROMWKB는 주어진 WKB를 라인 스트링 객체로 반환하는 함수이다.

ST_LINEFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_linefromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 라인 스트링 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_LINEFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_LINEFROMWKB(ST_ASBINARY(GEOM)) F
      FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';

SST_ASTEXT(GEOM)                T_LINEFROMWKB(ST_ASBINARY(GEOM))

-----
LINESTRING(1 1,2 2)              00000000000000F03F000000000000F03F0000000000...
LINESTRING(10 10,20 20,30 40)    000000000000024400000000000002440000000000...
```

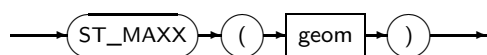
4.35. ST_MAXX

ST_MAXX는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 X 좌표값 중 큰 값을 반환하는 함수이다. EMPTY 공간 객체에 대해서는 NULL을 반환한다.

ST_MAXX의 세부 내용은 다음과 같다.

- 문법

st_maxx



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_MAXX 함수를 사용하는 예이다.

```
SQL> SELECT ID, ST_MAXX(GEOM) FROM GIS;
```

```

      ID ST_MAXX(GEOM)
-----
      101             1
      102             2
      103             2
      104             4
      105             2
      106            12
      107             5
      108             3
```

109	
110	3

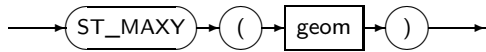
4.36. ST_MAXY

ST_MAXY는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 Y 좌표값 중 큰 값을 반환하는 함수이다. EMPTY 공간 객체에 대해서는 NULL을 반환한다.

ST_MAXY의 세부 내용은 다음과 같다.

- 문법

st_maxy



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_MAXY 함수를 사용하는 예이다.

```
SQL> SELECT ID, ST_MAXY(GEOM) FROM GIS;
```

ID	ST_MAXY(GEOM)
101	1
102	2
103	2
104	5
105	2
106	12
107	5
108	3
109	
110	3

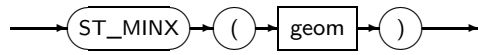
4.37. ST_MINX

ST_MINX는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 X 좌표값 중 작은 값을 반환하는 함수이다. EMPTY 공간 객체에 대해서는 NULL을 반환한다.

ST_MINX 의 세부 내용은 다음과 같다.

- 문법

st_minx



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_MINX 함수를 사용하는 예이다.

```
SQL> SELECT ID, ST_MINX(GEOM) FROM GIS;
```

ID	ST_MINX(GEOM)
101	1
102	1
103	1
104	1
105	1
106	0
107	1
108	1
109	
110	1

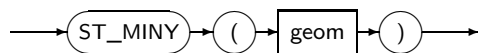
4.38. ST_MINY

ST_MINY는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 Y 좌표값 중 작은 값을 반환하는 함수이다. EMPTY 공간 객체에 대해서는 NULL을 반환한다.

ST_MINY 의 세부 내용은 다음과 같다.

- 문법

st_miny



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_MINY 함수를 사용하는 예이다.

```
SQL> SELECT ID, ST_MINY(GEOM) FROM GIS;
```

ID	ST_MINY(GEOM)
101	1
102	1
103	1
104	1
105	1
106	0
107	1
108	1
109	
110	1

4.39. ST_MLINEFROMTEXT

ST_MLINEFROMTEXT는 주어진 WKT를 멀티 라인 스트링 객체로 반환하는 함수이다.

ST_MLINEFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mlinefromtext



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 라인 스트링 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MLINEFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_MLINEFROMTEXT( 'MULTILINESTRING((1 1,2 2),(3 3,4 5))' ) FROM DUAL;

ST_MLINEFROMTEXT( 'MULTILINESTRING((11,22),(33,45))' )
-----
000000000000F03F000000000000F03F000000000000104000000000000144005000000200...
```

4.40. ST_MLINEFROMWKB

ST_MLINEFROMWKB는 주어진 WKB를 멀티 라인 스트링 객체로 반환하는 함수이다.

ST_MLINEFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mlinefromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 라인 스트링 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MLINEFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_MLINEFROMWKB(ST_ASBINARY(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTILINESTRING';

ST_MLINEFROMWKB(ST_ASBINARY(GEOM))
-----
000000000000F03F000000000000F03F0000000000001040000000000001440050000002...
00000000000024400000000000002440000000000003E400000000000003440050000002...
```

4.41. ST_MPOINTFROMTEXT

ST_MPOINTFROMTEXT는 주어진 WKT를 멀티 포인트 객체로 반환하는 함수이다.

ST_MPOINTFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mpointfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 포인트 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MPOINTFROMTEXT 함수를 사용하는 예이다.

```

SQL> SELECT ST_MPOINTFROMTEXT('MULTIPOINT(10 10,20 20)') FROM DUAL;

ST_MPOINTFROMTEXT('MULTIPOINT(1010,2020)')
-----
00000000000002440000000000000024400000000000003440000000000000344004000000002...
```

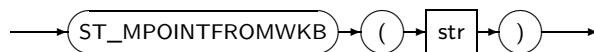
4.42. ST_MPOINTFROMWKB

ST_MPOINTFROMWKB는 주어진 WKB를 멀티 포인트 객체로 반환하는 함수이다.

ST_MPOINTFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mpointfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 포인트 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MPOINTFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_MPOINTFROMWKB(ST_ASBINAR(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTIPOINT';

ST_MPOINTFROMWKB(ST_ASBINAR(GEOM))
-----
000000000000F03F000000000000F03F0000000000000400000000000000400400000002...
000000000000244000000000000024400000000000034400000000000034400400000002...
```

4.43. ST_MPOLYFROMTEXT

ST_MPOLYFROMTEXT는 주어진 WKT를 멀티 폴리곤 객체로 반환하는 함수이다.

ST_MPOLYFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mpolyfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 폴리곤 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MPOLYFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_MPOLYFROMTEXT('MULTIPOLYGON(((10 10,10 20,20 20,20 15,10 10)),
      ((60 60,70 70,80 60,60 60)))') FROM DUAL;

ST_MPOLYFROMTEXT('MULTIPOLYGON(((1010,1020,2020,2015,1010)),((6060,7070,8060,
-----
000000000000244000000000000024400000000000054400000000008051400600000002...
```

4.44. ST_MPOLYFROMWKB

ST_MPOLYFROMWKB는 주어진 WKB를 멀티 폴리곤 객체로 반환하는 함수이다.

ST_MPOLYFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mpolyfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 멀티 폴리곤 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_MPOLYFROMWKB 함수를 사용하는 예이다.

```

SQL> SELECT ST_MPOLYFROMWKB(ST_ASBINARY(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTIPOLYGON';

ST_MPOLYFROMWKB(ST_ASBINARY(GEOM))
-----
0000000000000F03F000000000000F03F00000000000014400000000000001440060000000200...
0000000000000244000000000000024400000000000054400000000000805140060000000200...

```

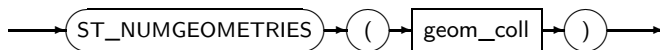
4.45. ST_NUMGEOMETRIES

ST_NUMGEOMETRIES는 GEOMETRY 컬렉션, 멀티 포인트, 멀티 라인 스트링, 멀티 폴리곤 객체에서 GEOMETRY 객체 수를 반환하는 함수이다.

ST_NUMGEOMETRIES의 세부 내용은 다음과 같다.

- 문법

st_numgeometries



- 구성요소

구성요소	설명
geom_coll	컬렉션 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_NUMGEOMETRIES 함수를 사용하는 예이다.

```
SQL> SELECT ST_NUMGEOMETRIES(GEOM),ST_ASTEXT(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'GEOMETRYCOLLECTION';

ST_NUMGEOMETRIES(GEOM) ST_ASTEXT(GEOM)
-----
2 GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
3 GEOMETRYCOLLECTION(POINT(10 10),POINT(30 30),LINESTRING(15 15,20 20))
```

4.46. ST_NUMINTERIORRING

ST_NUMINTERIORRING은 폴리곤 내부 링의 수를 반환하는 함수이다.

ST_NUMINTERIORRING의 세부 내용은 다음과 같다.

- 문법

st_numinteriorring



- 구성요소

구성요소	설명
poly	폴리곤 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_NUMINTERIORRING 함수를 사용하는 예이다.

```
SQL> SELECT ST_NUMINTERIORRING(GEOM), ST_ASTEXT(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON';

NUMINTERIORRING ST_ASTEXT(GEOM)
-----
0 POLYGON((1 1,2 1,2 2,1 2,1 1))
2 POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,6 11,9 11,9 10, ...
0 POLYGON((10 10,10 20,20 20,20 15,10 10))
2 POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11,7 12,10 12,10 11, ...
```

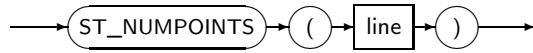
4.47. ST_NUMPOINTS

ST_NUMPOINTS는 라인 스트링의 포인트 수를 반환하는 함수이다.

ST_NUMPOINTS의 세부 내용은 다음과 같다.

- 문법

st_numpoints



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_NUMPOINTS 함수를 사용하는 예이다.

```
SQL> SELECT ST_NUMPOINTS(GEOM), ST_ASTEXT(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';
```

```
ST_NUMPOINTS(GEOM) ST_ASTEXT(GEOM)
```

```
-----
```

```
2                LINESTRING(1 1,2 2)
```

```
3                LINESTRING(10 10,20 20,30 40)
```

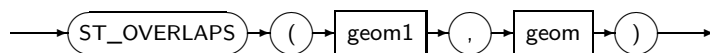
4.48. ST_OVERLAPS

ST_OVERLAPS는 주어진 두 GEOMETRY 객체가 서로 겹치는 영역이 존재하며, 완전히 포함되지는 않을 경우 1을 반환하는 함수이다.

ST_OVERLAPS의 세부 내용은 다음과 같다.

- 문법

st_overlaps



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_OVERLAPS 함수를 사용하는 예이다.

```
SQL> SELECT ST_OVERLAPS(A.GEOM,B.GEOM), ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
      FROM GIS A,GIS B
      WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

ST_OVERLAPS	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
0	POINT(1 1)	MULTIPOINT(1 1,2 2)
0	POINT(1 1)	LINESTRING(1 1,2 2)
0	POINT(1 1)	EMPTY
0	POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
0	POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11, ...
0	MULTIPOINT(1 1,2 2)	POINT(1 1)

4.49. ST_POINTFROMTEXT

ST_POINTFROMTEXT는 주어진 WKT를 포인트 객체로 반환하는 함수이다.

ST_POINTFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_pointfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 포인트 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_POINTFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_POINTFROMTEXT('POINT(10 10)') FROM DUAL;
```

```
ST_POINTFROMTEXT('POINT(1010)')
```

```
-----
0000000000000244000000000000024400000000000002440000000000000244001000000000...
```

4.50. ST_POINTFROMWKB

ST_POINTFROMWKB는 주어진 WKB를 포인트 객체로 반환하는 함수이다.

ST_POINTFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_pointfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 포인트 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_POINTFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_POINTFROMWKB(ST_ASBINARY(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POINT';
```

```
ST_POINTFROMWKB(ST_ASBINARY(GEOM))
```

```
-----
00000000000000F03F000000000000F03F000000000000F03F000000000000F03F010000000000...
000000000000024400000000000002440000000000002440000000000002440010000000000...
```

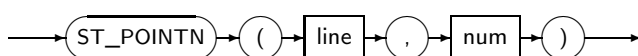
4.51. ST_POINTN

ST_POINTN은 라인 스트링 객체의 N번째 포인트를 반환하는 함수이다.

ST_POINTN의 세부 내용은 다음과 같다.

- 문법

st_pointn



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.
num	N번째 포인트를 지정한다.

- 예제

다음은 ST_POINTN 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM), ST_ASTEXT(ST_POINTN(GEOM,1)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';
```

ST_ASTEXT(GEOM)	ST_ASTEXT(ST_POINTN(GEOM,1))
LINESTRING(1 1,2 2)	POINT(1 1)
LINESTRING(10 10,20 20,30 40)	POINT(10 10)

4.52. ST_POINTONSURFACE

ST_POINTONSURFACE는 해당 공간 객체 위에 놓이는 것이 보장되는 하나의 점을 반환한다.

ST_POINTONSURFACE의 세부 내용은 다음과 같다.

- 문법

st_pointonsurface



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_POINTONSURFACE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_POINTONSURFACE(GEOM)), ST_ASTEXT(GEOM)
      FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON';
```

ST_POINTONSURFACE(GEOM)	ST_ASTEXT(GEOM)
POINT(1.5 1.5)	POLYGON((1 1,2 1,2 2,1 2,1 1))
POINT(3 6)	POLYGON((0 0,0 12,12 12 0,0 0),(6 10,6 11, ...
POINT(15 15)	POLYGON((10 10,10 20,20 20 15,10 10))

POINT(4 7)

POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11,7 12, ...

4.53. ST_POLYFROMTEXT

ST_POLYFROMTEXT는 주어진 WKT를 폴리곤 객체로 반환하는 함수이다.

ST_POLYFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_polyfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 폴리곤 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_POLYFROMTEXT 함수를 사용하는 예이다.

```
SQL> SELECT ST_POLYFROMTEXT('POLYGON((1 1,2 1,2 2,1 2,1 1))') FROM DUAL;

ST_POLYFROMTEXT('POLYGON((11,21,22,12,11))')
-----
0000000000000F03F000000000000F03F00000000000004000000000000040030000000...
```

4.54. ST_POLYFROMWKB

ST_POLYFROMWKB는 주어진 WKB를 폴리곤 객체로 반환하는 함수이다.

ST_POLYFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_polyfromwkb



- 구성요소

구성요소	설명
str	GEOMETRY 폴리곤 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_POLYFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_POLYFROMWKB(ST_ASBINARY(GEOM)) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POLYGON';
```

```
ST_POLYFROMWKB(ST_ASBINARY(GEOM))
```

```
-----
00000000000000F03F000000000000F03F0000000000000400000000000000400300000000...
0000000000000000000000000000000000000000000000000000000000000000002840000000000000028400300000000...
0000000000000024400000000000002440000000000000344000000000000034400300000000...
00000000000000F03F000000000000F03F0000000000002A400000000000002A400300000000...
```

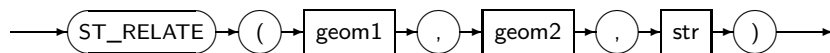
4.55. ST_RELATE

ST_RELATE는 GEOMETRY 객체 1과 GEOMETRY 객체 2가 주어진 관계를 만족시키면 1을 반환하는 함수이다. GEOMETRY 인자로 GEOMETRYCOLLECTION 타입이 올 경우에는 런타임 에러를 발생시킨다.

ST_RELATE의 세부 내용은 다음과 같다.

- 문법

st_relate



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
str	두 객체의 관계를 의미하는 문자열을 나타내는 CHAR, VARCAHR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_RELATE 함수를 사용하는 예이다.

```
SQL> SELECT ST_RELATE(A.GEOM,B.GEOM,'TTTTTTTTT'),
           ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
FROM GIS A,GIS B
WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

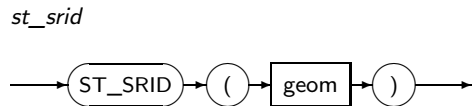
ST_RELATE	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
0	POINT(1 1)	MULTIPOINT(1 1,2 2)
0	POINT(1 1)	LINESTRING(1 1,2 2)
0	POINT(1 1)	EMPTY
0	POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
0	POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11, ...
0	MULTIPOINT(1 1,2 2)	POINT(1 1)

4.56. ST_SRID

ST_SRID는 주어진 GEOMETRY 객체의 SRID를 반환하는 함수이다.

ST_SRID의 세부 내용은 다음과 같다.

- 문법



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_SRID 함수를 사용하는 예이다.

```
SQL> SELECT ID,ST_SRID(GEOM) FROM GIS;
```

ST_SRID(GEOM)	ST_ASTEXT(B.GEOM)
101	MULTIPOINT(1 1,2 2)
101	LINESTRING(1 1,2 2)
101	EMPTY

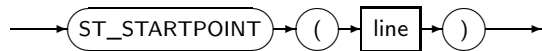
4.57. ST_STARTPOINT

ST_STARTPOINT는 라인 스트링의 첫 포인트를 반환하는 함수이다.

ST_STARTPOINT의 세부 내용은 다음과 같다.

- 문법

st_startpoint



- 구성요소

구성요소	설명
line	라인 스트링 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_STARTPOINT 함수를 사용하는 예이다.

```
SQL> SELECT ST_astext(geom),ST_ASTEXT(ST_STARTPOINT(GEOM))
      FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'LINESTRING';
```

ST_ASTEXT(GEOM)	ST_ASTEXT(ST_STARTPOINT(GEOM))
LINESTRING(1 1,2 2)	POINT(1 1)
LINESTRING(10 10,20 20,30 40)	POINT(10 10)

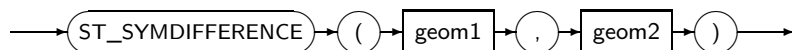
4.58. ST_SYMDIFFERENCE

ST_SYMDIFFERENCE는 주어진 두 GEOMETRY 객체의 INTERSECTION을 제외한 영역을 나타내는 GEOMETRY 객체를 반환하는 함수이다.

ST_SYMDIFFERENCE의 세부 내용은 다음과 같다.

- 문법

st_symdifference



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_SYMDIFFERENCE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM), ST_ASTEXT(B.GEOM),
           ST_ASTEXT(ST_SYMDIFFERENCE(A.GEOM, B.GEOM))
FROM GIS A, GIS B
WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
AND ST_DIMENSION(A.GEOM) = ST_DIMENSION(B.GEOM);
```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)	ST_SYMDIFFERENCE(A.GEOM, B.GEOM)
POINT(1 1)	POINT(1 1)	EMPTY
MULTIPOINT(1 1, 2 2)	POINT(1 1)	POINT(2 2)
POINT(10 10)	POINT(1 1)	MULTIPOINT(1 1, 10 10)
MULTILINESTRING(...)	LINESTRING(1 1, 2 2)	LINESTRING(3 3, 4 5)
LINESTRING(...)	LINESTRING(1 1, 2 2)	MULTILINESTRING((10 10, 20 20, ...
EMPTY	EMPTY	EMPTY

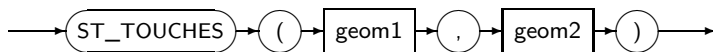
4.59. ST_TOUCHES

ST_TOUCHES는 주어진 두 GEOMETRY 객체가 한 점 이상을 공유하지만 INTERSECTION이 존재하지는 않을 때 1을 반환하는 함수이다. GEOMETRY 인자로 GEOMETRYCOLLECTION 타입이 올 경우에는 런타임 에러를 발생시킨다.

ST_TOUCHES의 세부 내용은 다음과 같다.

- 문법

st_touches



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_TOUCHES 함수를 사용하는 예이다.

```
SQL> SELECT ST_TOUCHES(A.GEOM,B.GEOM), ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
      FROM GIS A,GIS B
      VWHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
```

ST_TOUCHES	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
0	POINT(1 1)	MULTIPOINT(1 1,2 2)
1	POINT(1 1)	LINESTRING(1 1,2 2)
0	POINT(1 1)	EMPTY
0	POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
1	POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11, ...
0	MULTIPOINT(1 1,2 2)	POINT(1 1)

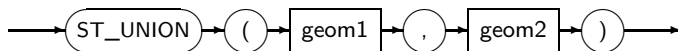
4.60. ST_UNION

ST_UNION은 주어진 두 GEOMETRY 객체의 영역을 합한 GEOMETRY 객체를 반환하는 함수이다.

ST_UNION의 세부 내용은 다음과 같다.

- 문법

st_union



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_UNION 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM),
      ST_ASTEXT(ST_UNION(A.GEOM,B.GEOM)) FROM GIS A,GIS B
      WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_DIMENSION(A.GEOM) = ST_DIMENSION(B.GEOM);
```

ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)	ST_UNION(A.GEOM,B.GEOM)
POINT(1 1)	MULTIPOINT(1 1,2 2)	MULTIPOINT(1 1,2 2)
POINT(1 1)	POINT(10 10)	MULTIPOINT(1 1,10 10)
MULTIPOINT(1 1,2 2)	POINT(10 10)	MULTIPOINT(1 1,2 2,10 10)
POINT(1 1)	MULTIPOINT(10 10,20 20)	MULTIPOINT(1 1,10 10,20 20)
LINESTRING(1 1,2 2)	LINESTRING(10 10,20 20, ...	MULTILINESTRING((1 1, ...

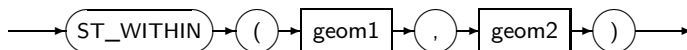
4.61. ST_WITHIN

ST_WITHIN은 GEOMETRY 객체 1이 GEOMETRY 객체 2 내부에 포함되면 1을 반환하는 함수이다.

ST_WITHIN의 세부 내용은 다음과 같다.

- 문법

st_within



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_WITHIN 함수를 사용하는 예이다.

```

SQL> SELECT ST_WITHIN(A.GEOM,B.GEOM), ST_ASTEXT(A.GEOM),ST_ASTEXT(B.GEOM)
      FROM GIS A,GIS B
      WHERE ST_GEOMETRYTYPE(A.GEOM) NOT LIKE 'GEOMETRYCOLLECTION'
      AND ST_GEOMETRYTYPE(B.GEOM) NOT LIKE 'GEOMETRYCOLLECTION';
  
```

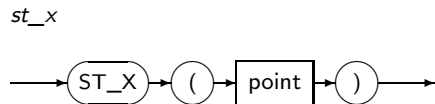
ST_WITHIN	ST_ASTEXT(A.GEOM)	ST_ASTEXT(B.GEOM)
1	POINT(1 1)	MULTIPOINT(1 1,2 2)
0	POINT(1 1)	LINESTRING(1 1,2 2)
0	POINT(1 1)	EMPTY
0	POINT(1 1)	POLYGON((10 10,10 20,20 20,20 15,10 10))
0	POINT(1 1)	POLYGON((1 1,1 13,13 13,13 1,1 1),(7 11, ...
0	MULTIPOINT(1 1,2 2)	POINT(1 1)

4.62. ST_X

ST_X는 포인트 객체의 X 좌표를 반환하는 함수이다.

ST_X의 세부 내용은 다음과 같다.

- 문법



- 구성요소

구성요소	설명
point	포인트 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_X 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM),ST_X(GEOM) FROM GIS
      WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POINT';
```

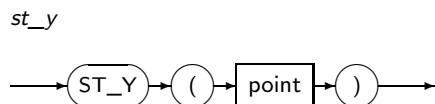
```
ST_ASTEXT(GEOM) ST_X(GEOM)
-----
POINT(1 1)      1
POINT(10 10)    10
```

4.63. ST_Y

ST_Y는 포인트 객체의 Y 좌표를 반환하는 함수이다.

ST_Y의 세부 내용은 다음과 같다.

- 문법



- 구성요소

구성요소	설명
point	포인트 객체를 나타내는 GEOMETRY TYPE이어야 한다.

- 예제

다음은 ST_Y 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM),ST_Y(GEOM) FROM GIS
       WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'POINT';
```

```
ST_ASTEXT(GEOM) ST_Y(GEOM)
-----
POINT(1 1)      1
POINT(10 10)    10
```


제5장 Spatial 유틸리티

본 장에서는 Tiberio Spatial에서 제공하는 유틸리티인 gisLoader의 사용법에 대해서 기술한다.

5.1. gisLoader

Tiberio Spatial에서는 shapefile 형식으로 되어 있는 공간 데이터를 Tiberio로 적재하는 데 사용할 수 있는 **gisLoader**라는 독립 실행형 유틸리티를 제공한다.

gisLoader 유틸리티는 shapefile 형태로 되어 있는 파일의 데이터를 읽어들이어서 tbLoader의 입력이 되는 데이터 파일을 생성한다. 즉, 사용자는 먼저 gisLoader 유틸리티를 사용하여 shapefile을 tbLoader의 입력 파일로 변환하게 되고, 그 다음 tbLoader를 사용해 최종적으로 공간 데이터를 Tiberio에 적재할 수가 있게 된다.

shapefile을 구성하는 .shp, .shx, .dbf 파일 중에서 .shx 파일은 gisLoader 유틸리티에서 해석하지 않는다. 즉, 이 파일에 해당되는 shapefile 정보는 모두 유실된다. gisLoader에서 해석이 가능한 shapefile의 shape type은 Null, Point, Polyline, Polygon, Multipoint로 한정된다. 이 외의 shape type이 shapefile 내에 존재할 경우 gisLoader 유틸리티는 수행을 멈추고 에러를 출력한다.

gisLoader가 생성하는 파일은 tbLoader의 입력 파일인 .ctl 파일과 .dat 파일이다(컨트롤 파일과 데이터 파일). 다만 테이블을 생성할 수 있는 스키마 스크립트 등은 생성해 주지 않는다. 그러므로 사용자는 생성된 컨트롤 파일의 내용을 조회하여 이후 tbLoader를 수행할 때 데이터가 정상적으로 적재될 수 있는지를 먼저 확인해야 한다. 예를 들어 기존의 테이블이 존재하지 않는 상태라면 데이터를 적재할 테이블을 미리 생성해 놓아야 한다.

다음은 gisLoader에서 생성한 .ctl 파일의 예제이다.

```
LOAD DATA
  INFILE 'test_table.dat'
  LOGFILE 'test_table.log'
  BADFILE 'test_table.bad'
  APPEND
  INTO TABLE test_table
  FIELDS TERMINATED BY ','
  OPTIONALLY ENCLOSED BY '"'
  ESCAPED BY '\\'
  LINES TERMINATED BY '\n'
  LOBFILE LOAD PARTIALLY DELIMITED BY ';'
  GEOMETRY LOAD AS BLOB
  (ID, FIPSSTCO, TRACT2000, BLOCK2000, STFID, geom OUTFILE)
```

적재할 테이블이 존재하지 않을 때 적재가 가능한 테이블은 다음의 create table 문을 통해 생성이 가능하다.

```
create table test_table (ID number, FIPSSTCO number, TRACT2000 number,
BLOCK2000 number, STFID number, geom GEOMETRY);
```

참고

tbLoader의 자세한 사용법은 "Tibero 유틸리티 안내서"를 참고한다.

사용법

다음은 gisLoader의 사용법에 대한 설명이다.

● 사용법

```
gisLoader <shapefile name> <table name> <idcol=newcolname> <endian=big/little>
<multipolygon=Y|N>
```

항목	설명
shapefile name	이미 존재하는 .shp 파일의 파일 이름을 명시한다.
table name	최종적으로 tbLoader를 통해 데이터를 적재하게 될 데이터베이스의 테이블 이름을 명시한다.
idcol	dbf 파일이 아닌 shp 파일의 record number를 이용하여 생성되는 컬럼의 이름을 명시한다. 생략하면 새로 컬럼을 생성하지 않는다.
endian	생성되는 Geometry type에서 사용될 endian 정보(big/little)를 명시한다. 생략하면 해당 머신 endian 정보를 사용한다.
multipolygon	.shp 파일의 Polygon type을 모두 multipolygon type으로 변환한다. Polygon type으로 변환하려면 N으로 설정한다.

● 예제

```
gisLoader test.shp test_table idcol=pk endian=little multipolygon=N
```

색인

A

ALL_GEOMETRY_COLUMNS 뷰, 4

G

GEOMETRY 객체 타입, 1

GEOMETRY_COLUMNS_BASE 테이블, 3

gisLoader, 59

R

REGISTER_GEOMETRY_COLUMNS 프러시저, 6

REGISTER_SPATIAL_REF_SYS 프러시저, 7

S

Spatial 뷰, 4

ALL_GEOMETRY_COLUMNS, 4

SPATIAL_REF_SYS, 5

Spatial 테이블, 3

GEOMETRY_COLUMNS_BASE, 3

SPATIAL_REF_SYS_BASE, 4

Spatial 프러시저, 6

REGISTER_GEOMETRY_COLUMNS 프러시저, 6

REGISTER_SPATIAL_REF_SYS, 7

UNREGISTER_GEOMETRY_COLUMNS, 7

UNREGISTER_SPATIAL_REF_SYS, 8

SPATIAL_REF_SYS 뷰, 5

SPATIAL_REF_SYS_BASE 테이블, 4

SQL 함수

ST_AREA, 11

ST_ASBINARY, 11

ST_ASTEXT, 12

ST_BOUNDARY, 13

ST_BUFFER, 14

ST_CENTROID, 14

ST_CONTAINS, 15

ST_CONVEXHULL, 16

ST_CROSSES, 17

ST_DIFFERENCE, 17

ST_DIMENSION, 18

ST_DISJOINT, 19

ST_DISTANCE, 20

ST_ENDPOINT, 21

ST_ENVELOPE, 21

ST_EQUALS, 22

ST_EXTERIORRING, 23

ST_GEOMCOLLFROMTEXT, 24

ST_GEOMCOLLFROMWKB, 24

ST_GEOMETRYN, 27

ST_GEOMETRYTYPE, 27

ST_GEOMFROMTEXT, 25

ST_GEOMFROMWKB, 26

ST_INTERIORRINGN, 28

ST_INTERSECTION, 29

ST_INTERSECTS, 30

ST_ISCLOSED, 31

ST_ISEMPTY, 31

ST_ISRING, 32

ST_ISSIMPLE, 32

ST_ISVALID, 33

ST_LENGTH, 34

ST_LINEFROMTEXT, 34

ST_LINEFROMWKB, 35

ST_MAXX, 36

ST_MAXY, 37

ST_MINX, 37

ST_MINY, 38

ST_MLINEFROMTEXT, 39

ST_MLINEFROMWKB, 40

ST_MPOINTFROMTEXT, 40

ST_MPOINTFROMWKB, 41

ST_MPOLYFROMTEXT, 42

ST_MPOLYFROMWKB, 42

ST_NUMGEOMETRIES, 43

ST_NUMINTERIORRING, 44

ST_NUMPOINTS, 44

ST_OVERLAPS, 45

ST_POINTFROMTEXT, 46

ST_POINTFROMWKB, 47

ST_POINTN, 47

ST_POINTONSURFACE, 48

ST_POLYFROMTEXT, 49
 ST_POLYFROMWKB, 49
 ST_RELATE, 50
 ST_SRID, 51
 ST_STARTPOINT, 52
 ST_SYMDIFFERENCE, 52
 ST_TOUCHES, 53
 ST_UNION, 54
 ST_WITHIN, 55
 ST_X, 56
 ST_Y, 56
 ST_AREA 함수, 11
 ST_ASBINARY 함수, 11
 ST_ASTEXT 함수, 12
 ST_BOUNDARY 함수, 13
 ST_BUFFER 함수, 14
 ST_CENTROID 함수, 14
 ST_CONTAINS 함수, 15
 ST_CONVEXHULL 함수, 16
 ST_CROSSES 함수, 17
 ST_DIFFERENCE 함수, 17
 ST_DIMENSION 함수, 18
 ST_DISJOINT 함수, 19
 ST_DISTANCE 함수, 20
 ST_ENDPOINT 함수, 21
 ST_ENVELOPE 함수, 21
 ST_EQUALS 함수, 22
 ST_EXTERIORRING 함수, 23
 ST_GEOMCOLLFROMTEXT 함수, 24
 ST_GEOMCOLLFROMWKB 함수, 24
 ST_GEOMETRYN 함수, 27
 ST_GEOMETRYTYPE 함수, 27
 ST_GEOMFROMTEXT 함수, 25
 ST_GEOMFROMWKB 함수, 26
 ST_INTERIORRINGN 함수, 28
 ST_INTERSECTION 함수, 29
 ST_INTERSECTS 함수, 30
 ST_ISCLOSED 함수, 31
 ST_ISEMPTY 함수, 31
 ST_ISRING 함수, 32
 ST_ISSIMPLE 함수, 32
 ST_ISVALID 함수, 33
 ST_LENGTH 함수, 34

ST_LINEFROMTEXT 함수, 34
 ST_LINEFROMWKB 함수, 35
 ST_MAXX 함수, 36
 ST_MAXY 함수, 37
 ST_MINX 함수, 37
 ST_MINY 함수, 38
 ST_MLINEFROMTEXT 함수, 39
 ST_MLINEFROMWKB 함수, 40
 ST_MPOINTFROMTEXT 함수, 40
 ST_MPOINTFROMWKB 함수, 41
 ST_MPOLYFROMTEXT 함수, 42
 ST_MPOLYFROMWKB 함수, 42
 ST_NUMGEOMETRIES 함수, 43
 ST_NUMINTERIORRING 함수, 44
 ST_NUMPOINTS 함수, 44
 ST_OVERLAPS 함수, 45
 ST_POINTFROMTEXT 함수, 46
 ST_POINTFROMWKB 함수, 47
 ST_POINTN 함수, 47
 ST_POINTONSURFACE 함수, 48
 ST_POLYFROMTEXT 함수, 49
 ST_POLYFROMWKB 함수, 49
 ST_RELATE 함수, 50
 ST_SRID 함수, 51
 ST_STARTPOINT 함수, 52
 ST_SYMDIFFERENCE 함수, 52
 ST_TOUCHES 함수, 53
 ST_UNION 함수, 54
 ST_WITHIN 함수, 55
 ST_X 함수, 56
 ST_Y 함수, 56
 SYSGIS.REGISTER_GEOMETRY_COLUMNS, 6
 SYSGIS.REGISTER_SPATIAL_REF_SYS, 7
 SYSGIS.UNREGISTER_GEOMETRY_COLUMNS, 7
 SYSGIS.UNREGISTER_SPATIAL_REF_SYS, 8

U

UNREGISTER_GEOMETRY_COLUMNS 프리시저, 7
 UNREGISTER_SPATIAL_REF_SYS 프리시저, 8
 USER_GEOMETRY_COLUMNS 뷰, 5

ㄱ

공간 인덱스, 9

