Tibero

Spatial 참조 안내서

Tibero 7

대한민국 경기도 성남시 분당구 황새울로258번길 29, BS 타워 9층 우)13595

Website

http://www.tmaxtibero.com

기술서비스센터

Tel : +82-1544-8629

E-Mail : info@tmax.co.kr

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: 2024-08-22

소프트웨어 버전: Tibero 7.2.2

안내서 버전: v7.2.2

내용 목차

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 gisLoader의 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
참고	참고 또는 주의사항
주의	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

시스템 사용 환경

	요구 사항
Platform	HP-UX 11i v3(11.31)
	Solaris (Solaris 11)
	AIX (AIX 7.1/AIX 7.2/AIX 7.3)
	GNU (X86, 64, IA64)
	Red Hat Enterprise Linux 7 kernel 3.10.0 이상
	Windows(x86) 64bit
Hardware	최소 2.5GB 하드디스크 공간
	1GB 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, 패키지와 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	Tibero에서 Geometry 타입에 대한 설명과 Spatial 기능 관련 프러시저 함수 목록 및 사용 방법 등을 기술한 안내서이다.

안내서	설명
Spatial 참조 안내서	
Tibero TEXT 참조 안내서	Tibero의 제공하는 Text Index를 소개하고, Text Index를 생성 하고 사용하는 방법을 기술하는 안내서이다.
Tibero TDP.NET 안내서	Tibero Data Provider for .NET 기능을 기술하는 안내서이다.
Tibero IMCS 안내서	Tibero에서 제공하는 In-Memory Column Store(이하 IMCS) 기능을 기술하는 안내서이다.

제1장 Spatial 소개

본 장에서는 Tibero Spatial의 기본적인 개념을 설명한다.

1.1. 개요

Tibero 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

Tibero Spatial에서는 SQL-MM 표준에 따른 7가지 타입의 GEOMETRY 객체를 지원한다.

다음은 Tibero Spatial에서 지원하는 GEOMETRY 타입이다. GEOMETRY 타입은 binary 형태로 저장되며, 구조는 WKB(Well Known Binary)의 형식을 따른다. WKB 형식에서 GEOMETRY 타입의 객체의 좌표 정보는 부동소숫점으로 저장된다. 따라서, 수치에 대한 연산의 결과는 약간의 오차를 포함하게 된다. 또한, 부동소숫점 연산 환경에 따라서 결과값의 차이가 발생할 수도 있다.

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.

1.3. Spatial Referencing System(SRS)

Tibero Spatial에서는 실제 공간을 묘사하기 위한 좌표를 실제 공간과 연결하여 계산하는 여러 좌표계를 지원한다. GEOMETRY가 어떤 좌표계에 속해있느냐에 따라 Spatial 함수들의 결과값이 달라질 수 있다.

- **직교좌표계(Cartesian Coordinate)**

직교좌표계는 공간 상의 위치를 2차원 평명 또는 3차원 공간 상에 나타낸 좌표계이다. 지구의 곡률이 드러나지 않는 국소적인 공간을 효율적으로 묘사할 때 유용하다. Spatial에서는 SRID가 지정되지 않은 GEOMETRY는 기본적으로 직교좌표계에서 연산한다.

- **구면좌표계(Spherical Coordinate)**

  구면좌표계 또는 Geography 좌표계는 공간상의 위치를 2차원 구면 또는 타원면에(spheroid) 나타낸 좌표계이다. 구면좌표계는 위도/경도를 통해 실제 위치에 접근하며, 전지구적인 공간을 정확하게 묘사할 때 유용하다. 일반적으로 구면체 혹은 회전타원체를 통해 실제 지구상의 위치를 표현한다. 구면좌표계에서 사용하는 구면체 혹은 회전 타원체의 반지름, 장반경, 단반경 정보는 GEOMETRY에 주어진 SRID에 따라서 달라지며, GEOMETRY마다 개별적으로 다른 회전 타원체를 저장할 수는 없다.

- **SRID(Spatial Reference Identifier)**

  Spatial은 좌표계의 Identifier 타입인 SRID를 제공한다. SRID는 GEOMETRY가 해당하는 좌표계가 직교좌표계인지 구면좌표계인지를 포함하여 특정 좌표계에 대한 정보를 담고 있다.

제2장 Spatial 구조

본 장에서는 Tibero Spatial과 관련된 스키마 객체에 대하여 설명한다.

2.1. 테이블

다음은 관련 테이블 목록이다.

테이블	설명
SPATIAL_REF_SYS_BASE	데이터베이스에서 사용하는 Spatial Reference System의 메타데이터를 저장하는 테이블이다.
UNITS_OF_MEASURE	데이터베이스에서 사용하는 단위계의 정보를 저장하는 테이블이다.

2.1.1. SPATIAL_REF_SYS_BASE

데이터베이스에서 사용하는 Spatial Reference System의 메타데이터를 저장하는 테이블이다.

- 컬럼

컬럼	데이터 타입	설명
OWNER	VARCHAR(30)	Spatial Reference System를 추가한 DB 사용자이다.
SRID	INTEGER	Spatial Reference System의 데이터베이스 내에서의 ID이다.
AUTH_NAME	VARCHAR(256)	Spatial Reference System에서 사용하는 표준의 이름이다.
AUTH_SRID	NUMBER	표준에 의해 정의된 Spatial Reference System의 ID이다.
SRTEXT	VARCHAR(2000)	Spatial Reference System의 Well-Known Text 표현이다.
PROJ4TEXT	VARCHAR(2000)	좌표 변환을 위한 proj4 라이브러리의 좌표 정의 string 표현이다.

- 참조

  SPATIAL_REF_SYS

2.1.2. UNITS_OF_MEASURE

데이터베이스에서 사용하는 단위계의 정보를 저장하는 테이블이다.

- 컬럼

컬럼	데이터 타입	설명
SHORT_NAME	VARCHAR2(255)	단위계의 축약형 이름이다.
UOM_CODE	NUMBER(10)	단위계의 ID이다.
UNIT_OF_MEAS_NAME	VAR CHAR2(2083)	단위계의 이름이다. URL이나 ULI 또한 될 수 있다.
UNIT_OF_MEAS_TYPE	VARCHAR2(50)	각, 넓이, 길이 등의 단위계의 유형이다.
TARGET_UOM_CODE	NUMBER	변환에 사용되는 단위계의 ID이다.
FACTOR_B	NUMBER	단위 변환을 위한 인자이다. TARGET_UOM_CODE에 명시된 단위계의 얼마가 해당 단위계와 동일한지를 나타낸다. 간단한 값으로 표현이 불가능할 경우, FACTOR_C와 함께 FACTOR_B/FACTOR_C로써 해당 값을 나타낸다.
FACTOR_C	NUMBER	단위 변환을 위한 인자이다.
REMARKS	VARCHAR2(255)	단위계의 비고이다.
INFORMA TION_SOURCE	VARCHAR2(255)	단위계 정보의 출처이다. 예를 들어 'ISO 1000'등이 있다.
DATA_SOURCE	VARCHAR2(40)	단위계 정보를 제공한 조직이다. 예를 들어 'EPSG'등이 있다.
REVISION_DATE	DATE	단위계 정보의 최신 개정일이다.
CHANGE_ID	VARCHAR(255)	단위계의 변경일자에 대한 정보이다.
DEPRECATED	NUMBER(1)	해당 단위계의 사용 가능 여부이다.

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	데이터베이스에 등록되어 있는 모든 GEOMETRY COLUMN의 메타데이터를 나열한다.
SPATIAL_REF_SYS	데이터베이스에서 사용하는 Spatial Reference System 정보를 나열한다.
USER_GEOMETRY_COLUMNS	데이터베이스에 등록되어 있는 GEOMETRY COLUMN의 메타데이터를 나열한다.

뷰	설명
AREA_UNITS	UNITS_OF_MEASURE에 등록되어 있는 너비 단위계들의 정보를 나열한다.
DIST_UNITS	UNITS_OF_MEASURE에 등록되어 있는 거리 단위계들의 정보를 나열한다.
ANGLE_UNITS	UNITS_OF_MEASURE에 등록되어 있는 각도 단위계들의 정보를 나열한다.

2.2.1. ALL_GEOMETRY_COLUMNS

ALL_GEOMETRY_COLUMNS는 데이터베이스의 모든 GEOMETRY COLUMN에 대한 메타 데이터를 나열하는 뷰이다. 본 뷰는 OGC 표준에 맞게 설정되었다.

- 컬럼

컬럼	데이터 타입	설명
F_TABLE_CATALOG	VARCHAR(64)	GEOMETRY 컬럼이 속한 데이터베이스의 이름이다.
F_TABLE_SCHEMA	VARCHAR(128)	GEOMETRY 컬럼이 속한 TABLE의 소유자이다.
F_TABLE_NAME	VARCHAR(128)	GEOMETRY 컬럼이 속한 TABLE의 이름이다.
F_GEOMETRY_COLUMN	VARCHAR(128)	GEOMETRY 컬럼의 이름이다.
STORAGE_TYPE	VARCHAR(0)	항상 NULL이다.
GEOMETRY_TYPE	NUMBER	GEOMETRY 타입의 정수 표기법이다.
TYPE	VARCHAR (65532)	GEOMETRY 타입을 나타내는 문자열이다.
COORD_DIMENSION	NUMBER	GEOMETRY의 DIMENSION이다.
MAX_PPR	VARCHAR(0)	항상 NULL이다.
SRID	NUMBER	GEOMETRY의 Spatial Reference System ID이다.

- 참조

  USER_GEOMETRY_COLUMNS

2.2.2. SPATIAL_REF_SYS

SPATIAL_REF_SYS는 데이터베이스에서 사용하는 Spatial Reference System 정보를 나열하는 뷰이다.

- 컬럼

  SPATIAL_REF_SYS_BASE 테이블에서 OWNER 컬럼을 제외한 컬럼들로 구성된다.

- 참조

  SPATIAL_REF_SYS_BASE

2.2.3. USER_GEOMETRY_COLUMNS

USER_GEOMETRY_COLUMNS은 현재 사용자가 소유한 모든 테이블의 GEOMETRY COLUMN을 나열하는 뷰이다.

- 컬럼

  ALL_GEOMETRY_COLUMNS과 같다.

- 참조

  ALL_GEOMETRY_COLUMNS

2.2.4. AREA_UNITS

AREA_UNITS은 너비 단위계의 기본 정보를 나열하는 뷰이다.

- 컬럼

컬럼	데이터 타입	설명
SHORT_NAME	VARCHAR2(255)	너비 단위계의 축약형 이름이다.
UNIT_OF_MEAS_NAME	VAR CHAR2(2083)	너비 단위계의 이름이다. URL이나 ULI 또한 될 수 있다.
CONVERSION_FACTOR	NUMBER	다른 너비 단위계로 변환하기 위한 인자이다. UNITS_OF_MEASURE 테이블의 FACTOR_B/FACTOR_C 값이다.

- 참조

  UNITS_OF_MEASURE

2.2.5. DIST_UNITS

DIST_UNITS은 길이 단위계의 기본 정보를 나열하는 뷰이다.

- 컬럼

컬럼	데이터 타입	설명
SHORT_NAME	VARCHAR2(255)	길이 단위계의 축약형 이름이다.
UNIT_OF_MEAS_NAME	VAR CHAR2(2083)	길이 단위계의 이름이다. URL이나 ULI 또한 될 수 있다.
CONVERSION_FACTOR	NUMBER	다른 길이 단위계로 변환하기 위한 인자이다. UNITS_OF_MEASURE 테이블의 FACTOR_B/FACTOR_C 값이다.

- 참조

  UNITS_OF_MEASURE

2.2.6. ANGLE_UNITS

ANGLE_UNITS은 각도 단위계의 기본 정보를 나열하는 뷰이다.

- 컬럼

컬럼	데이터 타입	설명
SHORT_NAME	VARCHAR2(255)	각도 단위계의 축약형 이름이다.
UNIT_OF_MEAS_NAME	VAR CHAR2(2083)	각도 단위계의 이름이다. URL이나 ULI 또한 될 수 있다.
CONVERSION_FACTOR	NUMBER	다른 각도 단위계로 변환하기 위한 인자이다. UNITS_OF_MEASURE 테이블의 FACTOR_B/FACTOR_C 값이다.

- 참조

  UNITS_OF_MEASURE

2.3. 프러시저

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

다음은 Spatial에서 제공하는 프러시저 목록이다.

프러시저	설명
REGISTER_SRS	SPATIAL_REF_SYS_BASE 테이블에 Spatial Reference System 메타데이터를 등록한다.

프러시저	설명
UNREGISTER_SRS	SPATIAL_REF_SYS_BASE 테이블에 등록한 Spatial Reference System 메타데이터를 삭제한다.

2.3.1. REGISTER_SRS

SPATIAL_REF_SYS_BASE 테이블에 Spatial Reference System 메타데이터를 등록하는 프러시저이다.

REGISTER_SRS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
EXEC REGISTER_SRS
              (
         SRID      IN INTEGER,
         AUTH_NAME IN VARCHAR2,
         AUTH_SRID IN INTEGER,
         SRTEXT    IN VARCHAR2,
         PROJ4TEXT IN VARCHAR2
);
```

- 파라미터

파라미터	설명
SRID	Spatial Reference System의 데이터베이스 내에서의 ID이다.
AUTH_NAME	Spatial Reference System에서 사용하는 표준의 이름이다.
AUTH_SRID	표준에 의해 정의된 Spatial Reference System의 ID이다.
SRTEXT	Spatial Reference System의 Well-Known Text 표현이다.
PROJ4TEXT	좌표 변환을 위한 proj4 라이브러리의 좌표 정의 string 표현이다.

2.3.2. UNREGISTER_SRS

SPATIAL_REF_SYS_BASE 테이블에 등록한 Spatial Reference System 메타데이터를 삭제하는 프러시저이다.

UNREGISTER_SRS 프러시저의 세부 내용은 다음과 같다.

- 프로토타입

```
EXEC UNREGISTER_SRS
(
         AUTH_NAME IN VARCHAR2
```

```
        AUTH_SRID IN INTEGER
);
```

- 파라미터

파라미터	설명
AUTH_NAME	삭제할 Spatial Reference System에서 사용하는 표준의 이름이다.
AUTH_SRID	삭제할 Spatial Reference System에서 사용하는 표준의 ID이다.

제3장 공간 인덱스

Tibero Spatial에서는 공간 질의 성능 향상을 위해 RTREE 방식으로 구현된 공간 인덱스를 제공한다. 좌표계에 따라 평면 좌표계, 회전타원체 좌표계에 대한 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
```

항목	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_CONTAINS
  - ST_COVEREDBY
  - ST_COVERS
  - ST_CROSSES
  - ST_DWITHIN
  - ST_EQUALS
  - ST_INTERSECTS
  - ST_OVERLAPS
  - ST_TOUCHES
  - ST_WITHIN
- 회전타원체 좌표계의 공간 인덱스는 해당 테이블의 GEOMETRY 컬럼에 제약 조건을 추가하여 생성 가능하다.
  - 사용법

    ```
    ALTER TABLE table_name ADD CONSTRAINTS
                    constraint_name CHECK(ST_SRID(col_name)=srid);
    ```

  - 예제

    ```
    SQL>DROP INDEX RT_IDX_GIS;
    Index 'RT_IDX_GIS' dropped.
    SQL>UPDATE GIS SET GEOM=ST_SETSRID(GEOM,4326);
    ```

```
10 rows updated.
SQL>ALTER TABLE GIS ADD CONSTRAINTS SRID4326 CHECK(ST_SRID(GEOM)=4326);
Table 'GIS' altered.
SQL>CREATE INDEX RT_IDX_4326 ON GIS(GEOM) RTREE;
Index 'RT_IDX_4326' created.
COMMIT;
```

- 기존의 GEOMETRY 컬럼의 좌표계를 변경하여 공간 인덱스를 생성해야할 경우, 기존의 제약조건이 있다면 이를 삭제하고 GEOMETRY들의 SRID를 변경한 후, 해당 SRID에 대한 제약조건을 추가하여 공간 인덱스를 생성한다.
  - 사용법

```
ALTER TABLE table_name DROP CONSTRAINTS constraint_name;
```

  - 예제

```
SQL>DROP INDEX RT_IDX_4326;
Index 'RT_IDX_4326' dropped.
SQL>ALTER TABLE GIS DROP CONSTRAINTS SRID4326;
Table 'GIS' altered.
SQL>UPDATE GIS SET GEOM=ST_SETSRID(GEOM,0);
10 rows updated.
SQL>ALTER TABLE GIS ADD CONSTRAINTS SRID4326 CHECK(ST_SRID(GEOM)=0);
Table 'GIS' altered.
SQL>CREATE INDEX RT_IDX_GIS ON GIS(GEOM) RTREE;
Index 'RT_IDX_GIS' created.
COMMIT;
```

3.2. 공간 인덱스 제거

공간 인덱스의 제거는 다른 인덱스와 동일하다.

- 사용법

```
DROP INDEX index_name
```

항목	설명
index_name	삭제할 공간 인덱스의 이름이다.

제4장 Spatial 함수

본 장에서는 Tibero Spatial에서 제공되는 함수에 대해 설명한다.

4.1. 개요

Spatial 함수는 기본적으로 제공되는 함수들이 있고, 특정 환경(c++11 이상, linux, x86, 64bit)에서만 제공되는 함수들이 있다. 또한 기본적으로 제공되지만, 위 환경에서 SRID를 인자로 줘서 좌표계를 고려한 연산을 제공하는 함수들도 있다.

본 안내서에서는 특정 환경에서만 제공되는 함수들은 함수명 뒤에 '#' 기호를 기본적으로 제공되지만 특정 환경에서 좌표계를 고려한 연산을 추가 제공하는 함수들은 함수명 뒤에 '$' 기호를 붙여서 표기한다.

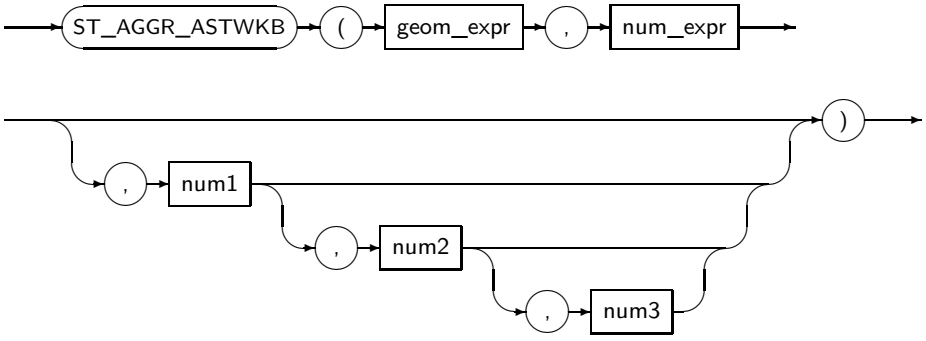
4.2. ST_AGGR_ASTWKB(#)

ST_AGGR_ASTWKB는 GEOMETRY 객체를 TWKB(Tiny Well Known Binary) 형태로 반환하는 집합 함수이다. GEOMETRY 객체마다 식별자를 포함한 집합의 TWKB를 반환한다.

ST_AGGR_ASTWKB의 세부 내용은 다음과 같다.

- 문법

st_aggrastwkb



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 임의의 연산식이다.
num_expr	geom_expr 객체 각각에 해당하는 식별자이다.

구성요소	설명
num1	TWKB로 표현될 좌표의 소수점 이하 유효숫자의 갯수이다. 기본값으로 0이 설정되어 있다.
num2	반환될 TWKB에 인코딩된 GEOMETRY 객체의 길이 요소를 포함할지를 결정하는 인자이다. 1을 넣을 경우 반환되는 TWKB가 길이를 포함하게된다. 기본 설정은 길이를 포함하지 않는다.
num3	반환될 TWKB에 GEOMETRY 객체의 경계상자를 포함할지를 결정하는 인자이다. 1을 넣을 경우 반환되는 TWKB가 경계 상자를 포함하게된다. 기본 설정은 경계 상자를 포함하지 않는다.

- 예제

  다음은 ST_AGGR_ASTWKB 함수를 사용하는 예이다.

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

ST_AGGR_ASTWKB(GEOM,ID)
--------------------------------------------------------------------------------
07040ACA01CC01CE01D001D201D401D601D801DA01DC01010002020400020202020200020202020
02050002020202020202020202040300010502020200000201000001030003050000001818000017
1700050C14000206000001050005000D000606000005050006000201050202020000020100000101
05040400040400000303000700020100020202000204040202071007000201000202020002040402
02
```

```
SQL> SELECT ST_AGGR_ASTWKB(GEOM, ID, 3, 1, 1) FROM GIS;

ST_AGGR_ASTWKB(GEOM,ID,3,1,1)
--------------------------------------------------------------------------------
6707DB0200C0BB0100C0BB010ACA01CC01CE01D001D201D401D601D801DA01DC0161030AD00F00D0
0F00D00FD00F640311D00FD00FD00FD00F02D00FD00FD00FD00F620311D00FD00FD00FD00F02D00F
D00FD00FD00F65031BD00FF02ED00FC03E0202D00FD00FD00FD00F02D00FD00FD00FA01F63031AD0
0FD00FD00FD00F0105D00FD00FD00F0000D00FCF0F0000CF0F63033E00C0BB0100C0BB0103050000
00C0BB01C0BB010000BFBB01BFBB010005E05DA09C0100D00FF02E0000CF0FEF2E000500AF6D00F0
2EF02E0000EF2EEF2E0066032DD00FC03ED00FC03E020105D00FD00FD00F0000D00FCF0F0000CF0F
0105A01FA01F00A01FA01F00009F1F9F1F0067032AD00FA01FD00FA01F0261030AD00F00D00F00D0
0FD00F620311A01FD00FA01FD00F02A01FA01FD00FD00F67120067032AD00FA01FD00FA01F026103
0AD00F00D00F00D00FD00F620311A01FD00FA01FD00F02A01FA01FD00FD00F
```

4.3. ST_AREA($)

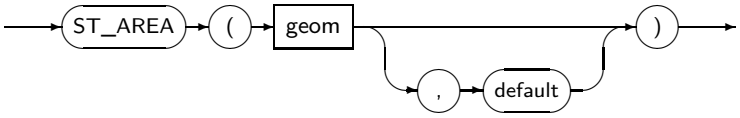
ST_AREA은 GEOMETRY 객체의 면적을 반환하는 함수이다. POLYGON 객체에 대해 면적을 계산하여 반환한다. POINT, LINESTRING에 대해서는 0을 반환한다. MULTI 타입의 GEOMETRY 객체에 대해서는 내부의 GEOMETRY들의 계산 결과의 합을 값을 반환한다. 좌표계 정보가 회전체인 경우, 평방미터 단위

로 해당 객체의 곡면의 면적을 구한다. 회전타원체에 대한 연산이 기본 설정이며, 더 빠른 계산을 원하면 구면좌표계에 대한 연산으로 바꿀 수 있다.

ST_AREA의 세부 내용은 다음과 같다.

- 문법

st_area



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
default	연산 방식을 설정한다. 좌표계 정보가 회전체 좌표계인 경우 기본으로 회전타원체에 대한 연산으로 설정되어있고, 0을 인자로 주면 구면좌표계에 대해 연산한다.

- 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)))
```

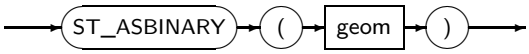
4.4. ST_ASBINARY

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 타입이어야 한다.

- 예제

다음은 ST_ASBINARY 함수를 사용하는 예이다.

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

ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
ST_ASBINARY(GEOM)
--------------------------------------------------------------------------------
POINT(1 1)
0101000000000000000000F03F000000000000F03F
```

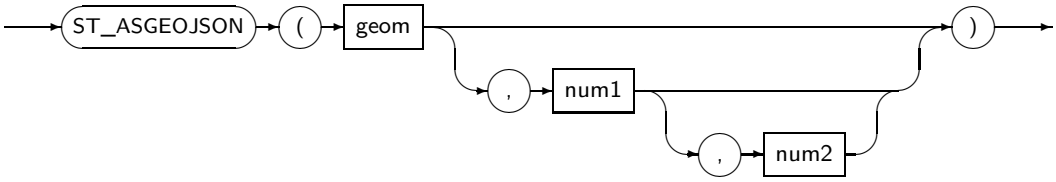
4.5. ST_ASGEOJSON

ST_ASGEOJSON는 GEOMETRY 객체를 GEOJSON 형태로 반환하는 함수이다.

ST_ASGEOJSON의 세부 내용은 다음과 같다.

- 문법

st_asgeojson



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	좌표의 유효숫자의 갯수를 입력한다. 기본값으로 9가 설정되어 있다.
num2	반환될 GEOJASON 형식의 옵션을 설정한다. 기본값으로 8이 설정되어 있다.

구성요소	설명
	– 0 : 아무 옵션도 추가하지 않는다. – 1 : 해당 GEOMETRY의 최소 경계 사각형을 추가한다. – 2 : Short CRS 형식으로 좌표를 출력한다.(예 EPSG:4326) – 4 : Long CRS 형식으로 좌표를 출력한다.(예: urn:ogc:def:crs:EPSG::4326) – 8 : SRID가 4326이 아닌 경우 Short CRS 형식으로 좌표를 출력한다. (기본값)

- 예제

다음은 ST_ASGEOJSON 함수를 사용하는 예이다.

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

ST_ASGEOJSON(GEOM)
--------------------------------------------------------------------------------
{"type":"Point","coordinates":[1,1]}
{"type":"MultiPoint","coordinates":[[1,1],[2,2]]}
{"type":"LineString","coordinates":[[1,1],[2,2]]}
{"type":"MultiLineString","coordinates":[[[1,1],[2,2]],[[3,3],[4,5]]]}
{"type":"Polygon","coordinates":[[[1,1],[2,1],[2,2],[1,2],[1,1]]]}
{"type":"Polygon","coordinates":[[[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]]]}
{"type":"MultiPolygon","coordinates":[[[[1,1],[2,1],[2,2],[1,2],[1,1]]],[[[3,3],
[3,5],[5,5],[5,3],[3,3]]]]}
{"type":"GeometryCollection","geometries":[{"type":"Point","coordinates":[1,1]},
{"type":"LineString","coordinates":[[2,2],[3,3]]}]}
{"type":"GeometryCollection","geometries":[]}
{"type":"GeometryCollection","geometries":[{"type":"Point","coordinates":[1,1]},
{"type":"LineString","coordinates":[[2,2],[3,3]]}]}
```

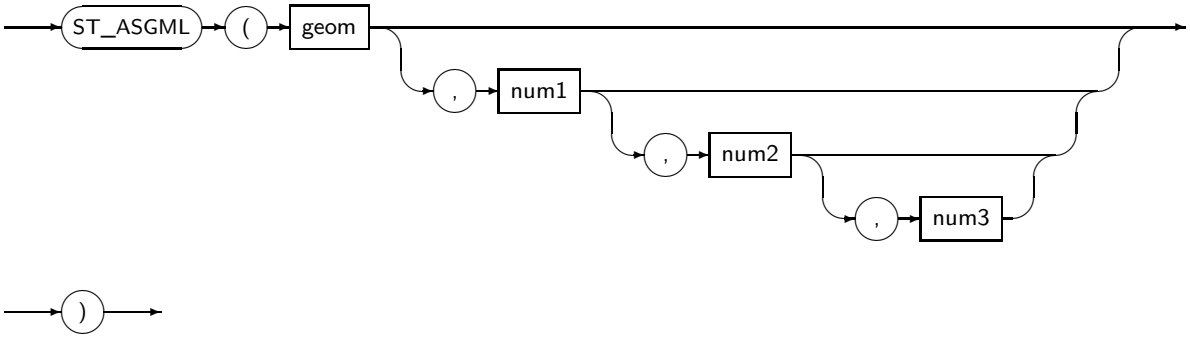
4.6. ST_ASGML

ST_ASGML는 GEOMETRY 객체를 GML 형태로 반환하는 함수이다.

ST_ASGML의 세부 내용은 다음과 같다.

- 문법

st_asgml



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	좌표의 유효숫자의 갯수를 입력한다. 기본값으로 15가 설정되어 있다.
num2	반환될 GML 형식의 옵션을 설정한다. 기본값으로 0(Short CRS)이 설정되어 있다. – 0 : Short CRS 형식으로 좌표를 출력한다. (기본값) – 1 : Long CRS 형식으로 좌표를 출력한다.

- 예제

  다음은 ST_ASGML 함수를 사용하는 예이다.

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

ST_ASGML(GEOM)
--------------------------------------------------------------------------------
<gml:Point><coordinates>1,1</coordinates></gml:Point>
<gml:MultiPoint><pointMember><gml:Point><coordinates>1,1</coordinates></gml:Poin
t></pointMember><pointMember><gml:Point><coordinates>2,2</coordinates></gml:Poin
t></pointMember></gml:MultiPoint>
<gml:LineString><coordinates>1,1 2,2</coordinates></gml:LineString>
<gml:MultiLineString><lineStringMember><gml:LineString><coordinates>1,1 2,2</coo
rdinates></gml:LineString></lineStringMember><lineStringMember><gml:LineString><
coordinates>3,3 4,5</coordinates></gml:LineString></lineStringMember></gml:Multi
LineString>
<gml:Polygon><outerBoundaryIs><LinearRing><coordinates>1,1 2,1 2,2 1,2 1,1</coor
dinates></LinearRing></outerBoundaryIs></gml:Polygon>
<gml:Polygon><outerBoundaryIs><LinearRing><coordinates>0,0 0,12 12,12 12,0 0,0</
coordinates></LinearRing></outerBoundaryIs><innerBoundaryIs><LinearRing><coordin
ates>6,10 6,11 9,11 9,10 6,10</coordinates></LinearRing></innerBoundaryIs><inner
BoundaryIs><LinearRing><coordinates>6,3 6,6 9,6 9,3 6,3</coordinates></LinearRin
```

```
g></innerBoundaryIs></gml:Polygon>
<gml:MultiPolygon><polygonMember><gml:Polygon><outerBoundaryIs><LinearRing><coor
dinates>1,1 2,1 2,2 1,2 1,1</coordinates></LinearRing></outerBoundaryIs></gml:Po
lygon></polygonMember><polygonMember><gml:Polygon><outerBoundaryIs><LinearRing><
coordinates>3,3 3,5 5,5 5,3 3,3</coordinates></LinearRing></outerBoundaryIs></gm
l:Polygon></polygonMember></gml:MultiPolygon>
<gml:MultiGeometry><geometryMember><gml:Point><coordinates>1,1</coordinates></gm
l:Point></geometryMember><geometryMember><gml:LineString><coordinates>2,2 3,3</c
oordinates></gml:LineString></geometryMember></gml:MultiGeometry>

<gml:MultiGeometry><geometryMember><gml:Point><coordinates>1,1</coordinates></gm
l:Point></geometryMember><geometryMember><gml:LineString><coordinates>2,2 3,3</c
oordinates></gml:LineString></geometryMember></gml:MultiGeometry>
```

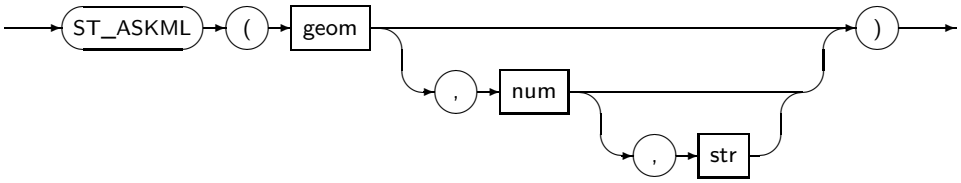
4.7. ST_ASKML

ST_ASKML는 GEOMETRY 객체를 KML 형태로 반환하는 함수이다.

ST_ASKML의 세부 내용은 다음과 같다.

- 문법

st_askml



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다. GEOMETRYCOLLECTION 타입의 객체가 올 경우 에러를 반환한다.
num	좌표의 유효숫자의 갯수를 입력한다. 기본값으로 15가 설정되어 있다.
str	반환될 KML의 네임스페이스 접두사를 설정한다. 기본 네임스페이스는 접두사를 쓰지 않는다.

- 예제

다음은 ST_ASKML 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASKML(ST_SETSRID(GEOM,4326)) FROM GIS WHERE ST_GEOMETRYTYPE(GEOM)
     NOT LIKE 'GEOMETRYCOLLECTION';
```

```
ST_ASKML(GEOM)
--------------------------------------------------------------------------------
<Point><coordinates>1,1</coordinates></Point>
<MultiGeometry><Point><coordinates>1,1</coordinates></Point><Point><coordinates>
2,2</coordinates></Point></MultiGeometry>
<LineString><coordinates>1,1 2,2</coordinates></LineString>
<MultiGeometry><LineString><coordinates>1,1 2,2</coordinates></LineString><LineS
tring><coordinates>3,3 4,5</coordinates></LineString></MultiGeometry>
<Polygon><outerBoundaryIs><LinearRing><coordinates>1,1 2,1 2,2 1,2 1,1</coordina
tes></LinearRing></outerBoundaryIs></Polygon>
<Polygon><outerBoundaryIs><LinearRing><coordinates>0,0 0,12 12,12 12,0 0,0</coor
dinates></LinearRing></outerBoundaryIs><innerBoundaryIs><LinearRing><coordinates
>6,10 6,11 9,11 9,10 6,10</coordinates></LinearRing></innerBoundaryIs><innerBoun
daryIs><LinearRing><coordinates>6,3 6,6 9,6 9,3 6,3</coordinates></LinearRing></
innerBoundaryIs></Polygon>
<MultiGeometry><Polygon><outerBoundaryIs><LinearRing><coordinates>1,1 2,1 2,2 1,
2 1,1</coordinates></LinearRing></outerBoundaryIs></Polygon><Polygon><outerBound
aryIs><LinearRing><coordinates>3,3 3,5 5,5 5,3 3,3</coordinates></LinearRing></o
uterBoundaryIs></Polygon></MultiGeometry>
```

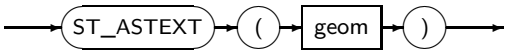
4.8. ST_ASTEXT

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 타입이어야 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))
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))
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))
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

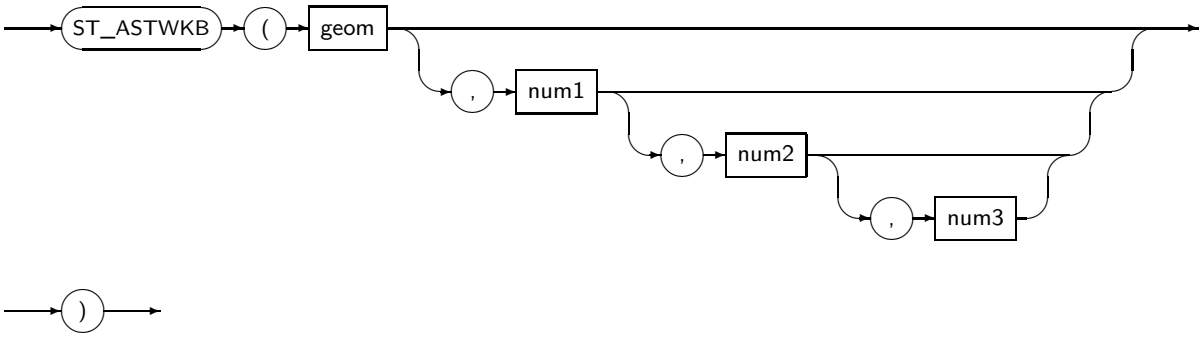
4.9. ST_ASTWKB(#)

ST_ASTWKB는 GEOMETRY 객체를 TWKB(Tiny Well Known Binary) 형태로 반환하는 함수이다.

ST_ASTWKB의 세부 내용은 다음과 같다.

- 문법

st_astwkb



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	TWKB로 표현될 좌표의 소수점 이하 유효숫자의 갯수이다. 기본값으로 0이 설정되어 있다.
num2	반환될 TWKB에 인코딩된 GEOMETRY 객체의 길이 요소를 포함할지를 결정하는 인자이다. 1을 넣을 경우 반환되는 TWKB가 길이를 포함하게된다. 기본 설정은 길이를 포함하지 않는다.
num3	반환될 TWKB에 GEOMETRY 객체의 경계상자를 포함할지를 결정하는 인자이다. 1을 넣을 경우 반환되는 TWKB가 경계 상자를 포함하게된다. 기본 설정은 경계 상자를 포함하지 않는다.

- 예제

  다음은 ST_ASTWKB 함수를 사용하는 예이다.

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

ST_ASTWKB(GEOM)
--------------------------------------------------------------------------------
01000202
04000202020202
02000202020202
0500020202020202020202020204
0300010502020200000201000001
0300030500000018180000171700050C1400020600000105000500 0D0006060000050500
0600020105020202000002010000010105040400040400000030300
0700020100020202000204040202
0710
0700020100020202000204040202
```

```
SQL> SELECT ST_ASTWKB(GEOM, 3, 1, 1) FROM GIS;

ST_ASTWKB(GEOM,3,1,1)
--------------------------------------------------------------------------------
61030AD00F00D00F00D00FD00F
640311D00FD00FD00FD00F02D00FD00FD00FD00F
620311D00FD00FD00FD00F02D00FD00FD00FD00F
65031BD00FF02ED00FC03E0202D00FD00FD00FD00F02D00FD00FD00FA01F
63031AD00FD00FD00FD00F0105D00FD00FD00F0000D00FCF0F0000CF0F
63033E00C0BB0100C0BB0103050000000C0BB01C0BB010000BFBB01BFBB010005E05DA09C0100D00F
F02E0000CF0FEF2E000500AF6D00F02EF02E0000EF2EEF2E00
66032DD00FC03ED00FC03E020105D00FD00FD00F0000D00FCF0F0000CF0F0105A01FA01F00A01FA0
1F00009F1F9F1F00
67032AD00FA01FD00FA01F0261030AD00F00D00F00D00FD00F620311A01FD00FA01FD00F02A01FA0
1FD00FD00F
671200
67032AD00FA01FD00FA01F0261030AD00F00D00F00D00FD00F620311A01FD00FA01FD00F02A01FA0
1FD00FD00F
```

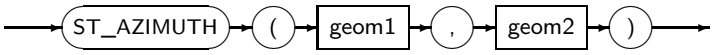
4.10. ST_AZIMUTH($)

ST_AZIMUTH은 POINT 객체 1과 POINT 객체 2를 잇는 선분과, POINT 객체 1의 수직 상방 선분이 이루는 각을 시계방향으로 측정하여 호도 단위로 반환한다. 이는 회전체 좌표계 상에서 북극기준 방위각에 해당한다. 회전체 좌표계 상의 GEOMETRY와 평면 좌표계 GEOMETRY를 인자로 받을 수 있다. POINT가 아닌 타입의 GEOMETRY 객체를 넣을 경우 예외를 반환한다.

ST_AZIMUTH의 세부 내용은 다음과 같다.

- 문법

st_azimuth



- 구성요소

구성요소	설명
geom1	POINT 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	POINT 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_AZIMUTH 함수를 사용하는 예이다.

```
SQL> SELECT ST_AZIMUTH(ST_GEOMFROMTEXT('POINT(1 1)', 4326),
     ST_GEOMFROMTEXT('POINT(1 0)',4326)) AZIMUTH FROM DUAL;

   AZIMUTH
----------
3.14159265
```

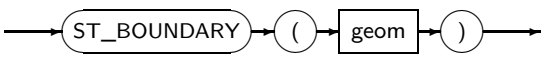
4.11. ST_BOUNDARY

ST_BOUNDARY는 주어진 GEOMETRY 객체의 경계를 반환하는 함수이다. GEOMETRY 객체의 타입이 GEOMETRYCOLLECTION일 경우 에러를 반환한다.

ST_BOUNDARY의 세부 내용은 다음과 같다.

- 문법

st_boundary



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_BOUNDARY 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM),ST_ASTEXT(ST_BOUNDARY(GEOM))
     FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) NOT LIKE 'GEOMETRYCOLLECTION';

ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(ST_BOUNDARY(GEOM))
--------------------------------------------------------------------------------
POINT(1 1)
GEOMETRYCOLLECTION EMPTY

MULTIPOINT((1 1),(2 2))
GEOMETRYCOLLECTION 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)

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))
MULTILINESTRING((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))

MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
MULTILINESTRING((1 1,2 1,2 2,1 2,1 1),(3 3,3 5,5 5,5 3,3 3))
```

4.12. ST_BUFFER($)

ST_BUFFER는 주어진 GEOMETRY 객체로부터 일정 거리 안에 있는 모든 점을 표현하는 GEOMETRY 객체를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다.

ST_BUFFER의 세부 내용은 다음과 같다.

- 문법

st_buffer

- 구성요소

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

- 예제

  다음은 ST_BUFFER 함수를 사용하는 예이다.

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

ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
ST_AREA(ST_BUFFER(GEOM,10))
---------------------------
POINT(1 1)
                 312.144515
```

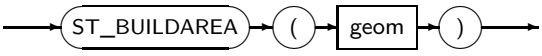
4.13. ST_BUILDAREA(#)

ST_BUILDAREA는 주어진 GEOMETRY의 선분 구성요소로 형성될 수 있는 POLYGON을 반환하는 함수이다. 입력 GEOMETRY의 선분 구성요소가 POLYGON을 형성하지 못할 경우 NULL을 반환한다.

ST_BUILDAREA의 세부 내용은 다음과 같다.

- 문법

  st_buildarea



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

  다음은 ST_BUILDAREA 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_BUILDAREA(GEOM))
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION EMPTY
```

```
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION EMPTY
POLYGON((1 1,1 2,2 2,2 1,1 1))
POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,9 10,9 11,6 11,6 10),(6 3,9 3,9 6,6 6,6
3))
MULTIPOLYGON(((1 1,1 2,2 2,2 1,1 1)),((3 3,3 5,5 5,5 3,3 3)))
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION EMPTY
```

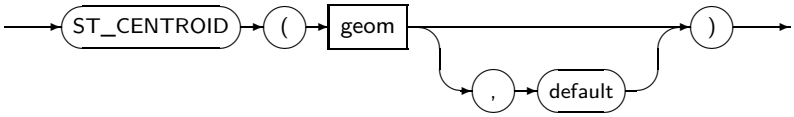
4.14. ST_CENTROID($)

ST_CENTROID는 주어진 GEOMETRY 객체의 중심을 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다. 더 빠른 연산을 원하면 구면좌표계에 대한 연산으로 바꿀 수 있다.

ST_CENTROID의 세부 내용은 다음과 같다.

- 문법

st_centroid



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
default	연산 방식을 설정한다. 좌표계 정보가 회전체 좌표계인 경우 기본으로 회전타원체에 대한 연산으로 설정되어있고, 0을 인자로 주면 구면좌표계에 대해 연산한다.

- 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((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))
POINT(5.8636363636364 6)
```

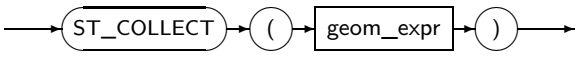
4.15. ST_COLLECT

ST_COLLECT는 GEOMETRY들을 MULTI 타입의 도형으로 묶어서 반환하는 집합 함수이다. 집합 함수이므로 GEOMETRY 객체들을 한데 모은 집합에 대한 결과를 반환한다. 변형된 형태로 두 GEOMETRY 객체를 인자로 받을 수 있다.

ST_COLLECT의 세부 내용은 다음과 같다.

- 문법

st_collect



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 임의의 연산식이다.

- 예제

다음은 ST_COLLECT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_COLLECT(A.GEOM, B.GEOM))
     FROM GIS A,GIS B WHERE ST_GEOMETRYTYPE(A.GEOM) LIKE 'POINT'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';

ST_ASTEXT(ST_COLLECT(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(1 1,2 2))
```

- 예제

다음은 ST_COLLECT 함수에 Table의 geometry column을 인자로 사용하는 예이다.

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

ST_ASTEXT(ST_COLLECT(GEOM))
```

```
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(POINT(1 1),MULTIPOINT((1 1),(2 2)),LINESTRING(1 1,2 2),MULTIL
INESTRING((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 0,0 0),(6 10,6 11,9 11,9 10,6 10),(6 3,6 6,9 6,9 3,6 3)),MULTIPOLYGON((
(1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3))))
```

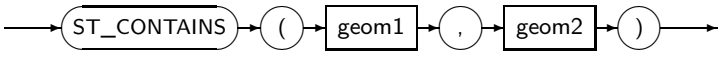
4.16. ST_CONTAINS

ST_CONTAINS는 GEOMETRY 객체 2가 GEOMETRY 객체 1의 외부에 존재하지 않고, 객체 1의 내부에 하나 이상의 점이 존재할 때만 1을 반환하는 함수이다.

ST_CONTAINS의 세부 내용은 다음과 같다.

- 문법

st_contains



- 구성요소

구성요소	설명
geom1	포함하는 GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	포함되는 GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'MULTILINESTRING'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';

ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
MULTILINESTRING((1 1,2 2),(3 3,4 5))
LINESTRING(1 1,2 2)
```

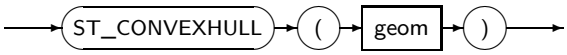
4.17. ST_CONVEXHULL

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 타입이어야 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 0,0 0),(6 10,6 11,9 11,9 10,6 10),(6 3,6 6,9 6,9 3,6
3))
POLYGON((0 0,0 12,12 12,12 0,0 0))
```

```
MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
POLYGON((1 1,1 2,3 5,5 5,5 3,2 1,1 1))

GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
LINESTRING(1 1,3 3)

GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION EMPTY

GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
LINESTRING(1 1,3 3)
```

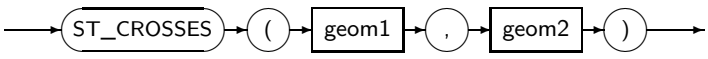
4.18. ST_CROSSES

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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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)
--------------------------------------------------------------------------------
```

```
POLYGON((1 1,2 1,2 2,1 2,1 1))
MULTILINESTRING((1 1,2 2),(3 3,4 5))

MULTILINESTRING((1 1,2 2),(3 3,4 5))
POLYGON((1 1,2 1,2 2,1 2,1 1))
```

4.19. ST_COVEREDBY(#)

ST_COVEREDBY는 GEOMETRY 객체 1의 어떤 점도 GEOMETRY 객체 2의 외부에 있지 않은 경우 1을 반환하는 함수이다. 회전체 좌표계 상의 GEOMETRY만을 인자로 받으며, 그렇지 않은 경우 예외를 반환한다.

ST_COVEREDBY의 세부 내용은 다음과 같다.

- 문법

st_coveredby

- 구성요소

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

- 예제

다음은 ST_COVEREDBY 함수를 사용하는 예이다.

```
SQL> SELECT ST_COVEREDBY(ST_GEOMFROMTEXT('POINT(0 0)',4326),
     ST_BUFFER(ST_GEOMFROMTEXT('POINT(0 0)',4326),1)) COVEREDBY FROM DUAL;

 COVEREDBY
----------
         1
```

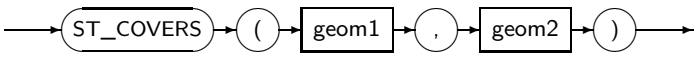
4.20. ST_COVERS(#)

ST_COVERS 는 GEOMETRY 객체 2의 어떤 점도 GEOMETRY 객체 1의 외부에 있지 않은 경우 1을 반환하는 함수이다. 회전체 좌표계 상의 GEOMETRY만을 인자로 받으며, 그렇지 않은 경우 예외를 반환한다.

ST_COVERS의 세부 내용은 다음과 같다.

- 문법

st_covers



- 구성요소

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

- 예제

다음은 ST_COVERS 함수를 사용하는 예이다.

```
SQL> SELECT ST_COVERS(ST_BUFFER(ST_GEOMFROMTEXT('POINT(0 0)',4326),1),
     ST_GEOMFROMTEXT('POINT(0 0)',4326)) COVERS FROM DUAL;

    COVERS
----------
         1
```

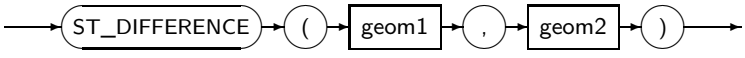
4.21. ST_DIFFERENCE($)

ST_DIFFERENCE는 GEOMETRY 객체 2와 겹치지 않는 객체 1의 부분을 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다.

ST_DIFFERENCE의 세부 내용은 다음과 같다.

- 문법

st_difference



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'LINESTRING'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';

ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(ST_DIFFERENCE(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2)
LINESTRING(1 1,2 2)
GEOMETRYCOLLECTION EMPTY
```

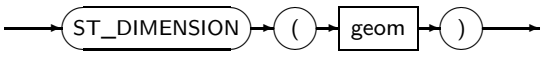
4.22. ST_DIMENSION

ST_DIMENSION은 GEOMETRY 객체의 차원을 반환하는 함수이다.

ST_DIMENSION의 세부 내용은 다음과 같다.

- 문법

  st_dimension



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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),(6 3,6 6,9 6,9 3,6
3))

                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))

                0
GEOMETRYCOLLECTION EMPTY

                1
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

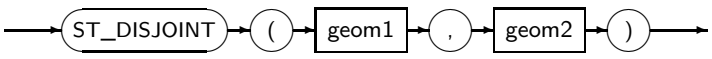
4.23. ST_DISJOINT

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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'LINESTRING'
     AND  ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';

ST_DISJOINT(A.GEOM,B.GEOM)
--------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                          0
LINESTRING(1 1,2 2)
LINESTRING(1 1,2 2)
```

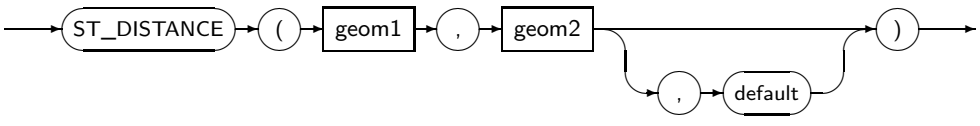
4.24. ST_DISTANCE($)

ST_DISTANCE는 두 GEOMETRY 객체 사이의 최단 거리를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다. 더 빠른 연산을 원하면 구면좌표계에 대한 연산으로 바꿀 수 있다.

ST_DISTANCE의 세부 내용은 다음과 같다.

- 문법

st_distance



- 구성요소

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

구성요소	설명
default	연산 방식을 설정한다. 좌표계 정보가 회전체 좌표계인 경우 기본으로 회전타원체에 대한 연산으로 설정되어있고, 0을 인자로 주면 구면좌표계에 대해 연산한다.

- 예제

  다음은 ST_DISTANCE 함수를 사용하는 예이다.

```
SQL> SELECT ST_DISTANCE(ST_GEOMFROMTEXT('LINESTRING(2 2,0 0)'),
     ST_GEOMFROMTEXT('LINESTRING(4 2,6 0)')) DISTANCE FROM DUAL;

  DISTANCE
----------
         2
```

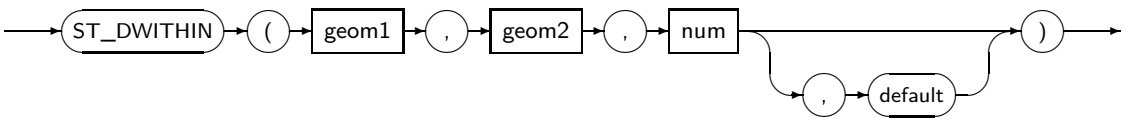
4.25. ST_DWITHIN

ST_DWITHIN는 두 GEOMETRY 객체가 지정된 거리안에 있는지 여부를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다. 더 빠른 연산을 원하면 구면좌표계에 대한 연산으로 바꿀 수 있다.

ST_DWITHIN의 세부 내용은 다음과 같다.

- 문법

st_dwithin



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num	지정된 거리를 입력한다. GEOMETRY의 경우 설정된 좌표계의 단위를, GEOGRAPHY의 경우 meter 단위를 사용한다.
default	연산 방식을 설정한다. 좌표계 정보가 회전체 좌표계인 경우 기본으로 회전타원체에 대한 연산으로 설정되어있고, 0을 인자로 주면 구면좌표계에 대해 연산한다.

- 예제

다음은 ST_DWITHIN 함수를 사용하는 예이다.

```
SQL> SELECT ST_DWITHIN(ST_GEOMFROMTEXT('LINESTRING(2 2,0 0)'),
     ST_GEOMFROMTEXT('LINESTRING(4 2,6 0)'), 3) DWITHIN FROM DUAL;

   DWITHIN
----------
         1
```

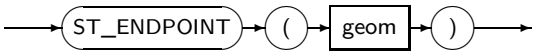
4.26. ST_ENDPOINT

ST_ENDPOINT는 LINESTRING GEOMETRY 객체의 마지막 점을 POINT 객체로 반환하는 함수이다. LINESTRING이 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_ENDPOINT의 세부 내용은 다음과 같다.

- 문법

st_endpoint



- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 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)
```

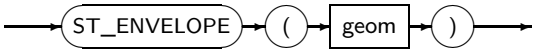
4.27. ST_ENVELOPE

ST_ENVELOPE는 주어진 GEOMETRY 객체를 둘러싸는 사각형을 POLYGON 형태로 반환하는 함수이다.

ST_ENVELOPE의 세부 내용은 다음과 같다.

- 문법

st_envelope



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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))
GEOMETRYCOLLECTION EMPTY
POLYGON((1 1,1 3,3 3,3 1,1 1))
```

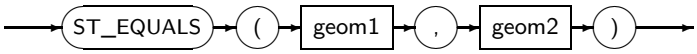
4.28. ST_EQUALS

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 타입이어야 한다.
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POINT'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'POINT';

ST_EQUALS(A.GEOM,B.GEOM)
------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                       1
POINT(1 1)
POINT(1 1)
```

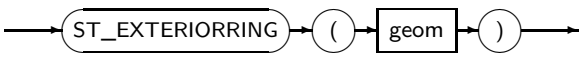
4.29. ST_EXTERIORRING

ST_EXTERIORRING은 POLYGON GEOMETRY 객체의 외부 링을 반환하는 함수이다. POLYGON이 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_EXTERIORRING의 세부 내용은 다음과 같다.

- 문법

st_exteriorring



- 구성요소

구성요소	설명
geom	POLYGON 객체를 나타내는 GEOMETRY 타입이어야 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)
```

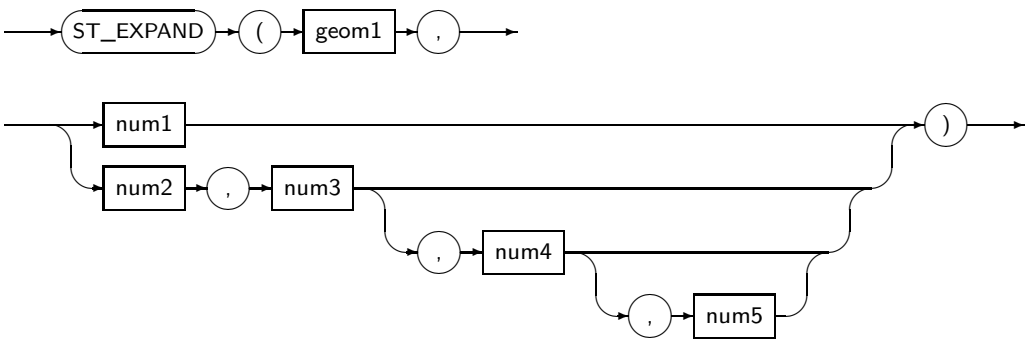
4.30. ST_EXPAND

ST_EXPAND은 입력 GEOMETRY를 모든 방향으로 확장한 경계 상자에 해당하는 POLYGON을 반환한다. 비어있는 GEOMETRY 객체에 대해선 확장에 관계없이 NULL을 반환한다.

ST_EXPAND의 세부 내용은 다음과 같다.

- 문법

st_expand



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	GEOMETRY 객체의 모든 축 방향으로 확장시킬 거리를 입력한다. 입력한 거리만큼 확장된 POLYGON을 형성한다.
num2	X 방향으로 확장시킬 거리를 입력한다. 입력한 거리만큼 X 방향으로 확장된 POLYGON을 형성한다.

구성요소	설명
num3	Y 방향으로 확장시킬 거리를 입력한다. 입력한 거리만큼 Y 방향으로 확장된 POLYGON을 형성한다.
num4	Z 방향으로 확장시킬 거리를 입력한다. 입력한 거리만큼 Z 방향으로 확장된 POLYGON을 형성한다.
num5	M 방향으로 확장시킬 거리를 입력한다. 입력한 거리만큼 M 방향으로 확장된 POLYGON을 형성한다.

- 예제

다음은 ST_EXPAND 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_EXPAND(GEOM,1,2)) FROM GIS;

ST_ASTEXT(ST_EXPAND(GEOM,1,2))
--------------------------------------------------------------------------------
POLYGON((0 -1,0 3,2 3,2 -1,0 -1))
POLYGON((0 -1,0 4,3 4,3 -1,0 -1))
POLYGON((0 -1,0 4,3 4,3 -1,0 -1))
POLYGON((0 -1,0 7,5 7,5 -1,0 -1))
POLYGON((0 -1,0 4,3 4,3 -1,0 -1))
POLYGON((-1 -2,-1 14,13 14,13 -2,-1 -2))
POLYGON((0 -1,0 7,6 7,6 -1,0 -1))
POLYGON((0 -1,0 5,4 5,4 -1,0 -1))
GEOMETRYCOLLECTION EMPTY
POLYGON((0 -1,0 5,4 5,4 -1,0 -1))
```

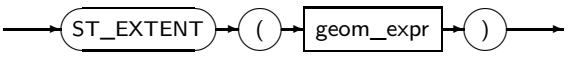
4.31. ST_EXTENT

ST_EXTENT은 GEOMETRY를 포함하는 경계 상자에 해당하는 POLYGON을 반환하는 집합 함수이다. 집합 함수이므로 GEOMETRY 객체들을 한데 모은 집합에 대한 결과를 반환한다.

ST_EXTENT의 세부 내용은 다음과 같다.

- 문법

st_extent



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 임의의 연산식이다.

- 예제

  다음은 ST_EXTENT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_EXTENT(GEOM))
--------------------------------------------------------------------------------
POLYGON((0 0,0 12,12 12,12 0,0 0))
```

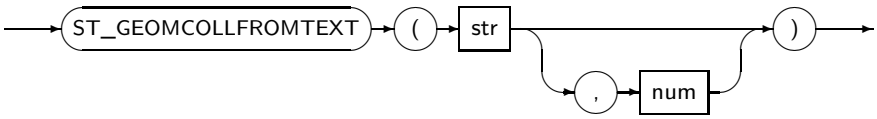
4.32. ST_GEOMCOLLFROMTEXT

ST_GEOMCOLLFROMTEXT는 주어진 WKT와 SRID를 바탕으로 GEOMETRYCOLLECTION 객체를 반환하는 함수이다.

ST_GEOMCOLLFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_geomcollfromtext



- 구성요소

구성요소	설명
str	GEOMETRYCOLLECTION 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	GEOMETRYCOLLECTION 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_GEOMCOLLFROMTEXT 함수를 사용하는 예이다.

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

GEOMCOLLFROMTEXT
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

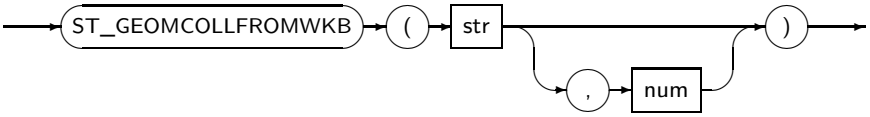
4.33. ST_GEOMCOLLFROMWKB

ST_GEOMCOLLFROMWKB는 주어진 WKB와 SRID를 바탕으로 GEOMETRYCOLLECTION 객체를 반환하는 함수이다.

ST_GEOMCOLLFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_geomcollfromwkb



- 구성요소

구성요소	설명
str	GEOMETRYCOLLECTION 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	GEOMETRYCOLLECTION 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_GEOMCOLLFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_GEOMCOLLFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
GEOMETRYCOLLECTION EMPTY
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

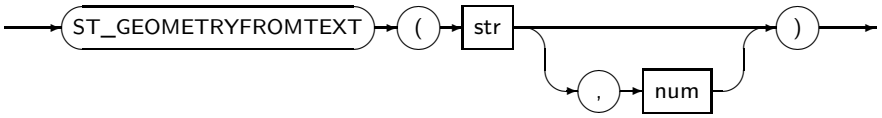
4.34. ST_GEOMETRYFROMTEXT

ST_GEOMETRYFROMTEXT는 주어진 WKT와 SRID를 바탕으로 GEOMETRY 객체로 반환하는 함수이다. 3차원을 표현하는 EWKT(Extended Well-Known Text) 형식의 TEXT도 지원한다. ST_GEOMFROMTEXT와 동일하다.

ST_GEOMETRYFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_geometryfromtext



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_GEOMETRYFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_GEOMETRYFROMTEXT('MULTIPOINT(11,22)'))
--------------------------------------------------------------------------------
MULTIPOINT((1 1),(2 2))
```

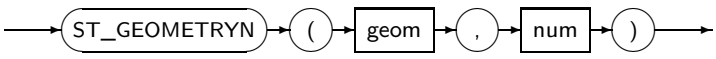
4.35. ST_GEOMETRYN

ST_GEOMETRYN은 GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING, MULTIPOLYGON 객체에서 N번째 GEOMETRY 객체를 반환하는 함수이다. MULTI가 아닌 GEOMETRY를 인자로 넣을 경우, N이 1이면 인자로 넣어준 GEOMETRY 객체를 그대로 반환하고, 1이 아닌 N이 들어올 경우 NULL을 반환한다. N이 내부 GEOMETRY 개수의 범위를 벗어나는 경우에도 NULL을 반환한다.

ST_GEOMETRYN의 세부 내용은 다음과 같다.

- 문법

st_geometryn



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

구성요소	설명
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)

LINESTRING(2 2,3 3)
```

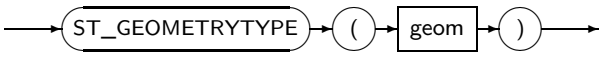
4.36. ST_GEOMETRYTYPE

ST_GEOMETRYTYPE은 GEOMETRY 객체의 타입을 반환하는 함수이다.

ST_GEOMETRYTYPE의 세부 내용은 다음과 같다.

- 문법

  st_geometrytype



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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
GEOMETRYCOLLECTION
GEOMETRYCOLLECTION
```

4.37. ST_GEOMFROMGEOJSON

ST_GEOMFROMGEOJSON는 주어진 GEOJSON 표현식으로부터 GEOMETRY 객체를 생성하여 반환하는 함수이다. JSON 도형 조각만을 인자로 받을 수 있다. 완전한 JSON 문서는 인자로 받을 수 없다.

ST_GEOMFROMGEOJSON의 세부 내용은 다음과 같다.

- 문법

st_geomfromgeojson

- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하는 GEOJSON 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.

- 예제

다음은 ST_GEOMFROMGEOJSON 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMFROMGEOJSON('{"type":
     "LineString","coordinates": [[30, 10], [10, 30], [40, 40]]}')) FROM DUAL;



ST_ASTEXT(ST_GEOMFROMGEOJSON('{"TYPE":"LINESTRING","COORDINATES":[[30,10],[10,30
--------------------------------------------------------------------------------
LINESTRING(30 10,10 30,40 40)
```

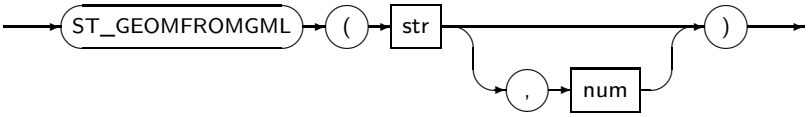
4.38. ST_GEOMFROMGML

ST_GEOMFROMGML는 주어진 GML 표현식으로부터 GEOMETRY 객체를 생성하여 반환하는 함수이다. GML 도형 조각만을 인자로 받을 수 있다. 완전한 GML 문서는 인자로 받을 수 없다.

ST_GEOMFROMGML의 세부 내용은 다음과 같다.

- 문법

st_geomfromgml



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하는 GML 형태의 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_GEOMFROMGML 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMFROMGML(
     '<gml:LineString><coordinates>1,1 2,2</coordinates></gml:LineString>'))
     FROM DUAL;

ST_ASTEXT(ST_GEOMFROMGML('<GML:LINESTRING><GML:COORDINATES>1,12,2</GML:
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2)
```

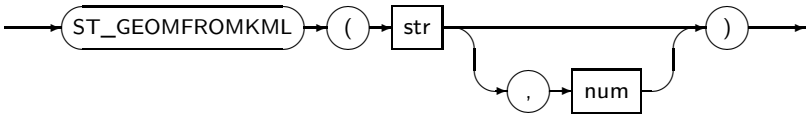
4.39. ST_GEOMFROMKML

ST_GEOMFROMKML는 주어진 KML 표현식으로부터 GEOMETRY 객체를 생성하여 반환하는 함수이다. KML 도형 조각만을 인자로 받을 수 있다. 완전한 KML 문서는 인자로 받을 수 없다.

ST_GEOMFROMKML의 세부 내용은 다음과 같다.

- 문법

st_geomfromkml



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하는 KML 형태의 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_GEOMFROMKML 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMFROMKML('<LineString>
     <coordinates>1,1 2,2</coordinates></LineString>')) FROM DUAL;

ST_ASTEXT(ST_GEOMFROMKML('<LINESTRING><COORDINATES>1,12,2</COORDINATES>
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2)
```

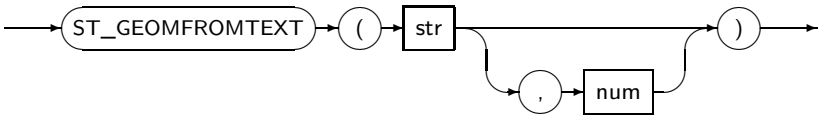
4.40. ST_GEOMFROMTEXT

ST_GEOMFROMTEXT는 주어진 WKT와 SRID를 바탕으로 GEOMETRY 객체로 반환하는 함수이다. 3차원을 표현하는 EWKT(Extended Well-Known Text) 형식의 TEXT도 지원한다.

ST_GEOMFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_geomfromtext



- 구성요소

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

구성요소	설명
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_GEOMFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_GEOMFROMTEXT('MULTIPOINT(11,22)'))
--------------------------------------------------------------------------------
MULTIPOINT((1 1),(2 2))
```

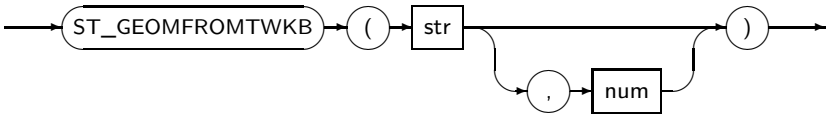
4.41. ST_GEOMFROMTWKB(#)

ST_GEOMFROMTWKB는 주어진 TWKB를 GEOMETRY 객체로 반환하는 함수이다.

ST_GEOMFROMTWKB의 세부 내용은 다음과 같다.

- 문법

st_geomfromtwkb



- 구성요소

구성요소	설명
str	GEOMETRY 객체를 표현하기 위한 TWKB 형식이 저장되어 있는 바이너리 타입(BLOB 또는 RAW) 값이다.
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_GEOMFROMTWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMFROMTWKB('02000202020202'))
     FROM DUAL;

ST_ASTEXT(ST_GEOMFROMTWKB('02000202020202'))
```

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

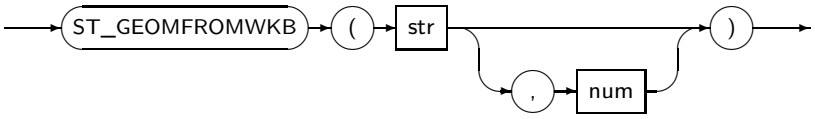
4.42. ST_GEOMFROMWKB

ST_GEOMFROMWKB는 주어진 WKB와 SRID를 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.
num	GEOMETRY 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_GEOMFROMWKB 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_GEOMFROMWKB(ST_ASBINARY(GEOM)))
     FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) LIKE 'MULTIPOINT';

ST_ASTEXT(ST_GEOMFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
MULTIPOINT((1 1),(2 2))
```

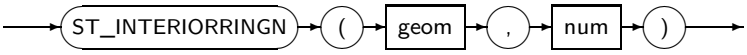
4.43. ST_INTERIORRINGN

ST_INTERIORRINGN은 POLYGON 객체의 N번째 내부 링을 반환하는 함수이다. POLYGON이 아닌 GEOMETRY를 인자로 넣을 경우 NULL을 반환한다. N이 내부 RING의 개수의 범위를 벗어나는 경우에도 NULL을 반환한다.

ST_INTERIORRINGN의 세부 내용은 다음과 같다.

- 문법

st_interiorringn



- 구성요소

구성요소	설명
geom	POLYGON 객체를 나타내는 GEOMETRY 타입이어야 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)
```

4.44. ST_INTERSECTION($)

ST_INTERSECTION은 GEOMETRY 객체 1과 GEOMETRY 객체 2가 공유하는 부분을 나타내는 GEOMETRY 객체를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다.

ST_INTERSECTION의 세부 내용은 다음과 같다.

- 문법

st_intersection

- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POLYGON'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(ST_INTERSECTION(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((2 1,1 1,1 2,2 2,2 1))

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))
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,1 2,2 2,2 1,1 1))

POLYGON((1 1,2 1,2 2,1 2,1 1))
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))
POLYGON((1 1,1 2,2 2,2 1,1 1))

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))
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))
POLYGON((0 0,0 12,12 12,12 0,0 0),(6 11,6 10,9 10,9 11,6 11),(6 6,6 3,9 3,9 6,6
6))
```

4.45. ST_INTERSECTS($)

ST_INTERSECTS는 두 GEOMETRY 객체가 공유하는 부분이 있을 때 1을 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다. 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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POLYGON'
     AND  ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_INTERSECTS(A.GEOM,B.GEOM)
----------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                           1
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))

                           1
POLYGON((1 1,2 1,2 2,1 2,1 1))
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))

                           1
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))
POLYGON((1 1,2 1,2 2,1 2,1 1))

                           1
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))
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))
```

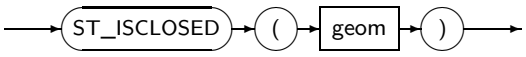
4.46. ST_ISCLOSED

ST_ISCLOSED는 LINESTRING 객체의 첫 점과 마지막 점이 동일할 때 1을 반환하는 함수이다. POINT나 POLYGON 객체에 대해서는 항상 1을 반환한다. MULTI 타입의 GEOMETRY 객체에 대해서는 내부 GEOMETRY가 모두 1일 경우 1을 반환한다.

ST_ISCLOSED의 세부 내용은 다음과 같다.

- 문법

  st_isclosed

- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

  다음은 ST_ISCLOSED 함수를 사용하는 예이다.

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

ST_ISCLOSED(GEOM)
-----------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
                0
LINESTRING(1 1,2 2)
```

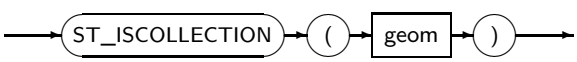
4.47. ST_ISCOLLECTION

ST_ISCOLLECTION는 GEOMETRY 객체가 GEOMETRYCOLLECTION, MULTI GEOMETRY 인 경우 1을 반환하는 함수이다.

ST_ISCOLLECTION의 세부 내용은 다음과 같다.

- 문법

  st_iscollection

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_ISCOLLECTION 함수를 사용하는 예이다.

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

ST_ISCOLLECTION(GEOM)
---------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
                    0
POINT(1 1)

                    1
MULTIPOINT((1 1),(2 2))

                    0
LINESTRING(1 1,2 2)

                    1
MULTILINESTRING((1 1,2 2),(3 3,4 5))

                    0
POLYGON((1 1,2 1,2 2,1 2,1 1))

                    0
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))

                    1
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
GEOMETRYCOLLECTION EMPTY

                    1
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

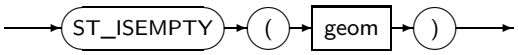
4.48. ST_ISEMPTY

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 타입이어야 한다.

- 예제

  다음은 ST_ISEMPTY 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(GEOM)
     FROM GIS WHERE ST_ISEMPTY(GEOM) > 0;

ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION EMPTY
```

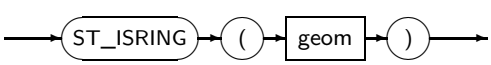
4.49. ST_ISRING

ST_ISRING은 주어진 LINESTRING이 닫혀있고(첫 점과 마지막 점이 동일) 심플하면 1을 반환하는 함수이다. LINESTRING이 아닌 타입의 GEOMETRY 객체를 넣을 경우 항상 0을 반환한다.

ST_ISRING의 세부 내용은 다음과 같다.

- 문법

  st_isring

- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 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)
```

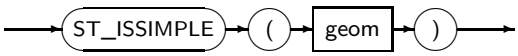
4.50. ST_ISSIMPLE

ST_ISSIMPLE은 GEOMETRY 객체가 자체 교차점이나 자체 접촉점을 가지지 않으면 1을 반환하는 함수이다. GEOMETRYCOLLECTION에 대해서는 예외를 반환한다.

ST_ISSIMPLE의 세부 내용은 다음과 같다.

- 문법

  st_issimple



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

  다음은 ST_ISSIMPLE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ISSIMPLE(GEOM),ST_ASTEXT(GEOM) FROM GIS
     WHERE ST_GEOMETRYTYPE(GEOM) NOT LIKE 'GEOMETRYCOLLECTION';

ST_ISSIMPLE(GEOM)
-----------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
               1
POINT(1 1)
```

```
                1
MULTIPOINT((1 1),(2 2))

                1
LINESTRING(1 1,2 2)

                1
MULTILINESTRING((1 1,2 2),(3 3,4 5))

                1
POLYGON((1 1,2 1,2 2,1 2,1 1))

                1
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))

                1
MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
```

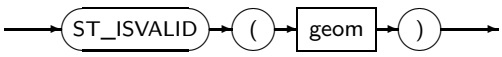
4.51. ST_ISVALID

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

ST_ISVALID의 세부 내용은 다음과 같다.

- 문법

  st_isvalid

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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))'))
     FROM DUAL;
```

```
ST_ISVALID(ST_GEOMFROMTEXT('POLYGON((00,11,01,10,00))'))
---------------------------------------------------------
                                                        0
```

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

ST_ISVALID(ST_GEOMFROMTEXT('POLYGON((00,10,11,01,00))'))
---------------------------------------------------------
                                                        1
```

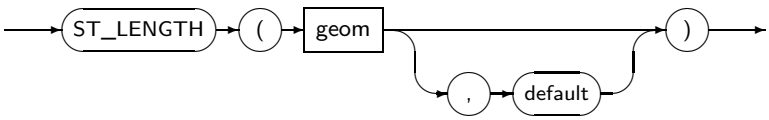
4.52. ST_LENGTH($)

ST_LENGTH는 LINESTRING 객체의 길이를 반환하는 함수이다. POINT, POLYGON에 대해서는 0을 반환한다. MULTI 타입의 GEOMETRY 객체에 대해서는 내부의 GEOMETRY들의 계산 결과의 합을 값을 반환한다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다. 더 빠른 연산을 원하면 구면좌표계에 대한 연산으로 바꿀 수 있다.

ST_LENGTH의 세부 내용은 다음과 같다.

- 문법

st_length



- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다.
default	연산 방식을 설정한다. 좌표계 정보가 회전체 좌표계인 경우 기본으로 회전타원체에 대한 연산으로 설정되어있고, 0을 인자로 주면 구면좌표계에 대해 연산한다.

- 예제

다음은 ST_LENGTH 함수를 사용하는 예이다.

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

ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
```

```
ST_LENGTH(GEOM)
---------------
LINESTRING(1 1,2 2)
     1.41421356

MULTILINESTRING((1 1,2 2),(3 3,4 5))
     3.65028154
```

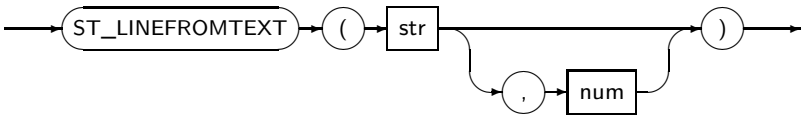
4.53. ST_LINEFROMTEXT

ST_LINEFROMTEXT는 주어진 WKT와 SRID를 바탕으로 LINESTRING 객체로 반환하는 함수이다.

ST_LINEFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_linefromtext



- 구성요소

구성요소	설명
str	LINESTRING 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	LINESTRING 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_LINEFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_LINEFROMTEXT('LINESTRING(11,22)'))
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2)
```

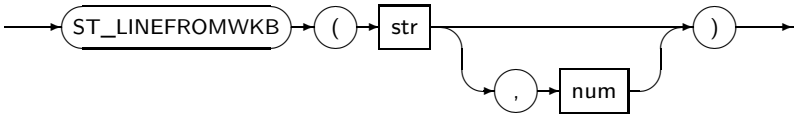
4.54. ST_LINEFROMWKB

ST_LINEFROMWKB는 주어진 WKB와 SRID를 바탕으로 LINESTRING 객체로 반환하는 함수이다.

ST_LINEFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_linefromwkb



- 구성요소

구성요소	설명
str	LINESTRING 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	LINESTRING 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_LINEFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_LINEFROMWKB(ST_ASBINARY(GEOM)))

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

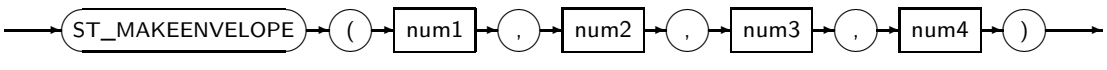
4.55. ST_MAKEENVELOPE

ST_MAKEENVELOPE는 주어진 최소값과 최대값으로 형성된 직사각형 POLYGON을 반환한다.

ST_MAKEENVELOPE의 세부 내용은 다음과 같다.

- 문법

st_makeenvelope



- 구성요소

구성요소	설명
num1	직사각형 POLYGON의 최소점(왼쪽 아래)의 x 좌표이다.
num2	직사각형 POLYGON의 최소점(왼쪽 아래)의 y 좌표이다.

구성요소	설명
num3	직사각형 POLYGON의 최대점(오른쪽 위)의 x 좌표이다.
num4	직사각형 POLYGON의 최대점(오른쪽 위)의 y 좌표이다.

- 예제

다음은 ST_MAKEENVELOPE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKEENVELOPE(-1,1,2,3)) FROM DUAL;

ST_ASTEXT(ST_MAKEENVELOPE(-1,1,2,3))
--------------------------------------------------------------------------------
POLYGON((-1 1,-1 3,2 3,2 1,-1 1))
```

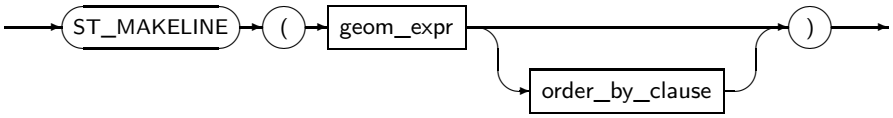
4.56. ST_MAKELINE

ST_MAKELINE는 POINT, MULTIPOINT, LINESTRING을 입력받아 LINESTRING을 반환한다. 집합 함수이므로 GEOMETRY 객체들을 한데 모은 집합에 대한 결과를 반환한다. ORDER BY 문구로 반환할 LINESTRING 객체의 형태를 설정할 수 있다.

ST_MAKELINE의 세부 내용은 다음과 같다.

- 문법

st_makeline



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 임의의 연산식이다. POINT, LINESTRING, MULTIPOINT 타입의 GEOMETRY 객체이어야 한다.
order_by_clause	GEOMETRY들을 어떻게 정렬할지를 명시한다.

- 예제

다음은 ST_MAKELINE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKELINE(A.GEOM, B.GEOM)) FROM GIS A, GIS B
     WHERE ST_GEOMETRYTYPE(A.GEOM) LIKE 'POINT'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';
```

```
ST_ASTEXT(ST_MAKELINE(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2)
```

- 예제

  다음은 ST_MAKELINE 함수에 Table의 geometry column을 인자로 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKELINE(GEOM)) FROM GIS
     WHERE ST_GEOMETRYTYPE(GEOM) IN ('POINT','MULTIPOINT','LINESTRING');

ST_ASTEXT(ST_MAKELINE(GEOM))
--------------------------------------------------------------------------------
LINESTRING(1 1,1 1,2 2,1 1,2 2)
```

- 예제

  다음은 ST_MAKELINE 함수에 order_by_clause를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKELINE(GEOM ORDER BY ID DESC))
     FROM GIS WHERE ST_GEOMETRYTYPE(GEOM) IN ('POINT','LINESTRING','MULTIPOINT');

ST_ASTEXT(ST_MAKELINE(GEOMORDERBYIDDESC))
--------------------------------------------------------------------------------
LINESTRING(1 1,2 2,1 1,2 2,1 1)
```

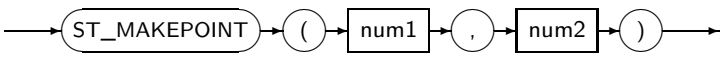
4.57. ST_MAKEPOINT

ST_MAKEPOINT는 2D POINT 객체를 반환한다.

ST_MAKEPOINT의 세부 내용은 다음과 같다.

- 문법

  st_makepoint



- 구성요소

구성요소	설명
num1	POINT의 x 좌표를 나타낸다.
num2	POINT의 y 좌표를 나타낸다.

- 예제

  다음은 ST_MAKEPOINT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKEPOINT(1,2)) FROM DUAL;

ST_ASTEXT(ST_MAKEPOINT(1,2))
--------------------------------------------------------------------------------
POINT(1 2)
```

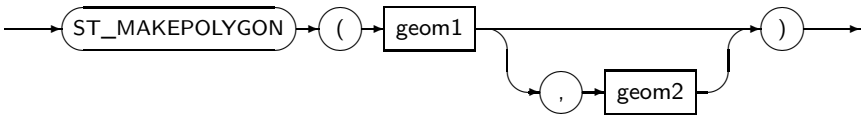
4.58. ST_MAKEPOLYGON

ST_MAKEPOLYGON는 주어진 구조로 형성된 POLYGON을 반환한다. 입력 도형이 닫힌 LINESTRING이어야 한다.

ST_MAKEPOLYGON의 세부 내용은 다음과 같다.

- 문법

st_makepolygon



- 구성요소

구성요소	설명
geom1	형성할 POLYGON의 SHELL을 나타내는 LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	형성할 POLYGON의 INTERIOR RING을 나타내는 LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다. INTERIOR RING이 다수인 경우 MULTILINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MAKEPOLYGON 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKEPOLYGON(ST_GEOMFROMTEXT(
     'LINESTRING(0 0,1 0,1 1,0 1,0 0)'))) FROM DUAL;

ST_ASTEXT(ST_MAKEPOLYGON(ST_GEOMFROMTEXT('LINESTRING(00,10,11,01,00)')))
--------------------------------------------------------------------------------
POLYGON((0 0,1 0,1 1,0 1,0 0))
```

```
SQL> SELECT ST_ASTEXT(ST_MAKEPOLYGON(
     ST_GEOMFROMTEXT('LINESTRING(0 0,10 0,10 10,0 10,0 0)'),
     ST_GEOMFROMTEXT('MULTILINESTRING((0 0,4 0,4 4,0 4,0 0),(5 5,7 5,7 7,5 7,5 5))
```

```
     '))) MAKEPOLYGON FROM DUAL;

MAKEPOLYGON
--------------------------------------------------------------------------------
POLYGON((0 0,10 0,10 10,0 10,0 0),(0 0,4 0,4 4,0 4,0 0),(5 5,7 5,7 7,5 7,5 5))
```

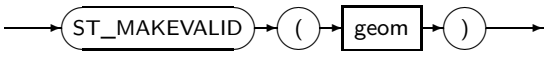
4.59. ST_MAKEVALID(#)

ST_MAKEVALID는 유효하지 않은 GEOMETRY를 점의 소실없이 유효한 GEOMETRY로 변경하여 반환한다. 이미 유요한 GEOMETRY의 경우 그대로 반환한다.

ST_MAKEVALID의 세부 내용은 다음과 같다.

- 문법

st_makevalid



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MAKEVALID 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_MAKEVALID(ST_GEOMFROMTEXT(
     'POLYGON((0 0,1 0,0 1,1 1,0 0))'))) FROM DUAL;

ST_ASTEXT(ST_MAKEVALID(ST_GEOMFROMTEXT('POLYGON((00,10,01,11,00))')))
--------------------------------------------------------------------------------
MULTIPOLYGON(((0.5 0.5,1 0,0 0,0.5 0.5)),((0.5 0.5,0 1,1 1,0.5 0.5)))
```

```
SQL> SELECT ST_ASTEXT(ST_MAKEVALID(ST_GEOMFROMTEXT(
     'POLYGON((0 0,1 0,1 1,0 1,0 0))'))) FROM DUAL;

ST_ASTEXT(ST_MAKEVALID(ST_GEOMFROMTEXT('POLYGON((00,10,11,01,00))')))
--------------------------------------------------------------------------------
POLYGON((0 0,1 0,1 1,0 1,0 0))
```

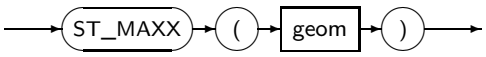
4.60. ST_MAXX

ST_MAXX는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 X 좌표값 중 큰 값을 반환하는 함수이다.

ST_MAXX의 세부 내용은 다음과 같다.

- 문법

st_maxx



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MAXX 함수를 사용하는 예이다.

```
SQL> SELECT ST_MAXX(ST_GEOMFROMTEXT('LINESTRING(-1 0,3 2)')) FROM DUAL;

ST_MAXX(ST_GEOMFROMTEXT('LINESTRING(00,32)'))
---------------------------------------------
                                            3
```

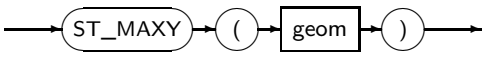
4.61. ST_MAXY

ST_MAXY는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 Y 좌표값 중 큰 값을 반환하는 함수이다.

ST_MAXY의 세부 내용은 다음과 같다.

- 문법

st_maxy



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MAXY 함수를 사용하는 예이다.

```
SQL> SELECT ST_MAXY(ST_GEOMFROMTEXT('LINESTRING(-1 0,3 2)')) FROM DUAL;

ST_MAXY(ST_GEOMFROMTEXT('LINESTRING(00,32)'))
---------------------------------------------
                                            2
```

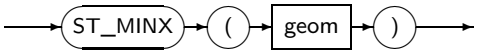
4.62. ST_MINX

ST_MINX는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 X 좌표값 중 작은 값을 반환하는 함수이다.

ST_MINX 의 세부 내용은 다음과 같다.

- 문법

st_minx



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MINX 함수를 사용하는 예이다.

```
SQL> SELECT ST_MINX(ST_GEOMFROMTEXT('LINESTRING(-1 0,3 2)')) FROM DUAL;

ST_MINX(ST_GEOMFROMTEXT('LINESTRING(-10,32)'))
----------------------------------------------
                                            -1
```

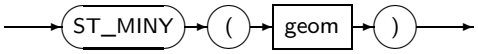
4.63. ST_MINY

ST_MINY는 해당 공간 객체를 둘러싸는 최소 경계 사각형의 Y 좌표값 중 작은 값을 반환하는 함수이다.

ST_MINY 의 세부 내용은 다음과 같다.

- 문법

st_miny



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MINY 함수를 사용하는 예이다.

```
SQL> SELECT ST_MINY(ST_GEOMFROMTEXT('LINESTRING(0 -1,3 2)')) FROM DUAL;

ST_MINY(ST_GEOMFROMTEXT('LINESTRING(0-1,32)'))
----------------------------------------------
                                            -1
```

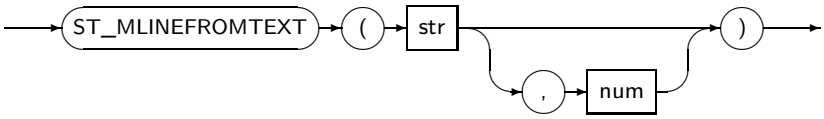
4.64. ST_MLINEFROMTEXT

ST_MLINEFROMTEXT는 주어진 WKT와 SRID를 바탕으로 MULTILINESTRING 객체로 반환하는 함수이다.

ST_MLINEFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mlinefromtext



- 구성요소

구성요소	설명
str	MULTILINESTRING 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTILINESTRING 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_MLINEFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MLINEFROMTEXT('MULTILINESTRING((11,22),(33,45))'))
--------------------------------------------------------------------------------
MULTILINESTRING((1 1,2 2),(3 3,4 5))
```

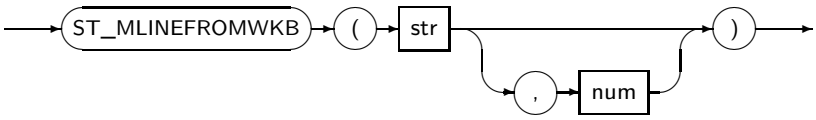
4.65. ST_MLINEFROMWKB

ST_MLINEFROMWKB는 주어진 WKB와 SRID를 바탕으로 MULTILINESTRING 객체로 반환하는 함수이다.

ST_MLINEFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mlinefromwkb



- 구성요소

구성요소	설명
str	MULTILINESTRING 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTILINESTRING 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_MLINEFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MLINEFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
MULTILINESTRING((1 1,2 2),(3 3,4 5))
```

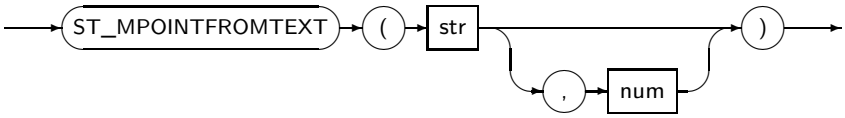
4.66. ST_MPOINTFROMTEXT

ST_MPOINTFROMTEXT는 주어진 WKT와 SRID를 바탕으로 MULTIPOINT 객체로 반환하는 함수이다.

ST_MPOINTFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mpointfromtext



- 구성요소

구성요소	설명
str	MULTIPOINT 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTIPOINT 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_MPOINTFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MPOINTFROMTEXT('MULTIPOINT(1010,2020)'))
--------------------------------------------------------------------------------
MULTIPOINT((10 10),(20 20))
```

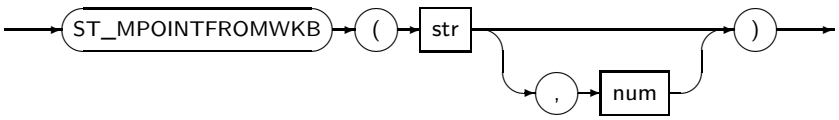
4.67. ST_MPOINTFROMWKB

ST_MPOINTFROMWKB는 주어진 WKB와 SRID를 바탕으로 MULTIPOINT 객체로 반환하는 함수이다.

ST_MPOINTFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mpointfromwkb



- 구성요소

구성요소	설명
str	MULTIPOINT 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTIPOINT 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_MPOINTFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MPOINTFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
MULTIPOINT((1 1),(2 2))
```

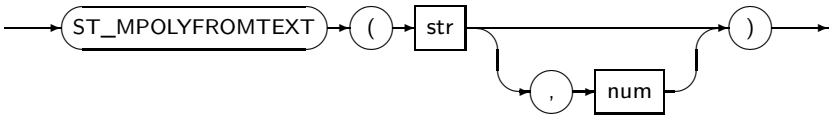
4.68. ST_MPOLYFROMTEXT

ST_MPOLYFROMTEXT는 주어진 WKT와 SRID를 바탕으로 MULTIPOLYGON 객체로 반환하는 함수이다.

ST_MPOLYFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_mpolyfromtext



- 구성요소

구성요소	설명
str	MULTIPOLYGON 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTIPOLYGON 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_MPOLYFROMTEXT 함수를 사용하는 예이다.

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

MPOLYFROMTEXT
--------------------------------------------------------------------------------
MULTIPOLYGON(((10 10,10 20,20 20,20 15,10 10)),((60 60,70 70,80 60,60 60)))
```

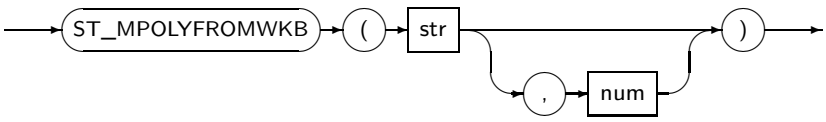
4.69. ST_MPOLYFROMWKB

ST_MPOLYFROMWKB는 주어진 WKB와 SRID를 바탕으로 MULTIPOLYGON 객체로 반환하는 함수이다.

ST_MPOLYFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_mpolyfromwkb



- 구성요소

구성요소	설명
str	MULTIPOLYGON 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	MULTIPOLYGON 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_MPOLYFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MPOLYFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
MULTIPOLYGON(((1 1,2 1,2 2,1 2,1 1)),((3 3,3 5,5 5,5 3,3 3)))
```

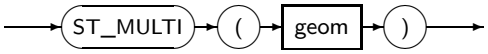
4.70. ST_MULTI

ST_MULTI는 GEOMETRY를 MULTI 타입의 GEOMETRY로 반환한다. MULTI 타입의 객체를 인자로 넣을 경우 그대로 반환한다.

ST_MULTI의 세부 내용은 다음과 같다.

- 문법

st_multi



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_MULTI 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_MULTI(GEOM))
--------------------------------------------------------------------------------
MULTILINESTRING((1 1,2 2))
```

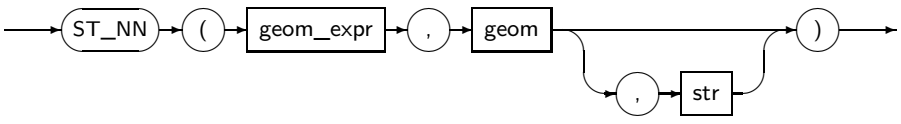
4.71. ST_NN($)

ST_NN은 조건절에 사용되는 함수로, TABLE의 GEOMETRY 컬럼에 대해 입력으로 주어진 GEOMETRY와 가장 가까운 GEOMETRY가 포함된 행들에 대해 1을 반환하는 함수이다. 거리가 짧은순으로 행들을 반환한다.

ST_NN의 세부 내용은 다음과 같다.

- 문법

st_nn



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다. RTREE가 빌드되어 있는 GEOMETRY TYPE의 컬럼이어야 한다.
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
str	ST_NN에 대해 1을 반환할 행의 개수 등의 추가 정보를 주어진 양식에 맞도록 입력한다. WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다. str 양식은 총 3가지로 구성되며 각각의 인자들은 콤마(,)로 구분하여 사용한다. – res_num : GEOMETRY 컬럼들 가까운 순으로 선별할 GEOMETRY의 개수를 입력한다. 입력하지 않을 경우 전체 컬럼에 대해 전부 1을 반환한다. – batch_size : RTREE에서 탐색할 GEOMETRY의 개수 단위를 입력한다. 사용자가 특정 값에 대해 효율적인 연산이 가능하다고 판단한 값을 입력한다. 입력하지 않을 경우 시스템에서 설정한 기본값으로 동작한다. – distance : 병목으로 설정할 거리를 입력한다. 입력된 거리 이하에 있는 행들을 반환하게 된다. – unit : 병목으로 설정할 거리의 단위를 입력한다.

- 예제

  다음은 ST_NN 함수를 사용하는 예이다.

```
SQL> SELECT ID FROM GIS WHERE ST_NN(GEOM,
     ST_GEOMFROMTEXT('POINT(0 0)'),'RES_NUM=3,BATCH_SIZE=10')=1;

        ID
----------
       106
       101
       102
```

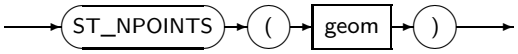
4.72. ST_NPOINTS

ST_NPOINTS는 GEOMETRY가 가지고 있는 POINT의 개수를 반환한다.

ST_NPOINTS의 세부 내용은 다음과 같다.

- 문법

st_npoints



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_NPOINTS 함수를 사용하는 예이다.

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

ST_NPOINTS(GEOM)
----------------
               1
               2
               2
               4
               5
              15
              10
               3
               0
               3
```

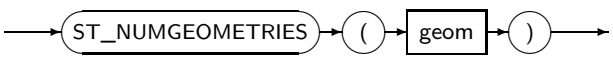
4.73. ST_NUMGEOMETRIES

ST_NUMGEOMETRIES는 GEOMETRYCOLLECTION, MULTIPOING, MULTILINESTRING, MULTIPOLYGON 객체에서 GEOMETRY 객체 수를 반환하는 함수이다. MULTI 타입이 아닌 타입의 GEOMETRY 객체를 넣을 경우 1을 반환한다.

ST_NUMGEOMETRIES의 세부 내용은 다음과 같다.

- 문법

st_numgeometries



- 구성요소

구성요소	설명
geom	COLLECTION 객체를 나타내는 GEOMETRY 타입이어야 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))

                      0
GEOMETRYCOLLECTION EMPTY

                      2
GEOMETRYCOLLECTION(POINT(1 1),LINESTRING(2 2,3 3))
```

4.74. ST_NUMINTERIORRING

ST_NUMINTERIORRING은 POLYGON 객체의 내부 링 개수를 반환하는 함수이다. POLYGON이 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_NUMINTERIORRING의 세부 내용은 다음과 같다.

- 문법

  st_numinteriorring

- 구성요소

구성요소	설명
geom	POLYGON 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

  다음은 ST_NUMINTERIORRING 함수를 사용하는 예이다.

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

ST_NUMINTERIORRING(GEOM)
------------------------
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 10,6 10),(6 3,6 6,9 6,9 3,6
3))
```

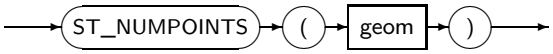
4.75. ST_NUMPOINTS

ST_NUMPOINTS는 GEOMETRY가 가지고 있는 POINT의 개수를 반환한다. ST_NPOINTS와 같은 사양의 함수이다.

ST_NUMPOINTS의 세부 내용은 다음과 같다.

- 문법

  st_numpoints

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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)
```

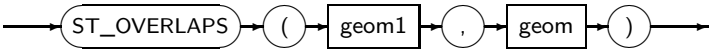
4.76. ST_OVERLAPS

ST_OVERLAPS는 주어진 두 GEOMETRY 객체가 동일한 차원이고, 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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'LINESTRING'
     AND  ST_GEOMETRYTYPE(B.GEOM) LIKE 'MULTILINESTRING';

ST_OVERLAPS(A.GEOM,B.GEOM)
--------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                         0
LINESTRING(1 1,2 2)
MULTILINESTRING((1 1,2 2),(3 3,4 5))
```

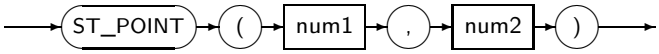
4.77. ST_POINT

ST_POINT는 2D POINT 객체를 반환하는 함수이다.

ST_POINT의 세부 내용은 다음과 같다.

- 문법

st_point



- 구성요소

구성요소	설명
num1	POINT의 x 좌표를 나타낸다.
num2	POINT의 y 좌표를 나타낸다.

- 예제

다음은 ST_POINT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_POINT(1,2)) FROM DUAL;

ST_ASTEXT(ST_POINT(1,2))
--------------------------------------------------------------------------------
POINT(1 2)
```

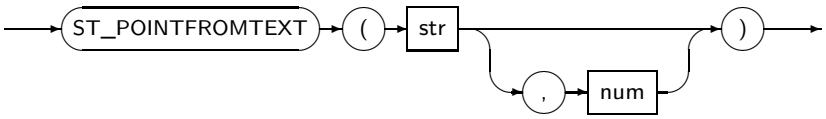
4.78. ST_POINTFROMTEXT

ST_POINTFROMTEXT는 주어진 WKT와 SRID를 바탕으로 POINT 객체로 반환하는 함수이다.

ST_POINTFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_pointfromtext



- 구성요소

구성요소	설명
str	POINT 객체을 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	POINT 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_POINTFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_POINTFROMTEXT('POINT(1010)'))
--------------------------------------------------------------------------------
POINT(10 10)
```

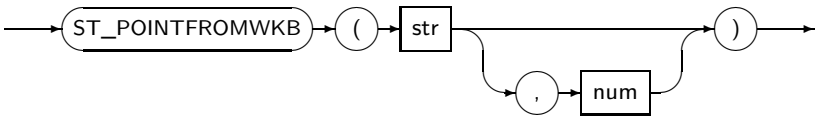
4.79. ST_POINTFROMWKB

ST_POINTFROMWKB는 주어진 WKB와 SRID를 바탕으로 POINT 객체로 반환하는 함수이다.

ST_POINTFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_pointfromwkb



- 구성요소

구성요소	설명
str	POINT 객체을 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	POINT 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_POINTFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_POINTFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
POINT(1 1)
```

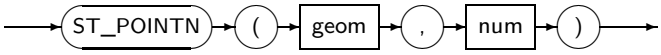
4.80. ST_POINTN

ST_POINTN은 LINESTRING 객체의 N번째 점을 POINT 객체로 반환하는 함수이다. LINESTRING이 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다. -1을 LINESTRING 객체의 마지막 POINT로 하여 반대 방향으로 셈하여 해당 음수 인자에 맞는 POINT 객체를 반환한다.

ST_POINTN의 세부 내용은 다음과 같다.

- 문법

st_pointn



- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 한다.
num	N번째 점을 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)
```

4.81. ST_POINTONSURFACE

ST_POINTONSURFACE는 해당 공간 객체 위에 놓이는 것이 보장되는 임의의 POINT 객체를 반환한다.

ST_POINTONSURFACE의 세부 내용은 다음과 같다.

- 문법

st_pointonsurface

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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_ASTEXT(ST_POINTONSURFACE(GEOM))
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
POINT(1.5 1.5)
POLYGON((1 1,2 1,2 2,1 2,1 1))

POINT(6 8)
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))
```

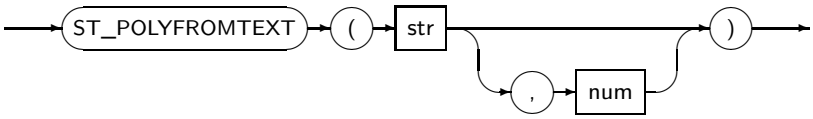
4.82. ST_POLYFROMTEXT

ST_POLYFROMTEXT는 주어진 WKT와 SRID를 바탕으로 POLYGON 객체로 반환하는 함수이다.

ST_POLYFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_polyfromtext



- 구성요소

구성요소	설명
str	GEOMETRY POLYGON 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	POLYGON 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_POLYFROMTEXT 함수를 사용하는 예이다.

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

```
ST_ASTEXT(ST_POLYFROMTEXT('POLYGON((11,21,22,12,11))'))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
```

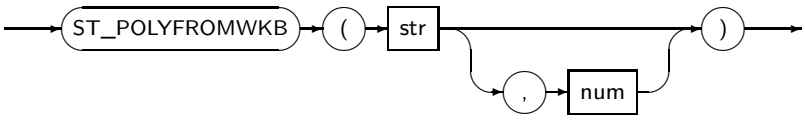
4.83. ST_POLYFROMWKB

ST_POLYFROMWKB는 주어진 WKB와 SRID를 바탕으로 POLYGON 객체로 반환하는 함수이다.

ST_POLYFROMWKB의 세부 내용은 다음과 같다.

- 문법

st_polyfromwkb



- 구성요소

구성요소	설명
str	POLYGON 객체를 표현하기 위한 WKB 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	POLYGON 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

다음은 ST_POLYFROMWKB 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_POLYFROMWKB(ST_ASBINARY(GEOM)))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
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))
```

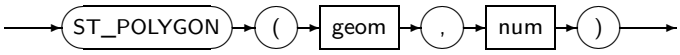
4.84. ST_POLYGON

ST_POLYGON는 주어진 LINESTRING과 SRID로 형성된 POLYGON을 반환한다. 입력 도형이 닫힌 LINESTRING이어야 한다.

ST_POLYGON의 세부 내용은 다음과 같다.

- 문법

st_polygon



- 구성요소

구성요소	설명
geom	형성할 POLYGON의 외각 껍데기를 나타내는 LINESTRING 객체의 GEOMETRY 타입이어야 한다. 내부 구멍을 포함하고있는 POLYGON 도형을 형성하려면 ST_MAKEPOLYGON을 사용한다.
num	형성할 POLYGON의 좌표계 정보(SRID)를 입력한다.

- 예제

다음은 ST_POLYGON 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_POLYGON(ST_GEOMFROMTEXT(
     'LINESTRING(0 0,1 0,1 1,0 1,0 0)'), 4326)) FROM DUAL;

ST_ASTEXT(ST_POLYGON(ST_GEOMFROMTEXT('LINESTRING(00,10,11,01,00)'),4326))
--------------------------------------------------------------------------------
POLYGON((0 0,1 0,1 1,0 1,0 0))
```

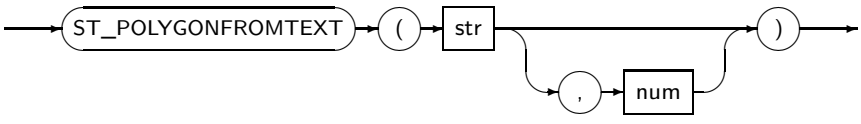
4.85. ST_POLYGONFROMTEXT

ST_POLYGONFROMTEXT는 주어진 WKT와 SRID를 바탕으로 POLYGON 객체로 반환하는 함수이다.

ST_POLYGONFROMTEXT의 세부 내용은 다음과 같다.

- 문법

st_polygonfromtext



- 구성요소

구성요소	설명
str	POLYGON 객체를 표현하기 위한 WKT 형식으로 된 CHAR, VARCHAR, NCHAR, NVARCHAR 타입 중 하나이다.
num	POLYGON 객체의 좌표계 정보(SRID)를 입력한다. SRID를 입력하지 않을 경우 0(시스템 기본 좌표계)이 설정된다.

- 예제

  다음은 ST_POLYGONFROMTEXT 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_POLYGONFROMTEXT('POLYGON((11,21,22,12,11))'))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
```

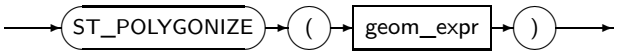
4.86. ST_POLYGONIZE(#)

ST_POLYGONIZE은 GEOMETRY 집합들의 선분 구성요소로부터 형성될 수 있는 POLYGON들을 반환하는 집합 함수이다. 집합 함수이므로 GEOMETRY 객체들을 한데 모은 집합에 대한 결과를 반환한다.

ST_POLYGONIZE의 세부 내용은 다음과 같다.

- 문법

  st_polygonize



- 구성요소

구성요소	설명
geom_expr	GEOMETRY 객체를 나타내는 임의의 연산식이다.

- 예제

  다음은 ST_POLYGONIZE 함수를 사용하는 예이다.

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

ST_ASTEXT(ST_POLYGONIZE(GEOM))
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,9 10,9 11,6 11,6 10),
```

```
(6 3,9 3,9 6,6 6,6 3),(3 3,5 3,5 5,3 5,3 3)),POLYGON((6 10,6 11,9 11,9 10,6 10))
,POLYGON((6 3,6 6,9 6,9 3,6 3)),POLYGON((3 3,3 5,5 5,5 3,3 3)))
```

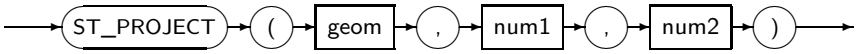
4.87. ST_PROJECT(#)

ST_PROJECT은 POINT 객체에서부터 주어진 미터 단위 거리와 호도 단위 방향(방위각)으로 투영된 POINT 객체를 반환한다. 호도의 방향이 방위각을 기준으로 설정되므로 동쪽은 $\pi/2$, 남쪽은 π, 서쪽은 $3\pi/2$로 표현된다. 회전체 좌표계 상의 GEOMETRY만을 인자로 받으며, 그렇지 않은 경우 예외를 반환한다. POINT가 아닌 타입의 GEOMETRY 객체를 넣을 경우 예외를 반환한다.

ST_PROJECT의 세부 내용은 다음과 같다.

- 문법

st_project



- 구성요소

구성요소	설명
geom	POINT 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	미터 단위에 해당하는 거리를 입력한다.
num2	호도 단위에 해당하는 방위각을 입력한다.

- 예제

다음은 ST_PROJECT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_PROJECT(
     ST_GEOMFROMTEXT('POINT(0 0)',4326),100000, 3.141592)) FROM DUAL;

ST_ASTEXT(ST_PROJECT(ST_GEOMFROMTEXT('POINT(00)',4326),100000,3.141592))
--------------------------------------------------------------------------------
POINT(5.87178138550817e-07 -0.90436872291257)
```

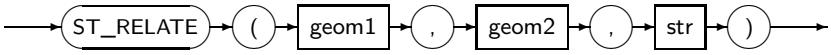
4.88. ST_RELATE

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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
str	두 객체의 관계를 의미하는 문자열를 타나내는 CHAR, VARCHAR, 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) LIKE 'LINESTRING'
     AND  ST_GEOMETRYTYPE(B.GEOM) LIKE 'LINESTRING';

ST_RELATE(A.GEOM,B.GEOM,'TTTTTTTTT')
------------------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                                     0
LINESTRING(1 1,2 2)
LINESTRING(1 1,2 2)
```

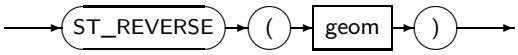
4.89. ST_REVERSE

ST_REVERSE는 주어진 GEOMETRY 객체의 꼭지점의 순서가 반대인 GEOMETRY를 반환하는 함수이다.

ST_REVERSE의 세부 내용은 다음과 같다.

- 문법

st_reverse



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_REVERSE 함수를 사용하는 예이다.

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

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

4.90. ST_SETSRID

ST_SETSRID는 주어진 GEOMETRY 객체의 SRID를 설정하여 반환한다.

ST_SETSRID의 세부 내용은 다음과 같다.

- 문법

st_setsrid

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

구성요소	설명
num	설정할 좌표계 정보(SRID)를 입력한다.

- 예제

  다음은 ST_SETSRID 함수를 사용하는 예이다.

```
SQL> SELECT ST_SRID(ST_SETSRID(ST_GEOMFROMTEXT('POINT(0 0)'), 4326))
     FROM DUAL;

ST_SRID(ST_SETSRID(ST_GEOMFROMTEXT('POINT(00)'),4326))
------------------------------------------------------
                                                  4326
```

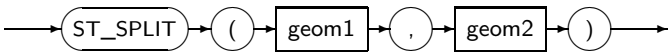
4.91. ST_SPLIT(#)

ST_SPLIT는 GEOMETRY를 기준이 되는 GEOMETRY로 나누어서 나오는 GEOMETRYCOLLECTION을 반환한다. LINESTRING은 (MULTI)POINT, (MULTI)LINESTRING, (MULTI)POLYGON에 의해 나누어질 수 있고, POLYGON은 (MULTI)LINESTRING에 의해 나누어질 수 있다.

ST_SPLIT의 세부 내용은 다음과 같다.

- 문법

  st_split



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다. 나누어질 GEOMETRY를 입력한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다. 나눌 기준이 되는 GEOMETRY를 입력한다.

- 예제

  다음은 ST_SPLIT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_SPLIT(ST_GEOMFROMTEXT(
     'LINESTRING(0 0,2 2)'),ST_GEOMFROMTEXT('POINT(1 1)'))) SPLIT FROM DUAL;

SPLIT
```

```
--------------------------------------------------------------------------------
GEOMETRYCOLLECTION(LINESTRING(0 0,1 1),LINESTRING(1 1,2 2))
```

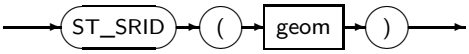
4.92. ST_SRID

ST_SRID는 주어진 GEOMETRY 객체의 SRID를 반환하는 함수이다.

ST_SRID의 세부 내용은 다음과 같다.

- 문법

st_srid



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

- 예제

다음은 ST_SRID 함수를 사용하는 예이다.

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

        ID ST_SRID(GEOM)
---------- -------------
       101             0
       102             0
       103             0
       104             0
       105             0
       106             0
       107             0
       108             0
       109             0
       110             0
```

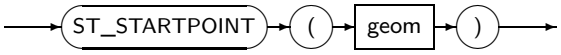
4.93. ST_STARTPOINT

ST_STARTPOINT는 LINESTRING 첫 POINT를 반환하는 함수이다. LINESTRING이 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_STARTPOINT의 세부 내용은 다음과 같다.

- 문법

st_startpoint



- 구성요소

구성요소	설명
geom	LINESTRING 객체를 나타내는 GEOMETRY 타입이어야 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)
```

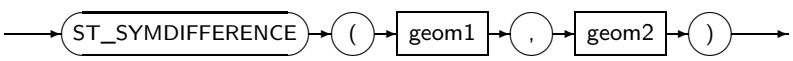
4.94. ST_SYMDIFFERENCE($)

ST_SYMDIFFERENCE는 주어진 두 GEOMETRY 객체의 INTERSECTION을 제외한 영역을 나타내는 GEOMETRY 객체를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다.

ST_SYMDIFFERENCE의 세부 내용은 다음과 같다.

- 문법

st_symdifference



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.

구성요소	설명
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POLYGON'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(ST_SYMDIFFERENCE(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))
GEOMETRYCOLLECTION EMPTY

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))
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,9 10,9 11,6 11,6 10),(6 3,9 3,9 6,6 6,6
3),(1 1,2 1,2 2,1 2,1 1))

POLYGON((1 1,2 1,2 2,1 2,1 1))
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))
POLYGON((0 0,0 12,12 12,12 0,0 0),(1 1,2 1,2 2,1 2,1 1),(6 10,9 10,9 11,6 11,6 1
0),(6 3,9 3,9 6,6 6,6 3))

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))
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))
GEOMETRYCOLLECTION EMPTY
```

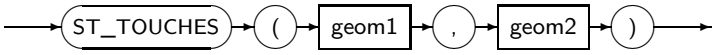
4.95. ST_TOUCHES

ST_TOUCHES는 주어진 두 GEOMETRY 객체가 한 점 이상을 공유하지만 교차영역이 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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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 WHERE ST_GEOMETRYTYPE(A.GEOM) LIKE 'LINESTRING'
     AND  ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_TOUCHES(A.GEOM,B.GEOM)
-------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                         0
LINESTRING(1 1,2 2)
POLYGON((1 1,2 1,2 2,1 2,1 1))

                         0
LINESTRING(1 1,2 2)
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))
```

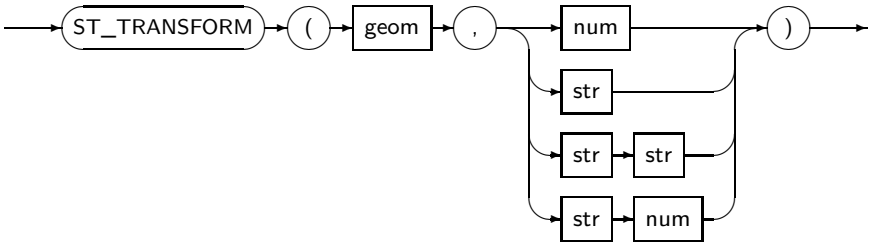
4.96. ST_TRANSFORM(#)

ST_TRANSFORM는 입력 GEOMETRY의 좌표를 다른 공간 참조 시스템(Spatial reference)으로 변환시킨 GEOMETRY를 반환한다. 인자로 SRID를 넣거나 PROJ4TEXT를 받는다. ST_TRANSFORM은 좌표를 변환하는 작업을 수행하지만 ST_SETSRID는 좌표를 변환하지 않고 SRID만 설정한다.

ST_TRANSFORM의 세부 내용은 다음과 같다.

- 문법

st_transform



- 구성요소

  – case1

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num	변환될 좌표계 정보(SRID)를 입력한다.

  – case2

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
str	변환될 좌표계 정보의 PROJ4TEXT를 입력한다.

  – case3

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
str	변환되기전 좌표계 정보의 PROJ4TEXT를 입력한다.
str	변환할 좌표계 정보의 PROJ4TEXT를 입력한다.

  – case4

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
str	변환되기전 좌표계 정보의 PROJ4TEXT를 입력한다.
num	변환할 좌표계 정보(SRID)를 입력한다.

- 예제

  다음은 ST_TRANSFORM 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_TRANSFORM(ST_GEOMFROMTEXT(
     'LINESTRING(0 0,1 1)',4326),2249)) FROM DUAL;

ST_ASTEXT(ST_TRANSFORM(ST_GEOMFROMTEXT('LINESTRING(00,11)',4326),2249))
--------------------------------------------------------------------------------
LINESTRING(30250865.9714116 -610981.481754334,30214669.4455592 41989.5268075016)
```

```
SQL> SELECT ST_ASTEXT(ST_TRANSFORM(ST_GEOMFROMTEXT('LINESTRING(0 0,1 1)'),
     '+proj=longlat +ellps=WGS72 +towgs84=0,0,1.9,0,0,0.814,-0.38 +no_defs',2249))

     TRANSFORM FROM DUAL;

TRANSFORM
--------------------------------------------------------------------------------
LINESTRING(30250930.6174208 -610897.702286165,30214732.3446824 42073.100013002)
```

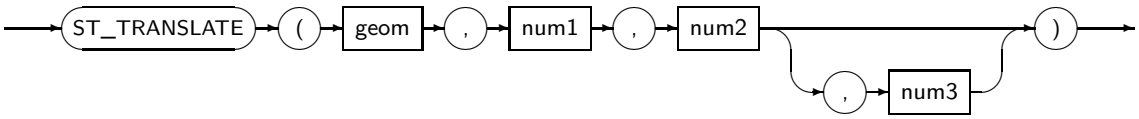
4.97. ST_TRANSLATE(#)

ST_TRANSLATE는 주어진 오프셋만큼 이동한 GEOMETRY를 반환한다.

ST_TRANSLATE의 세부 내용은 다음과 같다.

- 문법

st_translate



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
num1	X 방향으로 이동시킬 오프셋이다.
num2	Y 방향으로 이동시킬 오프셋이다.
num3	Z 방향으로 이동시킬 오프셋이다.

- 예제

  다음은 ST_TRANSLATE 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(ST_TRANSLATE(
     ST_GEOMFROMTEXT('LINESTRING(0 0,1 0)'),1,2)) FROM DUAL;

ST_ASTEXT(ST_TRANSLATE(ST_GEOMFROMTEXT('LINESTRING(00,10)'),1,2))
--------------------------------------------------------------------------------
LINESTRING(1 2,2 2)
```

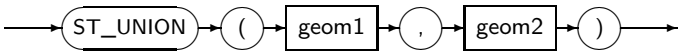
4.98. ST_UNION($)

ST_UNION은 주어진 두 GEOMETRY 객체의 영역을 합한 GEOMETRY 객체를 반환하는 함수이다. 좌표계 정보를 통해 회전체 좌표계 상의 GEOMETRY 객체에 대한 연산이 가능하다.

ST_UNION의 세부 내용은 다음과 같다.

- 문법

st_union



- 구성요소

구성요소	설명
geom1	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POLYGON'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(ST_SYMDIFFERENCE(A.GEOM,B.GEOM))
--------------------------------------------------------------------------------
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))
GEOMETRYCOLLECTION EMPTY
```

```
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))
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((0 0,0 12,12 12,12 0,0 0),(6 10,9 10,9 11,6 11,6 10),(6 3,9 3,9 6,6 6,6
3),(1 1,2 1,2 2,1 2,1 1))

POLYGON((1 1,2 1,2 2,1 2,1 1))
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))
POLYGON((0 0,0 12,12 12,12 0,0 0),(1 1,2 1,2 2,1 2,1 1),(6 10,9 10,9 11,6 11,6 1
0),(6 3,9 3,9 6,6 6,6 3))

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))
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))
GEOMETRYCOLLECTION EMPTY
```

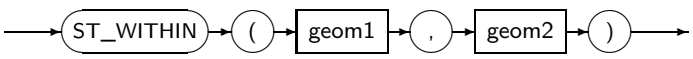
4.99. ST_WITHIN

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 타입이어야 한다.
geom2	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야 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) LIKE 'POLYGON'
     AND ST_GEOMETRYTYPE(B.GEOM) LIKE 'POLYGON';

ST_WITHIN(A.GEOM,B.GEOM)
```

```
------------------------
ST_ASTEXT(A.GEOM)
--------------------------------------------------------------------------------
ST_ASTEXT(B.GEOM)
--------------------------------------------------------------------------------
                       1
POLYGON((1 1,2 1,2 2,1 2,1 1))
POLYGON((1 1,2 1,2 2,1 2,1 1))

                       0
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))
POLYGON((1 1,2 1,2 2,1 2,1 1))

                       1
POLYGON((1 1,2 1,2 2,1 2,1 1))
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))

                       1
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))
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))
```

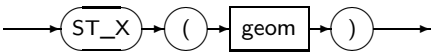
4.100. ST_X

ST_X는 POINT 객체의 X 좌표를 반환하는 함수이다. POINT가 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_X의 세부 내용은 다음과 같다.

- 문법

st_x



- 구성요소

구성요소	설명
point	POINT 객체를 나타내는 GEOMETRY 타입이어야 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
```

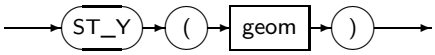
4.101. ST_Y

ST_Y는 POINT 객체의 Y 좌표를 반환하는 함수이다. POINT가 아닌 타입의 GEOMETRY 객체를 넣을 경우 NULL을 반환한다.

ST_Y의 세부 내용은 다음과 같다.

- 문법

  st_y

- 구성요소

구성요소	설명
point	POINT 객체를 나타내는 GEOMETRY 타입이어야 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
```

4.102. DBMS_GEOM 패키지

Tibero Spatial에서는 DBMS_GEOM 패키지를 통해 추가적인 함수를 지원한다. Linear Referencing System(LRS)관련 함수들도 해당 패키지에서 지원한다.

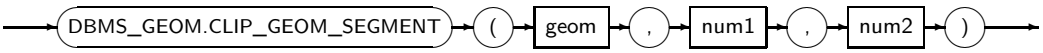
4.102.1. CLIP_GEOM_SEGMENT

CLIP_GEOM_SEGMENT는 주어진 LRS GEOMETRY 객체를 입력받은 시작 MEASURE 값과 끝 MEASURE 값으로 자른 LRS GEOMETRY 객체를 반환하는 함수이다.

CLIP_GEOM_SEGMENT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.clip_geom_segment



- 구성요소

구성요소	설명
geom	시작과 끝 MEASURE를 계산할 기준이 되는 GEOMETRY 객체이다.
num1	반환될 GEOMETRY 객체의 시작 MEASURE 값이다.
num2	반환된 GEOMETRY 객체의 끝 MEASURE 값이다.

- 예제

다음은 DBMS_GEOM.CLIP_GEOM_SEGMENT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.CLIP_GEOM_SEGMENT(ST_GEOMFROMTEXT(
     'LINESTRINGM(0 0 0,3 0 30,3 5 130)'), 10, 80)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.CLIP_GEOM_SEGMENT(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,3513
--------------------------------------------------------------------------------
LINESTRINGM(1 0 10,3 0 30,3 2.5 80)
```

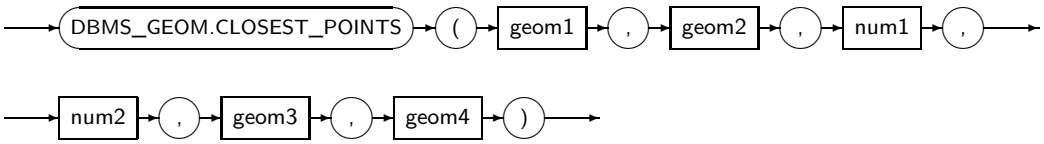
4.102.2. CLOSEST_POINT

CLOSEST_POINT는 두 GEOMETRY 객체에서 서로 가장 가까운 두 POINT 타입의 GEOMETRY 객체와 그 거리를 반환하는 프로시저이다.

CLOSEST_POINT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.closest_points



- 구성요소

구성요소	설명
geom1	geom2와 가장 가까운 위치를 구할 GEOMETRY 객체이다.
geom2	geom1과 가장 가까운 위치를 구할 GEOMETRY 객체이다.
num1	precision의 정도를 결정할 tolerance이다.
num2	두 GEOMETRY 객체간의 거리를 입력받을 NUMBER 변수이다.
geom3	geom1에서 geom2와 가장 가까운 위치를 받을 POINT 타입의 GEOMETRY 객체이다.
geom4	geom2에서 geom1과 가장 가까운 위치를 받을 POINT 타입의 GEOMETRY 객체이다.

- 예제

  다음은 DBMS_GEOM.CLOSEST_POINTS 프로시저를 사용하는 예이다.

```
SQL> set serveroutput on;
SQL> DECLARE
     2     dist NUMBER;
     3     geoma geometry;
     4     geomb geometry;
     5 BEGIN
     6     dbms_geom.closest_points(
     7         st_geomfromtext('linestring(0 0,3 0)'),
     8         st_geomfromtext('linestring(1 1,1 3)'),
     9         0.05,
    10         dist,
    11         geoma,
    12         geomb
    13     );
    14     dbms_output.put_line('dist = ' || dist);
    15     dbms_output.put_line(st_astext(geoma));
    16     dbms_output.put_line(st_astext(geomb));
    17 END;
    18 /
dist = 1
POINT(1 0)
POINT(1 1)
```

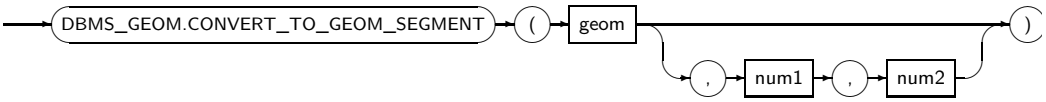
4.102.3. CONVERT_TO_GEOM_SEGMENT

CONVERT_TO_GEOM_SEGMENT은 MEASURE가 없는 기존의 GEOMETRY에 MEASURE를 계산한 LRS GEOMETRY를 반환하는 함수이다.

CONVERT_TO_GEOM_SEGMENT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.convert_to_geom_segment



- 구성요소

구성요소	설명
geom	LRS GEOMETRY를 계산하고자하는 GEOMETRY 객체이다.
num1	반환할 LRS GEOMETRY의 시작 MEASURE 값을 입력한다. 기본으로 0이 설정되어있다.
num2	반환할 LRS GEOMETRY의 끝 MEASURE 값을 입력한다. 기본으로 GEOMETRY의 길이값이 설정되어있다.

- 예제

다음은 DBMS_GEOM.CONVERT_TO_GEOM_SEGMENT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.CONVERT_TO_GEOM_SEGMENT(ST_GEOMFROMTEXT(
     'LINESTRING(0 0,3 0,3 5)'))) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.CONVERT_TO_GEOM_SEGMENT(ST_GEOMFROMTEXT('LINESTRING(00,30,35
--------------------------------------------------------------------------------
LINESTRINGM(0 0 0,3 0 3,3 5 8)

1 row selected.
```

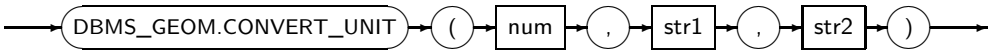
4.102.4. CONVERT_UNIT

CONVERT_UNIT은 UNITS_OF_MEASURE 테이블에 명시되어있는 단위계 정보를 이용해 단위를 변환하는 함수이다.

CONVERT_UNIT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.convert_unit



- 구성요소

구성요소	설명
num	변환하고자하는 단위계의 값이다.
str1	변환하기 전의 단위계의 이름이다.
str2	변환할 단위계의 이름이다.

- 참조

  UNITS_OF_MEASURE

- 예제

  다음은 DBMS_GEOM.CONVERT_UNIT 함수를 사용하는 예이다.

```
SQL> SELECT DBMS_GEOM.CONVERT_UNIT(20,'M','KM') FROM DUAL;

DBMS_GEOM.CONVERT_UNIT(20,'M','KM')
-----------------------------------
                                .02

1 row selected.
```

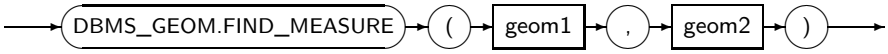
4.102.5. FIND_MEASURE

FIND_MEASURE는 LRS타입의 GEOMETRY 객체에서 특정 위치의 MEASURE 값을 반환하는 함수이다. MEASURE 값을 반환할 위치는 본 함수의 두번째 인자인 POINT 타입의 GEOMETRY 객체를 통해 나타낸다. 이 GEOMETRY 객체에서 가장 가까운 GEOM1의 위치에 해당하는 MEASURE 값을 반환한다.

FIND_MEASURE의 세부 내용은 다음과 같다.

- 문법

  dbms_geom.find_measure



- 구성요소

구성요소	설명
geom1	MEASURE 값을 구할 GEOMETRY 객체이다.
geom2	MEASURE를 구할 위치를 투영시킬 POINT 타입의 GEOMETRY 객체이다.

- 예제

  다음은 DBMS_GEOM.FIND_MEASURE 함수를 사용하는 예이다.

```
SQL> SELECT DBMS_GEOM.FIND_MEASURE(ST_GEOMFROMTEXT(
     'LINESTRINGM(0 0 0,3 0 30,3 5 80)'),ST_GEOMFROMTEXT('POINTM(2 4 1)'))
     FROM DUAL;

DBMS_GEOM.FIND_MEASURE(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,3580)'),ST_GEOMFROM
--------------------------------------------------------------------------------
                                                                              70


1 row selected.
```

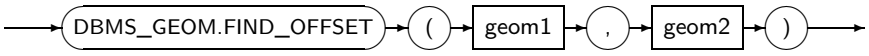
4.102.6. FIND_OFFSET

FIND_OFFSET는 LRS타입의 GEOMETRY 객체와 POINT 타입의 LRS GEOMETRY 객체 사이의 OFFSET을 반환하는 함수이다.

FIND_OFFSET의 세부 내용은 다음과 같다.

- 문법

  dbms_geom.find_offset



- 구성요소

구성요소	설명
geom1	OFFSET을 구할 기준이 되는 GEOMETRY 객체이다.
geom2	기준 GEOMETRY로부터 OFFSET을 계산할 위치에 놓인 POINT 타입의 GEOMETRY 객체이다.

- 예제

  다음은 DBMS_GEOM.FIND_OFFSET 함수를 사용하는 예이다.

```
SQL> SELECT DBMS_GEOM.FIND_OFFSET(ST_GEOMFROMTEXT(
     'LINESTRINGM(0 0 0,3 0 30,8 0 130)'), ST_GEOMFROMTEXT('POINTM(5 3 100)'))
```

```
     FROM DUAL;

DBMS_GEOM.FIND_OFFSET(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,80130)'),ST_GEOMFROM
--------------------------------------------------------------------------------
                                                                               3

1 row selected.
```

4.102.7. FROM_WGS84

FROM_WGS84는 wgs84형식의 위도, 경도 기준으로 표현된 GEOMETRY 타입의 객체를 주어진 점을 기준으로 한 가로, 세로 축으로 표현된 GEOMETRY 객체로 변환하는 함수이다. 이 때, GEOMETRY 객체의 좌표를 해석할 때는 경도, 위도 순서이지만, FROM_WGS84에서 변환할 기준점은 위도, 경도 순서로 인자를 받게된다.

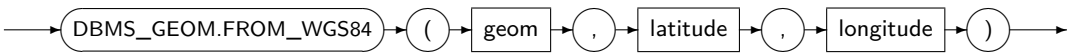
참고

FROM_WGS84 함수와 TO_WGS84 함수는 부동소숫점 연산을 반복하여 사용한다. 따라서, GEOMETRY 타입의 객체가 가지고 있는 좌표 정보의 왜곡이 발생할 수 있다. 또한, 부동소숫점 연산 방식에 따라 실행 환경별 출력 결과의 차이가 있다.

FROM_WGS84의 세부 내용은 다음과 같다.

- 문법

dbms_geom.from_wgs84



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야한다.
latitude	변환할 때 기준이 될 NUMBER 타입의 위도값이다.
longitude	변환할 때 기준이 될 NUMBER 타입의 경도값이다.

- 예제

다음은 DBMS_GEOM.FROM_WGS84 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT
     ('POINT(0 1)'), 0, 0)) FROM DUAL;
```

```
ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT('POINT(01)'),0,0))
--------------------------------------------------------------------------------
POINT(0 110.94633302763)

1 row selected.

SQL> SELECT ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT
     ('POINT(0 1)'), 15, 30)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT('POINT(01)'),15,30))
--------------------------------------------------------------------------------
POINT(-3331.5555052183 -1553.7714201354)

1 row selected.

SQL> SELECT ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT
      ('POINT(30 40)'), 40, 30)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.FROM_WGS84(ST_GEOMFROMTEXT('POINT(3040)'),40,30))
--------------------------------------------------------------------------------
POINT(-0 0)

1 row selected.
```

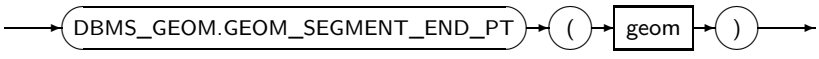
4.102.8. GEOM_SEGMENT_END_PT

GEOM_SEGMENT_END_PT는 GEOMETRY 객체에서 마지막 좌표에 해당하는 POINT 타입의 GEOMETRY를 반환하는 함수이다.

GEOM_SEGMENT_END_PT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.geom_segment_end_pt



- 구성요소

구성요소	설명
geom	마지막 좌표를 반환할 GEOMETRY 객체이다.

- 예제

다음은 DBMS_GEOM.GEOM_SEGMENT_END_PT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.GEOM_SEGMENT_END_PT(
     ST_GEOMFROMTEXT('LINESTRINGM(0 0 0,1 2 3)'))) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.GEOM_SEGMENT_END_PT(ST_GEOMFROMTEXT('LINESTRINGM(000,123)'))
--------------------------------------------------------------------------------
POINTM(1 2 3)

1 row selected.
```

4.102.9. GEOM_SEGMENT_START_PT

GEOM_SEGMENT_START_PT는 GEOMETRY 객체에서 첫번째 좌표에 해당하는 POINT 타입의 GEOMETRY를 반환하는 함수이다.

GEOM_SEGMENT_START_PT의 세부 내용은 다음과 같다.

- 문법

  dbms_geom.geom_segment_start_pt

- 구성요소

구성요소	설명
geom	첫번째 좌표를 반환할 GEOMETRY 객체이다.

- 예제

  다음은 DBMS_GEOM.GEOM_SEGMENT_START_PT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.GEOM_SEGMENT_START_PT(
     ST_GEOMFROMTEXT('LINESTRINGM(0 0 0,1 2 3)'))) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.GEOM_SEGMENT_START_PT(ST_GEOMFROMTEXT('LINESTRINGM(000,123)'
--------------------------------------------------------------------------------
POINTM(0 0 0)

1 row selected.
```

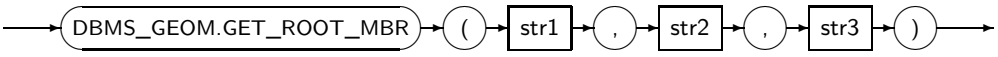
4.102.10. GET_ROOT_MBR

GET_ROOT_MBR는 GET_ROOT_MBR 함수는 공간 인덱스의 루트 노드에 해당하는 최소 경계 사각형을 POLYGON으로 반환하는 함수이다.

GET_ROOT_MBR의 세부 내용은 다음과 같다.

- 문법

dbms_geom.get_root_mbr



- 구성요소

구성요소	설명
str1	스키마 객체의 이름이다.
str2	테이블 객체의 이름이다.
str3	공간 인덱스 객체의 이름이다.

- 예제

다음은 DBMS_GEOM.GET_ROOT_MBR 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.GET_ROOT_MBR('TIBERO','GIS','RT_IDX_GIS'))
     FROM DUAL;

ST_ASTEXT(DBMS_GEOM.GET_ROOT_MBR('TIBERO','GIS','RT_IDX_GIS'))
--------------------------------------------------------------------------------
POLYGON((0 0,0 12,12 12,12 0,0 0))

1 row selected.
```

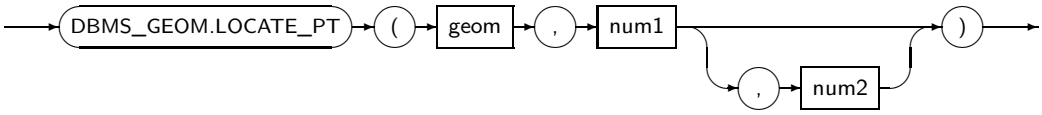
4.102.11. LOCATE_PT

LOCATE_PT는 주어진 LRS GEOMETRY 객체의 특정 MEASURE 값을 가지는 POINT 타입의 LRS GEOMETRY 객체를 반환하는 함수이다.

LOCATE_PT의 세부 내용은 다음과 같다.

- 문법

dbms_geom.locate_pt



- 구성요소

구성요소	설명
geom	MEASURE의 위치를 찾는 기준이 되는 GEOMETRY 객체이다.
num1	위치를 찾고자하는 MEASURE 값이다.
num2	MEASURE에 해당하는 좌표와 기준 GEOMETRY와의 OFFSET 값이다. 기본으로 0이 설정되어 있다. (기본으로 MEASURE에 해당하는 POINT가 기준 GEOMETRY 위에 있다.)

- 예제

  다음은 DBMS_GEOM.LOCATE_PT 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.LOCATE_PT(
     ST_GEOMFROMTEXT('LINESTRINGM(0 0 0,3 0 30,3 5 130)'), 20)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.LOCATE_PT(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,35130)'),20)
--------------------------------------------------------------------------------
POINTM(2 0 20)

1 row selected.

SQL> SELECT ST_ASTEXT(DBMS_GEOM.LOCATE_PT(ST_GEOMFROMTEXT(
     'LINESTRINGM(0 0 0,3 0 30,3 5 130)'), 50, 10)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.LOCATE_PT(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,35130)'),50,
--------------------------------------------------------------------------------
POINTM(-7 1 50)

1 row selected.
```

4.102.12. MEASURE_RANGE

MEASURE_RANGE는 LRS GEOMETRY 객체의 끝 MEASURE와 시작 MEASURE간의 차이를 반환하는 함수이다.

MEASURE_RANGE의 세부 내용은 다음과 같다.

- 문법

dbms_geom.measure_range

- 구성요소

구성요소	설명
geom	MEASURE의 범위를 계산할 GEOMETRY 객체이다.

- 예제

다음은 DBMS_GEOM.MEASURE_RANGE 함수를 사용하는 예이다.

```
SQL> SELECT DBMS_GEOM.MEASURE_RANGE(ST_GEOMFROMTEXT(
     'LINESTRINGM(0 0 0,3 0 30,3 5 130)')) FROM DUAL;

DBMS_GEOM.MEASURE_RANGE(ST_GEOMFROMTEXT('LINESTRINGM(000,3030,35130)'))
------------------------------------------------------------------------
                                                                    130

1 row selected.
```

4.102.13. ST_DUMPPOINTS

ST_DUMPPOINTS는 GEOMETRY 및 정수 배열(path)로 이루어진 집합을 반환하는 집합 반환함수이다.

인자로 넣어준 GEOMETRY의 모든 POINT와 그에 해당하는 알맞은 정수 배열을 맞게 설정한다. 예를 들어 LINESTRING을 입력으로 준 경우 path는 {i}로 설정되며, i는 LINESTRING의 i번째 POINT임을 의미한다. POLYGON을 입력으로 주면 path 는 {i,j}로 설정되며 i는 i번째 고리, j는 i번째 고리의 j번째 POINT를 나타낸다. MULTI 타입을 인자로 주면 각 path의 앞에 n번째 GEOMETRY임을 나타내는 정수가 추가되는 방식으로 동작한다.

ST_DUMPPOINTS의 세부 내용은 다음과 같다.

- 문법

dbms_geom.st_dumppoints

- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야한다.

- 예제

  다음은 DBMS_GEOM.ST_DUMPPOINTS 함수를 사용하는 예이다.

```
SQL>SELECT PATH, ST_ASTEXT(GEOM) FROM TABLE(
DBMS_GEOM.ST_DUMPPOINTS(ST_GEOMFROMTEXT('LINESTRING(1 1,2 2,3 3)')));

PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{1}
POINT(1 1)

{2}
POINT(2 2)

{3}
POINT(3 3)


3 rows selected.

SQL> SELECT PATH, ST_ASTEXT(GEOM) FROM TABLE(DBMS_GEOM.ST_DUMPPOINTS(
ST_GEOMFROMTEXT('POLYGON((0 0,4 0,4 4,0 4,0 0),(1 1,3 1,3 3,1 3,1 1))')));

PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{1,1}
POINT(0 0)

{1,2}
POINT(4 0)

{1,3}
POINT(4 4)

{1,4}
POINT(0 4)

{1,5}
POINT(0 0)

{2,1}
POINT(1 1)
```

```
{2,2}
POINT(3 1)


PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{2,3}
POINT(3 3)

{2,4}
POINT(1 3)

{2,5}
POINT(1 1)


10 rows selected.

SQL> SELECT PATH, ST_ASTEXT(GEOM) FROM TABLE(DBMS_GEOM.ST_DUMPPOINTS(
    ST_GEOMFROMTEXT('GEOMETRYCOLLECTION(POINT(0 0),LINESTRING(1 1,2 2),
    POLYGON((3 3,4 3,4 4,3 4,3 3)),POLYGON((1 1,1 4,4 4,4 1,1 1),
    (2 2,2 3,3 3,3 2,2 2)))')));

PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{1,1}
POINT(0 0)

{2,1}
POINT(1 1)

{2,2}
POINT(2 2)

{3,1,1}
POINT(3 3)

{3,1,2}
POINT(4 3)

{3,1,3}
POINT(4 4)
```

```
{3,1,4}
POINT(3 4)


PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{3,1,5}
POINT(3 3)

{4,1,1}
POINT(1 1)

{4,1,2}
POINT(1 4)

{4,1,3}
POINT(4 4)

{4,1,4}
POINT(4 1)

{4,1,5}
POINT(1 1)

{4,2,1}
POINT(2 2)


PATH
--------------------------------------------------------------------------------
ST_ASTEXT(GEOM)
--------------------------------------------------------------------------------
{4,2,2}
POINT(2 3)

{4,2,3}
POINT(3 3)

{4,2,4}
POINT(3 2)

{4,2,5}
POINT(2 2)
```

```
18 rows selected.
```

4.102.14. TO_WGS84

TO_WGS84는 가로, 세로 축으로 표현된 GEOMETRY 타입의 객체를 주어진 점을 기준으로 한 wgs84 형식의 위도, 경도 축으로 표현된 GEOMETRY 객체로 변환하는 함수이다. 이 때, GEOMETRY 객체의 좌표를 해석할 때는 경도, 위도 순서이지만, FROM_WGS84에서 변환할 기준점은 위도, 경도 순서로 인자를 받게된다.

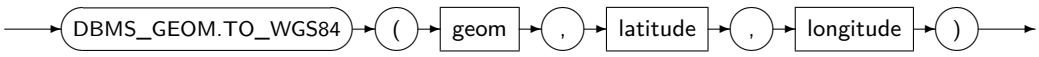
참고

FROM_WGS84 함수와 TO_WGS84 함수는 부동소숫점 연산을 반복하여 사용한다. 따라서, GEOMETRY 타입의 객체가 가지고 있는 좌표 정보의 왜곡이 발생 할 수 있다. 또한, 부동소숫점 연산 방식에 따라 실행 환경별 출력 결과의 차이가 있다.

TO_WGS84의 세부 내용은 다음과 같다.

- 문법

dbms_geom.to_wgs84



- 구성요소

구성요소	설명
geom	GEOMETRY 객체를 나타내는 GEOMETRY 타입이어야한다.
latitude	변환할 때 기준이 될 NUMBER 타입의 위도값이다.
longitude	변환할 때 기준이 될 NUMBER 타입의 경도값이다.

- 예제

다음은 DBMS_GEOM.FROM_WGS84 함수를 사용하는 예이다.

```
SQL> SELECT ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT
     ('POINT(0 110.9463302763)'), 0, 0)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT('POINT(0110.9463302763)'),0,0))
--------------------------------------------------------------------------------
POINT(0 0.99999997520128)
```

```
/* 실제 값 {POINT(0 1)}과 오차가 있다. */

1 row selected.

SQL> SELECT ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT
     ('POINT(-3331.5555052183 -1553.7714201354)'),15,30)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT('POINT(-3331.5555052183-1553.771420
--------------------------------------------------------------------------------
POINT(0 1.00000000000027)
/* 실제 값 {POINT(0 1)}과 오차가 있다. */

1 row selected.

SQL> SELECT ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT
     ('POINT(0 0)'), 40, 30)) FROM DUAL;

ST_ASTEXT(DBMS_GEOM.TO_WGS84(ST_GEOMFROMTEXT('POINT(00)'),40,30))
--------------------------------------------------------------------------------
POINT(30 40)

1 row selected.
```

제5장 Spatial 유틸리티

본 장에서는 Tibero Spatial에서 제공하는 유틸리티인 gisLoader와 tibero2shp, shp2tibero의 사용법에 대해서 기술한다.

5.1. gisLoader

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

gisLoader 유틸리티는 shapefile 형태로 되어 있는 파일의 데이터를 읽어들여서 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 Tibero에 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 생성해주지 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=[default: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 타입에서 사용될 endian 정보(big/little)를 명시한다. 생략하면 gisLoader를 사용하고 있는 기계의 endian 정보를 사용한다. endian 정보를 잘못 명시할 경우 에러가 발생하니 주의가 필요하다.
multipolygon	.shp 파일의 polygon, multipolygon 타입을 모두 multipolygon 타입으로 변환한다. 생략하면 기본으로 Y가 설정된다. N으로 설정하면 GEOMETRY 객체 수가 1인 것은 multipolygon이 아니라 polygon 타입으로 생성한다. ST_NUMGEOMETRIES 함수로 GEOMETRY 객체 수를 확인할 수 있다.

- 예제

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

5.2. tibero2shp

Tibero Spatial에서는 데이터베이스에 직접 연결되어 테이블의 GEOMETRY를 shapefile 형태로 저장 가능한 **tibero2shp**라는 독립 실행형 유틸리티를 제공한다.

tibero2shp 유틸리티는 Tibero 쿼리의 결과 GEOMETRY들을 export하여 shapefile을 생성한다. 쿼리의 결과 GEOMETRY란 테이블의 GEOMETRY 컬럼을 select하거나, 컬럼을 인자로 받아 GEOMETRY를 반환하는 GEOMETRY 함수의 반환값일 수 있다.

Shapefile의 특성상 한 파일에 동일한 타입의 GEOMETRY만을 담을 수 있으므로, 테이블에서 shapefile에 GEOMETRY를 적재할 쿼리의 결과가 일정한 타입의 GEOMETRY가 되어야한다.

Shapefile에 적재가능한 GEOMETRY 타입으로는 POINT, MULTIPOINT, LINESTRING, MULTI LINESTRING, POLYGON, MULTIPOLYGON이 있다. 적재될 GEOMETRY들의 타입이 POINT와 MULTI POINT인 경우, shapefile에서 POINT와 MULTIPOINT는 다른 타입으로 분류되므로 위 두 타입은 구분하여 넣어야한다. LINESTRING이나 POLYGON은 각각 MULTILINESTRING과 MULTIPOLYGON과 함께 적재 될 수 있다. LINESTRING이나 POLYGON 타입의 GEOMETRY들이 shapefile에 적재되는 경우에는 각각 MULTILINESTRING, MULTIPOLYGON의 타입으로 쓰여진다.

사용법

다음은 tibero2shp의 사용법에 대한 설명이다.

- 사용법

```
tibero2shp <option>
        <username>/<password>@<database name>
        <query>
```

항목	설명
option	tibero2shp에서 사용할 option을 명시한다.
username	데이터베이스의 사용자 이름을 명시한다.
password	데이터베이스의 사용자 비밀번호를 명시한다.
database name	데이터베이스의 이름을 명시한다.
query	shapefile에 적재할 GEOMETRY를 명시하는 쿼리를 입력한다. shapefile의 특성상 동일한 타입의 GEOMETRY객체를 명시하는 쿼리여야한다.

- [options]

항목	설명
-f, --file	생성할 shapefile의 파일 이름을 명시하여 설정한다.

항목	설명
	입력한 이름에 대한 .shp, .shx, .dbf 세가지의 파일이 생성된다. 생성할 shapefile의 경로를 입력할 수 있다. 파일의 이름은 입력하고 경로를 입력하지 않을 시 명령을 실행한 경로에 shapefile이 생성된다. -f 옵션을 사용하지 않을 경우 파일 이름은 shape가 기본값으로 설정된다.
-g, --geom	테이블에 다수의 GEOMETRY 컬럼이 다수일 경우 shapefile로 변환할 컬럼을 명시하여 설정한다. GEOMETRY 컬럼이 다수인데 변환할 컬럼을 명시하지 않은 경우, geom 컬럼에 대해 변환하는 것이 기본으로 설정되어있다.
-h, --help	도움말 화면을 출력한다.

- 예제

```
tibero2shp -f $TB_HOME/testshape tibero/tmax@tibero
        "SELECT GEOM FROM GIS WHERE ST_GEOMETRYTYPE(GEOM)='POLYGON'"
```

5.3. shp2tibero

Tibero Spatial에서는 shapefile 내용을 데이터베이스의 테이블에 적재하는 **shp2tibero**라는 독립 실행형 유틸리티를 제공한다.

shp2tibero 유틸리티는 shapefile의 내용에 해당하는 테이블을 생성한다.

Shapefile이 가질 수 있는 GEOMETRY 타입으로는 POINT, MULTIPOINT, LINESTRING, MULTILINESTRING, POLYGON, MULTIPOLYGON이 있다.

사용법

다음은 shp2tibero의 사용법에 대한 설명이다.

- 사용법

```
shp2tibero <option>
        <username>/<password>@<database name>
        <shapefile>
```

항목	설명
option	shp2tibero에서 사용할 option을 명시한다.

항목	설명
username	데이터베이스의 사용자 이름을 명시한다.
password	데이터베이스의 사용자 비밀번호를 명시한다.
database name	데이터베이스의 이름을 명시한다.
shapefile	이미 존재하는 .shp 파일의 파일 이름을 명시한다.

- [options]

항목	설명
-t, --table	shapefile을 변환할 테이블 명을 명시하여 설정한다. -t 옵션을 사용하지 않을 경우 생성될 테이블명은 shapefile명과 같게 설정된다.
-f, --file	생성할 shapefile의 파일 이름을 명시하여 설정한다. 입력한 이름에 대한 .shp, .shx, .dbf 3가지의 파일이 생성된다. 생성할 shapefile의 경로를 입력할 수 있다. 파일의 이름은 입력하고 경로를 입력하지 않을 시 명령을 실행한 경로에 shapefile이 생성된다. -f 옵션을 사용하지 않을 경우 파일 이름은 shape가 기본값으로 설정된다.
-d, --drop	생성할 테이블과 이름이 같은 테이블이 이미 있는 경우, 기존의 테이블의 삭제 여부를 설정한다. 이 옵션을 활성화할 경우 기존의 테이블이 삭제되고 shapefile로부터 새로운 테이블이 생성된다.
-i, --idcolumn	생성할 테이블의 id 컬럼을 생성하고, 이름을 설정한다.
-e, --endian	생성되는 Geometry type에서 사용될 endian 정보(big/little)를 명시한다. 생략하면 해당 머신 endian 정보를 사용한다.
-p, --polygon	GEOMETRY 객체의 타입을 POLYGON에서 MULTIPOLYGON으로 변환할지를 설정한다. 이 옵션을 활성화하면 POLYGON 타입을 MULTIPOLYGON으로 변환하지 않는다.
-h, --help	도움말 화면을 출력한다.

- 예제

```
shp2tibero -t tablename tibero/tmax@tibero
        testshape
```

색인

T

U

ㄱ