

# Contents

<b>GIS Extensions for Dremio - SQL Function Reference</b>	<b>8</b>
Authored by Brian Holman	8
(1) ST_Area	8
Definition	8
Syntax	8
Return Type	8
Examples	10
(2) ST_AsGeoJSON	10
Definition	10
Syntax	10
Return Type	10
Examples	10
(3) ST_AsText	10
Definition	10
Syntax	10
Return Type	10
Examples	10
(4) ST_Buffer	11
Definition	11
Syntax	11
Return Type	11
Examples	11
(5) ST_Centroid	11
Definition	11
Syntax	11
Return Type	11
Examples	11
(6) ST_Contains	12
Definition	12
Syntax	12
Return Type	12
Examples	12
(7) ST_CoordDim	12
Definition	12
Syntax	12
Return Type	12
Examples	12
(8) ST_Crosses	13
Definition	13
Syntax	13
Return Type	13
Examples	13
(9) ST_Difference	13

	Definition . . . . .	13
	Syntax . . . . .	13
	Return Type . . . . .	13
	Examples . . . . .	13
(10)	ST_Dimension . . . . .	14
	Definition . . . . .	14
	Syntax . . . . .	14
	Return Type . . . . .	14
	Examples . . . . .	14
(11)	ST_Disjoint . . . . .	14
	Definition . . . . .	14
	Syntax . . . . .	14
	Return Type . . . . .	14
	Examples . . . . .	14
(12)	ST_Distance . . . . .	15
	Definition . . . . .	15
	Syntax . . . . .	15
	Return Type . . . . .	15
	Examples . . . . .	15
(13)	ST_DWithin . . . . .	15
	Definition . . . . .	15
	Syntax . . . . .	15
	Return Type . . . . .	15
	Examples . . . . .	15
(14)	ST_Envelope . . . . .	16
	Definition . . . . .	16
	Syntax . . . . .	16
	Return Type . . . . .	16
	Examples . . . . .	16
(15)	ST_Equals . . . . .	16
	Definition . . . . .	16
	Syntax . . . . .	16
	Return Type . . . . .	16
	Examples . . . . .	17
(16)	ST_ExteriorRing . . . . .	17
	Definition . . . . .	17
	Syntax . . . . .	17
	Return Type . . . . .	17
	Examples . . . . .	17
(17)	ST_Generalize . . . . .	17
	Definition . . . . .	17
	Syntax . . . . .	17
	Return Type . . . . .	18
	Examples . . . . .	18
(18)	ST_GeodesicAreaWGS84 . . . . .	18

	Definition . . . . .	18
	Syntax . . . . .	18
	Return Type . . . . .	18
	Examples . . . . .	18
(19)	ST_GeodesicLengthWGS84 . . . . .	18
	Definition . . . . .	18
	Syntax . . . . .	18
	Return Type . . . . .	19
	Examples . . . . .	19
(20)	ST_GeometryN . . . . .	19
	Definition . . . . .	19
	Syntax . . . . .	19
	Return Type . . . . .	19
	Examples . . . . .	19
(21)	ST_GeometryType . . . . .	19
	Definition . . . . .	19
	Syntax . . . . .	19
	Return Type . . . . .	20
	Examples . . . . .	20
(22)	ST_GeomFromEWKB . . . . .	20
	Definition . . . . .	20
	Syntax . . . . .	20
	Return Type . . . . .	20
	Examples . . . . .	20
(23)	ST_GeomFromGeoJSON . . . . .	20
	Definition . . . . .	20
	Syntax . . . . .	20
	Return Type . . . . .	21
	Examples . . . . .	21
(24)	ST_GeomFromText . . . . .	21
	Definition . . . . .	21
	Syntax . . . . .	21
	Return Type . . . . .	21
(25)	ST_GeomFromText . . . . .	21
	Definition . . . . .	21
	Syntax . . . . .	21
	Return Type . . . . .	21
(26)	ST_GeomFromWKB . . . . .	22
	Definition . . . . .	22
	Syntax . . . . .	22
	Return Type . . . . .	22
(27)	ST_GeomFromWKB . . . . .	22
	Definition . . . . .	22
	Syntax . . . . .	22
	Return Type . . . . .	22

(28)	ST_GeoSize . . . . .	22
	Definition . . . . .	22
	Syntax . . . . .	22
	Return Type . . . . .	22
(29)	ST_InteriorRingN . . . . .	23
	Definition . . . . .	23
	Syntax . . . . .	23
	Return Type . . . . .	23
	Examples . . . . .	23
(30)	ST_Intersection . . . . .	23
	Definition . . . . .	23
	Syntax . . . . .	23
	Return Type . . . . .	23
	Examples . . . . .	23
(31)	ST_Intersects . . . . .	23
	Definition . . . . .	24
	Syntax . . . . .	24
	Return Type . . . . .	24
	Examples . . . . .	24
(32)	ST_Is3D . . . . .	24
	Definition . . . . .	24
	Syntax . . . . .	24
	Return Type . . . . .	24
	Examples . . . . .	24
(33)	ST_IsClosed . . . . .	24
	Definition . . . . .	25
	Syntax . . . . .	25
	Return Type . . . . .	25
	Examples . . . . .	25
(34)	ST_IsEmpty . . . . .	25
	Definition . . . . .	25
	Syntax . . . . .	25
	Return Type . . . . .	25
	Examples . . . . .	25
(35)	ST_IsMeasured . . . . .	25
	Definition . . . . .	25
	Syntax . . . . .	26
	Return Type . . . . .	26
	Examples . . . . .	26
(36)	ST_IsRing . . . . .	26
	Definition . . . . .	26
	Syntax . . . . .	26
	Return Type . . . . .	26
	Examples . . . . .	26
(37)	ST_IsSimple . . . . .	26

	Definition . . . . .	26
	Syntax . . . . .	27
	Return Type . . . . .	27
	Examples . . . . .	27
(38)	ST_JSONPath . . . . .	27
	Definition . . . . .	27
	Syntax . . . . .	27
	Return Type . . . . .	27
	Examples . . . . .	27
(39)	ST_Length . . . . .	28
	Definition . . . . .	28
	Syntax . . . . .	28
	Return Type . . . . .	28
	Examples . . . . .	28
(40)	ST_M . . . . .	28
	Definition . . . . .	28
	Syntax . . . . .	29
	Return Type . . . . .	29
	Examples . . . . .	29
(41)	ST_MaxM . . . . .	29
	Definition . . . . .	29
	Syntax . . . . .	29
	Return Type . . . . .	29
	Examples . . . . .	29
(42)	ST_MaxX . . . . .	29
	Definition . . . . .	29
	Syntax . . . . .	29
	Return Type . . . . .	30
	Examples . . . . .	30
(43)	ST_MaxY . . . . .	30
	Definition . . . . .	30
	Syntax . . . . .	30
	Return Type . . . . .	30
	Examples . . . . .	30
(44)	ST_MaxZ . . . . .	30
	Definition . . . . .	30
	Syntax . . . . .	30
	Return Type . . . . .	30
	Examples . . . . .	31
(45)	ST_MinM . . . . .	31
	Definition . . . . .	31
	Syntax . . . . .	31
	Return Type . . . . .	31
	Examples . . . . .	31
(46)	ST_MinX . . . . .	31

	Definition . . . . .	31
	Syntax . . . . .	31
	Return Type . . . . .	31
	Examples . . . . .	31
(47)	ST_MinY . . . . .	32
	Definition . . . . .	32
	Syntax . . . . .	32
	Return Type . . . . .	32
	Examples . . . . .	32
(48)	ST_MinZ . . . . .	32
	Definition . . . . .	32
	Syntax . . . . .	32
	Return Type . . . . .	32
	Examples . . . . .	32
(49)	ST_NumGeometries . . . . .	33
	Definition . . . . .	33
	Syntax . . . . .	33
	Return Type . . . . .	33
	Examples . . . . .	33
(50)	ST_NumInteriorRing . . . . .	33
	Definition . . . . .	33
	Syntax . . . . .	33
	Return Type . . . . .	33
	Examples . . . . .	33
(51)	ST_NumPoints . . . . .	34
	Definition . . . . .	34
	Syntax . . . . .	34
	Return Type . . . . .	34
	Examples . . . . .	34
(52)	ST_Overlaps . . . . .	34
	Definition . . . . .	34
	Syntax . . . . .	34
	Return Type . . . . .	34
	Examples . . . . .	34
(53)	ST_Point . . . . .	35
	Definition . . . . .	35
	Syntax . . . . .	35
	Return Type . . . . .	35
(54)	ST_PointN . . . . .	35
	Definition . . . . .	35
	Syntax . . . . .	35
	Return Type . . . . .	35
	Examples . . . . .	35
(55)	ST_PointZ . . . . .	35
	Definition . . . . .	36

	Syntax . . . . .	36
	Return Type . . . . .	36
(56)	ST_Relate . . . . .	36
	Definition . . . . .	36
	Syntax . . . . .	36
	Return Type . . . . .	36
	Examples . . . . .	36
(57)	ST_SetSRID . . . . .	36
	Definition . . . . .	36
	Syntax . . . . .	37
	Return Type . . . . .	37
(58)	ST_Simplify . . . . .	37
	Definition . . . . .	37
	Syntax . . . . .	37
	Return Type . . . . .	37
(59)	ST_StartPoint . . . . .	37
	Definition . . . . .	37
	Syntax . . . . .	37
	Return Type . . . . .	37
	Examples . . . . .	37
(60)	ST_SymmetricDiff . . . . .	37
	Definition . . . . .	38
	Syntax . . . . .	38
	Return Type . . . . .	38
	Examples . . . . .	38
(61)	ST_Touches . . . . .	38
	Definition . . . . .	38
	Syntax . . . . .	38
	Return Type . . . . .	38
	Examples . . . . .	38
(62)	ST_Transform . . . . .	39
	Definition . . . . .	39
	Syntax . . . . .	39
	Return Type . . . . .	39
	Examples . . . . .	39
(63)	ST_Union . . . . .	39
	Definition . . . . .	39
	Syntax . . . . .	39
	Return Type . . . . .	39
	Examples . . . . .	39
(64)	ST_Within . . . . .	40
	Definition . . . . .	40
	Syntax . . . . .	40
	Return Type . . . . .	40
	Examples . . . . .	40

(65) ST_X	40
Definition	40
Syntax	40
Return Type	40
Examples	40
(66) ST_Y	40
Definition	41
Syntax	41
Return Type	41
Examples	41
(67) ST_Z	41
Definition	41
Syntax	41
Return Type	41
Examples	41

# GIS Extensions for Dremio - SQL Function Reference

Authored by Brian Holman

*This independent project is not affiliated with, sponsored, or endorsed by Dremio Corporation. Dremio is a registered trademark of Dremio Corporation and they retain all trademark and other intellectual property rights. “Dremio” is used here by reference to integrating with their published User-Defined Functions Specification for advanced users to develop their own custom functions for use in SQL queries.*

The **GIS Extensions** allow Dremio to perform standard GIS functions within Dremio SQL with 66 industry-standard GIS functions. These extensions use the *Esri Java Geometry Library* for the underlying implementation of the core geometry functions. The author made heavy use of Esri’s *Spatial Framework for Hadoop* as a reference for a similar implementation that also relies on the same library.

There were two significant gaps in the Geometry Library supplied by Esri that limited transforming geometries from EPSG: 4326 to other coordinate systems and performing geodesic rather than 2D area and length calculations. Geodesic area function helpers backing the ST\_GeodesicAreaWGS84 function are copied almost exactly from the *Trino Geospatial Library* as found in our FunctionHelpers.stSphericalArea() and FunctionHelpers.computeSphericalExcess(). Conversion to other coordinate systems in the ST\_Transform function leverages the Proj4J Library. All of the referenced works are also published under the *Apache 2.0 License*.

## (1) ST\_Area

### Definition

Returns the area of polygon or multipolygon

### Syntax

ST\_Area(binary geometry)

### Return Type

number



Search Spaces and Datasets

New Query Data

SQL Editor

```

1 SELECT
2   ST_AsText(ST_GeomFromEWKB(the_geom)) AS place_wkt,
3   ST_GeodesicAreaWGS84(ST_GeomFromEWKB(the_geom))/4047 AS area_in_acres
4 FROM "postgis".acs."acs_2019_5yr_place"
5 WHERE name = 'Provo'

```

Search functions...

- + ST\_AREA (binary geometry) > number
- + ST\_ASGEOMETRY (binary geometry) > string
- + ST\_ASTEXT (binary geometry) > string
- + ST\_BUFFER (binary geometry, number distance) > binary
- + ST\_CENTROID (binary geometry) > binary
- + ST\_CONTAINS (binary geometry1, binary geometry2) > boolean
- + ST\_COORDDIM (binary geometry) > number
- + ST\_CROSSES (binary geometry1, binary geometry2) > boolean
- + ST\_DWITHIN (binary geometry1, binary geometry2, number distance) > boolean
- + ST\_DIFFERENCE (binary geometry1, binary geometry2) > binary
- + ST\_DIMENSION (binary geometry) > number
- + ST\_DISJOINT (binary geometry1, binary geometry2) > boolean
- + ST\_DISTANCE (binary geometry1, binary geometry2) > number
- + ST\_ENVELOPE (binary geometry) > binary
- + ST\_EQUALS (binary geometry1, binary geometry2) > boolean

Context: @bholman

Add Field Group By Join Column filter 2 fields

Job: Run | Records: 1 | Time: 2s

place_wkt	area_in_acres
MULTIPOLYGON (((-111.74091499972741 40.2371050003955, -111.7409620000962 40.23698500025762, -111.	28235.74996942792

Figure 1: DAC with GIS extensions

## Examples

Query	Result
<code>SELECT ST_Area(ST_GeomFromText('POLYGON ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))</code>	24.0

## (2) ST\_AsGeoJSON

### Definition

Returns the GeoJSON representation of *geometry*.

### Syntax

`ST_AsGeoJSON(binary geometry)`

### Return Type

string

## Examples

Query	Result
<code>SELECT ST_AsGeoJSON(ST_Point(1, 2))</code>	<code>'{"type":"Point","coordinates":[1,2],"crs":{"type":"name","properties":{"na</code>

## (3) ST\_AsText

### Definition

Returns the Well-Known Text (WKT) representation of *geometry*.

### Syntax

`ST_AsText(binary geometry)`

### Return Type

string

## Examples

Query	Result
<code>SELECT ST_AsText(ST_Point(1, 2))</code>	<code>'POINT (1 2)'</code>

## (4) ST\_Buffer

### Definition

Returns geometry object that is the buffer surrounding source *geometry* at specified *distance*.

### Syntax

ST\_Buffer(binary geometry, number distance)

### Return Type

binary

### Examples

Query	Result
SELECT ST_Buffer(ST_Point(0, 0), 1)	polygon approximating a unit circle

## (5) ST\_Centroid

### Definition

Takes a polygon, multipolygon, or multilinestring and returns the point that is in the center of the geometry's envelope. That means that the centroid point is halfway between the geometry's minimum and maximum x and y extents.

### Syntax

ST\_Centroid(binary geometry)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('point (2 3)')))	'POINT(2 3)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('MULTIPOINT ((0 0), (1 1), (1 -1), (6 0))')))	'POINT(2 0)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('linestring (0 0, 6 0)')))	'POINT(3 0)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((0 0, 0 8, 8 8, 8 0, 0 0))')))	'POINT(4 4)'

Query	Result
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((1 1, 5 1, 3 4))')))	'POINT(3 2)'

## (6) ST\_Contains

### Definition

Returns true if *geometry1* contains *geometry2*.

### Syntax

ST\_Contains(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Contains(ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'), ST_Point(2, 3))	true
SELECT ST_Contains(ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'), ST_Point(8, 8))	false

## (7) ST\_CoordDim

### Definition

Returns count of coordinate components.

### Syntax

ST\_CoordDim(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_CoordDim(ST_Point(1.5, 2.5))	2
SELECT ST_CoordDim(ST_GeomFromText('POINTZ (1.5 2.5 3)'))	3

## (8) ST\_Crosses

### Definition

Returns true if *geometry1* crosses *geometry2*, otherwise false.

### Syntax

ST\_Crosses(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Crosses(ST_GeomFromText('LINESTRING (0 0, 1 1)'), ST_GeomFromText('LINESTRING (1 0, 0 1)'))	true
SELECT ST_Crosses(ST_GeomFromText('LINESTRING (2 0, 2 3)'), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	true
SELECT ST_Crosses(ST_GeomFromText('LINESTRING (0 2, 0 1)'), ST_GeomFromText('LINESTRING (2 0, 1 0)'))	false

## (9) ST\_Difference

### Definition

Returns a geometry object that is the difference of the source objects.

### Syntax

ST\_Difference(binary geometry1, binary geometry2)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Difference(ST_GeomFromText('MULTIPOINT (1 1, 1.5 1.5, 2 2)'), ST_Point(1.5, 1.5)))	'MULTIPOINT ((1 1), (2 2))'
SELECT ST_AsText(ST_Difference(ST_GeomFromText('POLYGON ((0 0, 0 10, 10 10, 10 0))'), ST_GeomFromText('POLYGON ((0 0, 0 5, 5 5, 5 0))')))	'POLYGON ((5 0, 10 0, 10 10, 0 10, 0 5, 5 5, 5 0))'

## (10) ST\_\_Dimension

### Definition

Returns spatial dimension of geometry.

### Syntax

ST\_\_Dimension(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_Dimension(ST_Point(1.5, 2.5))	0
SELECT ST_Dimension(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)'))	1
SELECT ST_Dimension(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))'))	2

## (11) ST\_\_Disjoint

### Definition

Returns true if the intersection of the two geometries produces an empty set; otherwise, it returns false.

### Syntax

ST\_Disjoint(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Disjoint(ST_GeomFromText('LINESTRING (0 0, 0 1)'), ST_GeomFromText('LINESTRING (1 1, 1 0)'))	true
SELECT ST_Disjoint(ST_GeomFromText('LINESTRING (0 0, 1 1)'), ST_GeomFromText('LINESTRING (1 0, 0 1)'))	false

## (12) ST\_Distance

### Definition

Returns the distance between two geometry objects.

### Syntax

ST\_Distance(binary geometry1, binary geometry2)

### Return Type

number

### Examples

Query	Result
SELECT ST_Distance(ST_Point(0.0,0.0), ST_Point(3.0,4.0))	5.0

## (13) ST\_DWithin

### Definition

Returns true if the two geometries are within the specified distance of one another; otherwise, it returns false.

### Syntax

ST\_DWithin(binary geometry1, binary geometry2, number distance)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_DWithin(ST_GeomFromText('POLYGON ((10.02 20.01, 11.92 35.64, 25.02 34.15, 19.15 33.94, 10.02 20.01))'), ST_Point(1,2),100)	true
SELECT ST_DWithin(ST_GeomFromText('POLYGON ((101.02 200.01, 111.92 350.64, 250.02 340.15, 190.15 330.94, 101.02 200.01))'), ST_Point(10.02,20.01), 100)	false

## (14) ST\_Envelope

### Definition

Returns the minimum bounding box of the geometry object as a polygon

### Syntax

ST\_Envelope(binary geometry)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Envelope(ST_GeomFromText('LINESTRING (0 0, 2 2)'))))	'POLYGON ((0 0, 2 0, 2 2, 0 2, 0 0))'
SELECT ST_AsText(ST_Envelope(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))'))))	'POLYGON ((2 0, 3 0, 3 3, 2 3, 2 0))'

## (15) ST\_Equals

### Definition

Returns true if the two geometries occupy the same space even if they have a different number of vertices, otherwise it returns false.

### Syntax

ST\_Equals(binary geometry1, binary geometry2)

### Return Type

boolean



## Examples

Query	Result
SELECT ST_Equals(ST_GeomFromText('LINESTRING (0 0, 1 1)'),ST_GeomFromText('LINESTRING (1 1, 0 0)'))	true
SELECT ST_Equals(ST_GeomFromText('LINESTRING (0 0, 1 1)'),ST_GeomFromText('LINESTRING (1 0, 0 1)'))	false
SELECT ST_Equals(ST_GeomFromText('LINESTRING (0 0, 3 3)'),ST_GeomFromText('LINESTRING (3 3, 2 2, 1 1, 0 0)'))	true

## (16) ST\_ExteriorRing

### Definition

Returns the exterior ring of a polygon as a linestring.

### Syntax

ST\_ExteriorRing(binary geometry)

### Return Type

binary

## Examples

Query	Result
SELECT ST_AsText(ST_ExteriorRing(ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4 1))')))	'LINESTRING (1 1, 4 1, 1 4, 1 1)'
SELECT ST_AsText(ST_ExteriorRing(ST_GeomFromText('POLYGON ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))')))	'LINESTRING (0 0, 8 0, 0 8, 0 0)'

## (17) ST\_Generalize

### Definition

Simplifies geometries using the Douglas-Peucker algorithm. *maxDeviation* is the maximum allowed deviation from the generalized geometry to the original geometry. When *removeDegenerateParts* is true, the degenerate parts of the geometry will be removed from the output.

### Syntax

ST\_Generalize(binary geometry, number maxDeviation, boolean removeDegenerateParts)

## Return Type

binary

## Examples

Query	Result
<pre>SELECT ST_AsText(ST_Generalize(ST_GeomFromText('POLYGON ((0 0, 1 1, 2 0, 3 2, 4 1, 5 0, 5 10, 0 10))'), 2, true))</pre>	<pre>'POLYGON ((0 0, 5 0, 5 10, 0 10, 0 0))'</pre>

## (18) ST\_GeodesicAreaWGS84

### Definition

Returns the area in square meters of a geometry on the Earth's surface using spherical model. Requires the geometry to be in the WGS84 spatial reference.

### Syntax

ST\_GeodesicAreaWGS84(binary geometry)

## Return Type

number

## Examples

Query	Result
<pre>SELECT ST_GeodesicAreaWGS84(ST_GeomFromText('POLYGON ((-114.04702599994988 39.90609700007656, -114.0500520000997 37.0001909997149, -109.04517199998776 36.99897700038832, -109.05002599989996 41.000691000389395, -111.04681499981234 40.997875000031286, -111.04671399965133 42.00170200004732, -114.04147700036322 41.99387299963928, -114.04702599994988 39.90609700007656))'))/4047 AS utah_acreage</pre>	<pre>5.416484897473004E7</pre>

## (19) ST\_GeodesicLengthWGS84

### Definition

Returns distance along line on WGS84 spheroid, in meters, for geographic coordinates. Requires the geometry to be in the WGS84 spatial reference.

### Syntax

ST\_GeodesicLengthWGS84(binary geometry)

## Return Type

number

## Examples

Query	Result
<pre>SELECT SELECT ST_GeodesicLengthWGS84(ST_GeomFromText('MultiLineString((0.0 80.0, 0.3 80.4))', 4326))</pre>	45026.96274781222

## (20) ST\_GeometryN

### Definition

Takes a geometry collection and an integer index (1-based index) and returns the nth geometry object in the collection.

### Syntax

ST\_GeometryN(binary geometry, number index)

## Return Type

binary

## Examples

Query	Result
<pre>SELECT ST_AsText(ST_GeometryN(ST_GeomFromText('MULTIPOINT (10 40, 40 30, 20 20, 30 10)'), 3))</pre>	'POINT (20 20)'
<pre>SELECT ST_AsText(ST_GeometryN(ST_GeomFromText('MULTILINESTRING ((2 4, 10 10), (20 20, 7 8))'), 2))</pre>	'LINESTRING (20 20, 7 8)'

## (21) ST\_GeometryType

### Definition

Takes a geometry object and returns its geometry type (for example, Point, Line, Polygon, MultiPoint) as a string.

### Syntax

ST\_GeometryType(binary geometry)

## Return Type

string

## Examples

Query	Result
SELECT ST_GeometryType(ST_Point(1.5, 2.5))	'ST_POINT'
SELECT ST_GeometryType(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)'))	'ST_LINESTRING'
SELECT ST_GeometryType(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))'))	'ST_POLYGON'

## (22) ST\_GeomFromEWKB

### Definition

Converts a Hex encoded binary string from Postgres/PostGIS geometry to native geometry including embedded SRID.

### Syntax

ST\_GeomFromEWKB(string hexEncodedGeometry)

## Return Type

binary

## Examples

Query	Result
SELECT SELECT ST_AsText(ST_GeomFromEWKB(the_geom)) FROM table("postgis".external_query('SELECT ST_GeomFromText('POINT(-71.064544 42.28787)',4326) AS the_geom'))	'POINT (-71.064544 42.28787)'

## (23) ST\_GeomFromGeoJSON

### Definition

Constructs a geometry from GeoJSON.

### Syntax

ST\_GeomFromGeoJSON(string geoJsonString)

## Return Type

binary

## Examples

Query	Result
<code>SELECT ST_AsText(ST_GeomFromGeoJSON('{\"type\":\"Point\", \"coordinates\":[1.2, 2.4]}'))</code>	<code>'POINT (1.2 2.4)'</code>
<code>SELECT ST_AsText(ST_GeomFromGeoJSON('{\"type\":\"LineString\", \"coordinates\":[[1,2], [3,4]]}'))</code>	<code>'LINESTRING (1 2, 3 4)'</code>

## (24) ST\_GeomFromText

### Definition

Takes a well-known text representation and returns a geometry object.

### Syntax

`ST_GeomFromText(string wktString)`

## Return Type

binary

---

## (25) ST\_GeomFromText

### Definition

Takes a well-known text representation and a spatial reference ID and returns a geometry object.

### Syntax

`ST_GeomFromText(string wktString, number SRID)`

## Return Type

binary

---

## (26) ST\_GeomFromWKB

### Definition

Takes a well-known binary (WKB) representation and returns a geometry object.

### Syntax

ST\_GeomFromWKB(binary wkbValue)

### Return Type

binary

---

## (27) ST\_GeomFromWKB

### Definition

Takes a well-known binary (WKB) representation and a spatial reference ID and returns a geometry object.

### Syntax

ST\_GeomFromWKB(binary wkbValue, number SRID)

### Return Type

binary

---

## (28) ST\_GeoSize

### Definition

Takes a geometry object and returns its size in bytes.

### Syntax

ST\_GeoSize(binary geometry)

### Return Type

number

---

## (29) ST\_InteriorRingN

### Definition

Returns a LineString which is the nth interior ring of the input Polygon (1-based index)

### Syntax

ST\_InteriorRingN(binary geometry, number index)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_InteriorRingN(ST_GeomFromText('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'), 1))	'LINESTRING (1 1, 1 5, 5 1, 1 1)'

## (30) ST\_Intersection

### Definition

Returns a geometry object that is the geometric intersection of the source objects.

### Syntax

ST\_Intersection(binary geometry1, binary geometry2)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Intersection(ST_Point(1,1), ST_Point(1,1)))	'POINT (1 1)'
SELECT ST_AsText(ST_Intersection(ST_GeomFromText('LINESTRING(0 2, 0 0, 2 0)'), ST_GeomFromText('LINESTRING(0 3, 0 1, 1 0, 3 0)')))	'MULTILINESTRING ((1 0, 2 0), (0 2, 0 1))'
SELECT ST_AsText(ST_Intersection(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 3 0))'), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))')))	'POLYGON ((2 1, 2.666666666666667 1, 2 3, 2 1))'

## (31) ST\_Intersects

## Definition

Returns true if *geometry1* intersects with *geometry2*, otherwise returns false.

## Syntax

ST\_Intersects(binary geometry1, binary geometry2)

## Return Type

boolean

## Examples

Query	Result
SELECT ST_Intersects(ST_GeomFromText('LINESTRING (2 0, 2 3)'), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))'))	true
SELECT ST_Intersects(ST_GeomFromText('LINESTRING (8 7, 7 8)'), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))'))	false

## (32) ST\_Is3D

### Definition

Returns true if the geometry object is three-dimensional including height 'Z', otherwise returns false.

### Syntax

ST\_Is3D(binary geometry)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Is3D(ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	false
SELECT ST_Is3D(ST_GeomFromText('LINESTRING (0 0, 3 4, 0 4, 0 0))'))	false
SELECT ST_Is3D(ST_Point(3, 4))	false
SELECT ST_Is3D(ST_PointZ(3, 4, 2))	true

## (33) ST\_IsClosed



### Definition

Return true if the linestring or multi-line has start and end points that are coincident.

### Syntax

ST\_IsClosed(binary geometry)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_IsClosed(ST_GeomFromText('LINESTRING(0 0, 3 4, 0 4, 0 0)'))	true
SELECT ST_IsClosed(ST_GeomFromText('LINESTRING(0 0, 3 4)'))	false

## (34) ST\_IsEmpty

### Definition

Return true if the geometry object is empty of geometric information.

### Syntax

ST\_IsEmpty(binary geometry)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_IsEmpty(ST_Point(1.5, 2.5))	false
SELECT ST_IsEmpty(ST_GeomFromText('POINT EMPTY'))	true

## (35) ST\_IsMeasured

### Definition

Returns true if the geometry object is measured including an additional dimension 'M', otherwise returns false.

### Syntax

ST\_IsMeasured(binary geometry)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_IsMeasured(ST_PointZ(3, 4, 2))	false
SELECT ST_IsMeasured(ST_GeomFromText('POINT M (1 1 80)'))	true
SELECT ST_IsMeasured(ST_GeomFromText('POINT ZM (1 1 5 60)'))	true

## (36) ST\_IsRing

### Definition

Returns true if the geometry is a linestring and the linestring is closed and simple.

### Syntax

ST\_IsRing(binary geometry)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_IsRing(ST_GeomFromText('LINESTRING (0 0, 3 4, 0 4, 0 0)'))	true
SELECT ST_IsRing(ST_GeomFromText('LINESTRING (0 0, 1 1, 1 2, 2 1, 1 1, 0 0)'))	false
SELECT ST_IsRing(ST_GeomFromText('LINESTRING (0 0, 3 4)'))	false

## (37) ST\_IsSimple

### Definition

Returns true if the geometry object is simple as defined by the Open Geospatial Consortium (OGC), otherwise, it returns false

**Syntax**

ST\_IsSimple(binary geometry)

**Return Type**

boolean

**Examples**

Query	Result
SELECT ST_IsSimple(ST_Point(1.5, 2.5))	true
SELECT ST_IsSimple(ST_GeomFromText('LINESTRING (0 0, 1 1, 0 1, 1 0)'))	false

**(38) ST\_JSONPath**

**Definition**

Extract a portion of *jsonData* as a string by following the specified path in the JSON Object from *jsonPath*.

**Syntax**

ST\_JSONPath(string jsonPath, string jsonData)

**Return Type**

string

**Examples**

Query	Result
SELECT SELECT ST_JSONPath('/coordinates[Array][0]',ST_AsGeoJSON(ST_Envelope(the_geom))) FROM utah_county_taxparcels	undefined
SELECT SELECT ST_JSONPath('/crs[Object]/properties[Object]/name',ST_AsGeoJSON(ST_Envelope(the_geom))) FROM utah_county_taxparcels	undefined
SELECT Example JSON Paths (similar to XPath for XML):	undefined
SELECT '/data[Array]'	undefined
SELECT '/data[Array][1]/id[String]'	undefined
SELECT '/data[Array][1]/likes[Object]'	undefined
SELECT '/data[Array][1]/likes[Object]/summary[Object]/total_count[String]'	undefined
SELECT '/data[Array][3]'	undefined

Query	Result
SELECT '/data[Array][id=131272076894593_1420960724592382]/likes[Object]/summary[Object]/total_count'	undefined
SELECT '/fbids[String]'	undefined
SELECT '/quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/enterpriseValue[Object]/fmt[String]'	undefined
SELECT '/quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/forwardPE[Object]/raw[Double]'	undefined
SELECT 'quoteSummary[6]/result[4][0]/defaultKeyStatistics[6]/sharesOutstanding[6]/raw[1]'	undefined
SELECT 'quoteSummary[6]/result[Array]'	undefined
SELECT 'quoteSummary[6]/result[Array][0]'	undefined
SELECT 'quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/lastSplitDate[Object]/raw1[Long]'	undefined
SELECT 'quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/sharesOutstanding[Object]/raw[Integer]'	undefined

## (39) ST\_Length

### Definition

Returns the length of a line string or multiline string.

### Syntax

ST\_Length(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_Length(ST_GeomFromText('LINESTRING (0 0, 3 4)'))	5.0
SELECT ST_Length(ST_GeomFromText('MULTILINESTRING ((1 0, 2 0), (0 2, 0 1))'))	2.0

## (40) ST\_M

### Definition

Takes a Point as an input parameter and returns its measure m-coordinate.

**Syntax**

ST\_M(binary geometry)

**Return Type**

number

**Examples**

Query	Result
SELECT ST_M(ST_GeomFromText('POINT M (1 1 80)'))	80.0
SELECT ST_M(ST_GeomFromText('POINT ZM (1 1 5 60)'))	60.0

**(41) ST\_MaxM****Definition**

Takes a geometry as an input parameter and returns its maximum measure m-coordinate.

**Syntax**

ST\_MaxM(binary geometry)

**Return Type**

number

**Examples**

Query	Result
SELECT ST_MaxM(ST_GeomFromText('LINESTRING M (1.5 2.5 2, 3.0 2.2 1)'))	2.0
SELECT ST_MaxM(ST_GeomFromText('POINT M (1.5 2.5 3)'))	3.0

**(42) ST\_MaxX****Definition**

Takes a geometry as an input parameter and returns its maximum x-coordinate.

**Syntax**

ST\_MaxX(binary geometry)

## Return Type

number

## Examples

Query	Result
SELECT ST_MaxX(ST_GeomFromText('LINESTRING M (1.5 2.5 2, 3.0 2.2 1)'))	3.0
SELECT ST_MaxX(ST_GeomFromText('POINT M (1.5 2.5 3)'))	1.5

## (43) ST\_MaxY

### Definition

Takes a geometry as an input parameter and returns its maximum y-coordinate.

### Syntax

ST\_MaxY(binary geometry)

## Return Type

number

## Examples

Query	Result
SELECT ST_MaxY(ST_GeomFromText('LINESTRING M (1.5 2.5 2, 3.0 2.2 1)'))	2.5
SELECT ST_MaxY(ST_GeomFromText('POINT M (1.5 2.5 3)'))	2.5

## (44) ST\_MaxZ

### Definition

Takes a geometry as an input parameter and returns its maximum z-coordinate.

### Syntax

ST\_MaxZ(binary geometry)

## Return Type

number

## Examples

Query	Result
SELECT ST_MaxZ(ST_GeomFromText('LINESTRING ZM (1.5 2.5 2 60, 3.0 2.2 1 80)'))	2.0
SELECT ST_MaxZ(ST_GeomFromText('LINESTRING Z (1.5 2.5 3, 3.0 2.2 4)'))	4.0

## (45) ST\_MinM

### Definition

Takes a geometry as an input parameter and returns its minimum m-coordinate.

### Syntax

ST\_MinM(binary geometry)

### Return Type

number

## Examples

Query	Result
SELECT ST_MinM(ST_GeomFromText('LINESTRING M (1.5 2.5 2, 3.0 2.2 1)'))	1.0
SELECT ST_MinM(ST_GeomFromText('POINT M (1.5 2.5 3)'))	3.0

## (46) ST\_MinX

### Definition

Takes a geometry as an input parameter and returns its minimum x-coordinate.

### Syntax

ST\_MinX(binary geometry)

### Return Type

number

## Examples

Query	Result
SELECT ST_MinX(ST_GeomFromText('LINESTRING M (1.25 2.5 2, 3.0 2.2 1)'))	1.25
SELECT ST_MinX(ST_GeomFromText('POINT M (1.75 2.5 3)'))	1.75

## (47) ST\_MinY

### Definition

Takes a geometry as an input parameter and returns its minimum y-coordinate.

### Syntax

ST\_MinY(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_MinY(ST_GeomFromText('LINESTRING M (1.5 2.5 2, 3.0 2.2 1)'))	2.2
SELECT ST_MinY(ST_GeomFromText('POINT M (1.5 2.25 3)'))	2.25

## (48) ST\_MinZ

### Definition

Takes a geometry as an input parameter and returns its minimum z-coordinate.

### Syntax

ST\_MinZ(binary geometry)

### Return Type

number

### Examples



Query	Result
SELECT ST_MinZ(ST_GeomFromText('LINESTRING ZM (1.5 2.5 2 60, 3.0 2.2 1 80)'))	1.0
SELECT ST_MinZ(ST_GeomFromText('LINESTRING Z (1.5 2.5 3, 3.0 2.2 4)'))	3.0

## (49) ST\_NumGeometries

### Definition

Returns the number of geometries in the geometry collection.

### Syntax

ST\_NumGeometries(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_NumGeometries(ST_GeomFromText('MULTIPOINT ((10 40), (40 30), (20 20), (30 10))'))	4
SELECT ST_NumGeometries(ST_GeomFromText('MULTILINESTRING ((2 4, 10 10), (20 20, 7 8))'))	2

## (50) ST\_NumInteriorRing

### Definition

Returns the number of interior rings in the polygon geometry.

### Syntax

ST\_NumInteriorRing(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_NumInteriorRing(ST_GeomFromText('POLYGON ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))'))	1

## (51) ST\_NumPoints

### Definition

Returns the number of points (vertices) in the geometry. For polygons, both the starting and ending vertices are counted, even though they occupy the same location.

### Syntax

ST\_NumPoints(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_NumPoints(ST_Point(1.5, 2.5))	1
SELECT ST_NumPoints(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)'))	2
SELECT ST_NumPoints((ST_GeomFromText('POLYGON ((0 0, 10 0, 0 10, 0 0))')))	4

## (52) ST\_Overlaps

### Definition

Returns true if *geometry1* overlaps *geometry2*.

### Syntax

ST\_Overlaps(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Overlaps(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))'), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	true
SELECT ST_Overlaps(ST_GeomFromText('POLYGON ((2 0, 2 1, 3 1))'), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	false

## (53) ST\_Point

### Definition

Returns a 2D point geometry from the provided lon (x) and lat (y) values.

### Syntax

ST\_Pointnumber lon, number lat

### Return Type

binary

## (54) ST\_PointN

### Definition

Returns the point that is the nth vertex in an LineString or MultiPoint (1-based index)

### Syntax

ST\_PointN(binary geometry, number index)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_PointN(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)'), 2))	'POINT (3 2.2)'

## (55) ST\_PointZ

## Definition

Returns a 3D point geometry from the provided lon (x), lat (y), and elev (z) values.

## Syntax

ST\_PointZnumber lon, number lat, number elev

## Return Type

binary

---

## (56) ST\_Relate

### Definition

Compares the two geometries and returns true if the geometries meet the conditions specified by the DE-9IM pattern matrix string, otherwise, false is returned.

### Syntax

ST\_Relate(binary geometry1, binary geometry2, string relation)

### Return Type

binary

### Examples

Query	Result
SELECT ST_Relate(ST_GeomFromText('POLYGON ((2 0, 2 1, 3 1))'), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'), '****T****')	true
SELECT ST_Relate(ST_GeomFromText('POLYGON ((2 0, 2 1, 3 1))'), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'), 'T*****')	false
SELECT ST_Relate(ST_GeomFromText('LINESTRING (0 0, 3 3)'), ST_GeomFromText('LINESTRING (1 1, 4 4)'), 'T*****')	true
SELECT ST_Relate(ST_GeomFromText('LINESTRING (0 0, 3 3)'), ST_GeomFromText('LINESTRING (1 1, 4 4)'), '****T****')	false

## (57) ST\_SetSRID

### Definition

Sets the Spatial Reference ID of *SRID* of the geometry.

**Syntax**

ST\_SetSRID(binary geometry, number SRID)

**Return Type**

binary

---

**(58) ST\_Simplify**

**Definition**

Simplifies the geometry or determines if the geometry is simple. The goal is to produce a geometry that is valid to store without additional processing.

**Syntax**

ST\_Simplify(binary geometry)

**Return Type**

binary

---

**(59) ST\_StartPoint**

**Definition**

Returns the first point of a Linestring.

**Syntax**

ST\_StartPoint(binary geometry)

**Return Type**

binary

**Examples**

Query	Result
SELECT ST_AsText(ST_StartPoint(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)')))	'POINT(1.5 2.5)'

**(60) ST\_SymmetricDiff**

## Definition

Returns a geometry object that is the symmetric difference of the source objects.

## Syntax

ST\_SymmetricDiff(binary geometry1, binary geometry2)

## Return Type

binary

## Examples

Query	Result
SELECT ST_AsText(ST_SymmetricDiff(ST_GeomFromText('LINESTRING (0 2, 2 2)'), ST_GeomFromText('LINESTRING (1 2, 3 2)')))	'MULTILINESTRING ((0 2, 1 2), (2 2, 3 2))'
SELECT ST_AsText(ST_SymmetricDiff(ST_GeomFromText('POLYGON ((0 0, 2 0, 2 2, 0 2, 0 0))'), ST_GeomFromText('POLYGON ((1 1, 3 1, 3 3, 1 3, 1 1))')))	undefined
--> 'MULTIPOLYGON (((0 0, 2 0, 2 1, 1 1, 1 2, 0 2, 0 0)), ((2 1, 3 1, 3 3, 1 3, 1 2, 2 2, 2 1)))'	

## (61) ST\_Touches

### Definition

Returns true if none of the points common to both geometries intersect the interiors of both geometries, otherwise, it returns false. At least one geometry must be a LineString, Polygon, MultiLineString, or MultiPolygon.

### Syntax

ST\_Touches(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Touches(ST_Point(1, 2), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	true
SELECT ST_Touches(ST_Point(8, 8), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	false

## (62) ST\_Transform

### Definition

Takes the two-dimensional geometry as input and returns values converted from the spatial source reference specified by *sourceSRID* to the one specified by *targetSRID*.

### Syntax

ST\_Transform(binary geometry, number sourceSRID, number targetSRID)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Transform(ST_GeomFromText('POLYGON ((-114.04702599994988 39.90609700007656, -114.0500520000997 37.0001909997149, -109.04517199998776 36.99897700038832, -109.05002599989996 41.000691000389395, -111.04681499981234 40.997875000031286, -111.04671399965133 42.00170200004732, -114.04147700036322 41.99387299963928, -114.04702599994988 39.90609700007656))'), 4326, 3857))	'POLYGON ((-12695656.860801652 4852305.919673687, -12695993.71359747 4439133.410181124, -12138853.020503571 4438964.195256694, -12139393.365302108 5012443.58678148, -12361674.899993964 5012028.231889712, -12361663.65670747 5161234.398812287, -12695039.148993252 5160061.69329091, -12695656.860801652 4852305.919673687))'

## (63) ST\_Union

### Definition

Returns a geometry as the union of the two supplied geometries.

### Syntax

ST\_Union(binary geometry1, binary geometry2)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(ST_Union(ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'), ST_GeomFromText('POLYGON ((4 1, 4 4, 4 8, 8 1))'))	'POLYGON ((1 1, 4 1, 8 1, 4 8, 4 4, 1 4, 1 1))'

## (64) ST\_Within

### Definition

Returns true if *geometry1* is completely inside *geometry2*.

### Syntax

ST\_Within(binary geometry1, binary geometry2)

### Return Type

boolean

### Examples

Query	Result
SELECT ST_Within(ST_Point(2, 3), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	true
SELECT ST_Within(ST_Point(8, 8), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))'))	false

## (65) ST\_X

### Definition

Takes a Point as an input parameter and returns its longitude (x) coordinate.

### Syntax

ST\_X(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_X(ST_Point(5, 7))	5.0

## (66) ST\_Y



### Definition

Takes a Point as an input parameter and returns its latitude (y) coordinate.

### Syntax

ST\_Y(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_Y(ST_GeomFromText('POINT (5 7)'))	7.0

## (67) ST\_Z

### Definition

Takes a Point as an input parameter and returns its elevation (z) coordinate.

### Syntax

ST\_Z(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_Z(ST_GeomFromText('POINT Z (5 7 9)'))	9.0