

# GIS Extensions for Dremio - SQL Functions Reference

Brian K. Holman

## Abstract

Complete Reference to the 76 GIS-related functions and 21 H3 index functions including syntax, return type, and examples.

## Contents

<b>GIS Extensions for Dremio - SQL Function Reference</b>	<b>13</b>
Author . . . . .	13
Legal Disclaimer . . . . .	14
Third-Party Libraries . . . . .	15
(1) H3_AsText . . . . .	15
Definition . . . . .	15
Syntax . . . . .	15
Return Type . . . . .	15
Examples . . . . .	15
(2) H3_Boundary . . . . .	15
Definition . . . . .	15
Syntax . . . . .	16
Return Type . . . . .	16
Examples . . . . .	16
(3) H3_Center . . . . .	16
Definition . . . . .	16
Syntax . . . . .	16
Return Type . . . . .	16
Examples . . . . .	16
(4) H3_Compact . . . . .	17
Definition . . . . .	17
Syntax . . . . .	17
Return Type . . . . .	17

Examples . . . . .	17
(5) H3_Distance . . . . .	17
Definition . . . . .	17
Syntax . . . . .	17
Return Type . . . . .	18
Examples . . . . .	18
(6) H3_FromGeomPoint . . . . .	18
Definition . . . . .	18
Syntax . . . . .	18
Return Type . . . . .	18
Examples . . . . .	18
(7) H3_FromGeomPoly . . . . .	18
Definition . . . . .	19
Syntax . . . . .	19
Return Type . . . . .	19
Examples . . . . .	19
(8) H3_FromLongLat . . . . .	19
Definition . . . . .	19
Syntax . . . . .	19
Return Type . . . . .	19
Examples . . . . .	20
(9) H3_FromText . . . . .	20
Definition . . . . .	20
Syntax . . . . .	20
Return Type . . . . .	20
Examples . . . . .	20
(10) H3_HexRing . . . . .	20
Definition . . . . .	20
Syntax . . . . .	20
Return Type . . . . .	21
Examples . . . . .	21
(11) H3_IsPentagon . . . . .	21
Definition . . . . .	21
Syntax . . . . .	21
Return Type . . . . .	21
Examples . . . . .	21
(12) H3_IsValid . . . . .	21
Definition . . . . .	21
Syntax . . . . .	22

	Return Type . . . . .	22
	Examples . . . . .	22
(13)	H3_KRing . . . . .	22
	Definition . . . . .	22
	Syntax . . . . .	22
	Return Type . . . . .	22
	Examples . . . . .	22
(14)	H3_KRing_Distances . . . . .	22
	Definition . . . . .	23
	Syntax . . . . .	23
	Return Type . . . . .	23
	Examples . . . . .	23
(15)	H3_Polyfill . . . . .	23
	Definition . . . . .	23
	Syntax . . . . .	23
	Return Type . . . . .	23
	Examples . . . . .	23
(16)	H3_Resolution . . . . .	24
	Definition . . . . .	24
	Syntax . . . . .	24
	Return Type . . . . .	24
	Examples . . . . .	24
(17)	H3_ToCenterChild . . . . .	24
	Definition . . . . .	24
	Syntax . . . . .	24
	Return Type . . . . .	24
	Examples . . . . .	25
(18)	H3_ToChildren . . . . .	25
	Definition . . . . .	25
	Syntax . . . . .	25
	Return Type . . . . .	25
	Examples . . . . .	25
(19)	H3_ToParent . . . . .	25
	Definition . . . . .	25
	Syntax . . . . .	26
	Return Type . . . . .	26
	Examples . . . . .	26
(20)	H3_Uncompact . . . . .	26
	Definition . . . . .	26

	Syntax . . . . .	26
	Return Type . . . . .	26
	Examples . . . . .	26
(21)	H3_Wrap . . . . .	27
	Definition . . . . .	27
	Syntax . . . . .	27
	Return Type . . . . .	27
	Examples . . . . .	27
(22)	ST_AggrConvexHull . . . . .	27
	Definition . . . . .	27
	Syntax . . . . .	27
	Return Type . . . . .	27
	Examples . . . . .	27
(23)	ST_AggrIntersection . . . . .	28
	Definition . . . . .	28
	Syntax . . . . .	28
	Return Type . . . . .	28
	Examples . . . . .	28
(24)	ST_AggrUnion . . . . .	28
	Definition . . . . .	29
	Syntax . . . . .	29
	Return Type . . . . .	29
	Examples . . . . .	29
(25)	ST_Area . . . . .	29
	Definition . . . . .	29
	Syntax . . . . .	29
	Return Type . . . . .	29
	Examples . . . . .	29
(26)	ST_AsGeoJSON . . . . .	30
	Definition . . . . .	30
	Syntax . . . . .	30
	Return Type . . . . .	30
	Examples . . . . .	30
(27)	ST_AsText . . . . .	30
	Definition . . . . .	30
	Syntax . . . . .	30
	Return Type . . . . .	30
	Examples . . . . .	31
(28)	ST_Boundary . . . . .	31

	Definition . . . . .	31
	Syntax . . . . .	31
	Return Type . . . . .	31
	Examples . . . . .	31
(29)	ST_Buffer . . . . .	31
	Definition . . . . .	31
	Syntax . . . . .	32
	Return Type . . . . .	32
	Examples . . . . .	32
(30)	ST_Centroid . . . . .	32
	Definition . . . . .	32
	Syntax . . . . .	32
	Return Type . . . . .	32
	Examples . . . . .	32
(31)	ST_Contains . . . . .	33
	Definition . . . . .	33
	Syntax . . . . .	33
	Return Type . . . . .	33
	Examples . . . . .	33
(32)	ST_ConvexHull . . . . .	33
	Definition . . . . .	34
	Syntax . . . . .	34
	Return Type . . . . .	34
	Examples . . . . .	34
(33)	ST_CoordDim . . . . .	34
	Definition . . . . .	34
	Syntax . . . . .	34
	Return Type . . . . .	34
	Examples . . . . .	34
(34)	ST_Crosses . . . . .	35
	Definition . . . . .	35
	Syntax . . . . .	35
	Return Type . . . . .	35
	Examples . . . . .	35
(35)	ST_Densify . . . . .	35
	Definition . . . . .	36
	Syntax . . . . .	36
	Return Type . . . . .	36
	Examples . . . . .	36

(36) ST_Difference . . . . .	36
Definition . . . . .	36
Syntax . . . . .	36
Return Type . . . . .	36
Examples . . . . .	36
(37) ST_Dimension . . . . .	37
Definition . . . . .	37
Syntax . . . . .	37
Return Type . . . . .	37
Examples . . . . .	37
(38) ST_Disjoint . . . . .	37
Definition . . . . .	38
Syntax . . . . .	38
Return Type . . . . .	38
Examples . . . . .	38
(39) ST_Distance . . . . .	38
Definition . . . . .	38
Syntax . . . . .	38
Return Type . . . . .	38
Examples . . . . .	38
(40) ST_DWithin . . . . .	39
Definition . . . . .	39
Syntax . . . . .	39
Return Type . . . . .	39
Examples . . . . .	39
(41) ST_EndPoint . . . . .	39
Definition . . . . .	39
Syntax . . . . .	39
Return Type . . . . .	40
Examples . . . . .	40
(42) ST_Envelope . . . . .	40
Definition . . . . .	40
Syntax . . . . .	40
Return Type . . . . .	40
Examples . . . . .	40
(43) ST_EnvIntersects . . . . .	40
Definition . . . . .	41
Syntax . . . . .	41
Return Type . . . . .	41

Examples . . . . .	41
(44) ST_Equals . . . . .	41
Definition . . . . .	41
Syntax . . . . .	41
Return Type . . . . .	41
Examples . . . . .	41
(45) ST_ExteriorRing . . . . .	42
Definition . . . . .	42
Syntax . . . . .	42
Return Type . . . . .	42
Examples . . . . .	42
(46) ST_Generalize . . . . .	43
Definition . . . . .	43
Syntax . . . . .	43
Return Type . . . . .	43
Examples . . . . .	43
(47) ST_GeodesicAreaWGS84 . . . . .	43
Definition . . . . .	43
Syntax . . . . .	43
Return Type . . . . .	43
Examples . . . . .	44
(48) ST_GeodesicLengthWGS84 . . . . .	44
Definition . . . . .	44
Syntax . . . . .	44
Return Type . . . . .	44
Examples . . . . .	44
(49) ST_GeometryN . . . . .	45
Definition . . . . .	45
Syntax . . . . .	45
Return Type . . . . .	45
Examples . . . . .	45
(50) ST_GeometryType . . . . .	45
Definition . . . . .	45
Syntax . . . . .	45
Return Type . . . . .	45
Examples . . . . .	46
(51) ST_GeomFromEWKB . . . . .	46
Definition . . . . .	46
Syntax . . . . .	46

	Return Type . . . . .	46
	Examples . . . . .	46
(52)	ST_GeomFromGeoJSON . . . . .	46
	Definition . . . . .	46
	Syntax . . . . .	47
	Return Type . . . . .	47
	Examples . . . . .	47
(53)	ST_GeomFromText . . . . .	47
	Definition . . . . .	47
	Syntax . . . . .	47
	Return Type . . . . .	47
(54)	ST_GeomFromText . . . . .	47
	Definition . . . . .	48
	Syntax . . . . .	48
	Return Type . . . . .	48
(55)	ST_GeomFromWKB . . . . .	48
	Definition . . . . .	48
	Syntax . . . . .	48
	Return Type . . . . .	48
(56)	ST_GeomFromWKB . . . . .	48
	Definition . . . . .	48
	Syntax . . . . .	48
	Return Type . . . . .	49
(57)	ST_GeoSize . . . . .	49
	Definition . . . . .	49
	Syntax . . . . .	49
	Return Type . . . . .	49
(58)	ST_InteriorRingN . . . . .	49
	Definition . . . . .	49
	Syntax . . . . .	49
	Return Type . . . . .	49
	Examples . . . . .	49
(59)	ST_Intersection . . . . .	50
	Definition . . . . .	50
	Syntax . . . . .	50
	Return Type . . . . .	50
	Examples . . . . .	50
(60)	ST_Intersects . . . . .	50
	Definition . . . . .	51



	Syntax . . . . .	51
	Return Type . . . . .	51
	Examples . . . . .	51
(61)	ST_Is3D . . . . .	51
	Definition . . . . .	51
	Syntax . . . . .	51
	Return Type . . . . .	51
	Examples . . . . .	51
(62)	ST_IsClosed . . . . .	52
	Definition . . . . .	52
	Syntax . . . . .	52
	Return Type . . . . .	52
	Examples . . . . .	52
(63)	ST_IsEmpty . . . . .	52
	Definition . . . . .	52
	Syntax . . . . .	53
	Return Type . . . . .	53
	Examples . . . . .	53
(64)	ST_IsMeasured . . . . .	53
	Definition . . . . .	53
	Syntax . . . . .	53
	Return Type . . . . .	53
	Examples . . . . .	53
(65)	ST_IsRing . . . . .	54
	Definition . . . . .	54
	Syntax . . . . .	54
	Return Type . . . . .	54
	Examples . . . . .	54
(66)	ST_IsSimple . . . . .	54
	Definition . . . . .	54
	Syntax . . . . .	54
	Return Type . . . . .	54
	Examples . . . . .	55
(67)	ST_JSONPath . . . . .	55
	Definition . . . . .	55
	Syntax . . . . .	55
	Return Type . . . . .	55
	Examples . . . . .	55
(68)	ST_Length . . . . .	56

	Definition . . . . .	56
	Syntax . . . . .	56
	Return Type . . . . .	56
	Examples . . . . .	56
(69)	ST_M . . . . .	56
	Definition . . . . .	57
	Syntax . . . . .	57
	Return Type . . . . .	57
	Examples . . . . .	57
(70)	ST_MaxM . . . . .	57
	Definition . . . . .	57
	Syntax . . . . .	57
	Return Type . . . . .	57
	Examples . . . . .	57
(71)	ST_MaxX . . . . .	58
	Definition . . . . .	58
	Syntax . . . . .	58
	Return Type . . . . .	58
	Examples . . . . .	58
(72)	ST_MaxY . . . . .	58
	Definition . . . . .	58
	Syntax . . . . .	59
	Return Type . . . . .	59
	Examples . . . . .	59
(73)	ST_MaxZ . . . . .	59
	Definition . . . . .	59
	Syntax . . . . .	59
	Return Type . . . . .	59
	Examples . . . . .	59
(74)	ST_MinM . . . . .	60
	Definition . . . . .	60
	Syntax . . . . .	60
	Return Type . . . . .	60
	Examples . . . . .	60
(75)	ST_MinX . . . . .	60
	Definition . . . . .	60
	Syntax . . . . .	60
	Return Type . . . . .	60
	Examples . . . . .	60

(76) ST_MinY . . . . .	61
Definition . . . . .	61
Syntax . . . . .	61
Return Type . . . . .	61
Examples . . . . .	61
(77) ST_MinZ . . . . .	61
Definition . . . . .	61
Syntax . . . . .	62
Return Type . . . . .	62
Examples . . . . .	62
(78) ST_NumGeometries . . . . .	62
Definition . . . . .	62
Syntax . . . . .	62
Return Type . . . . .	62
Examples . . . . .	62
(79) ST_NumInteriorRing . . . . .	63
Definition . . . . .	63
Syntax . . . . .	63
Return Type . . . . .	63
Examples . . . . .	63
(80) ST_NumPoints . . . . .	63
Definition . . . . .	63
Syntax . . . . .	63
Return Type . . . . .	64
Examples . . . . .	64
(81) ST_Overlaps . . . . .	64
Definition . . . . .	64
Syntax . . . . .	64
Return Type . . . . .	64
Examples . . . . .	64
(82) ST_Point . . . . .	65
Definition . . . . .	65
Syntax . . . . .	65
Return Type . . . . .	65
(83) ST_PointN . . . . .	65
Definition . . . . .	65
Syntax . . . . .	65
Return Type . . . . .	65
Examples . . . . .	65

(84)	ST_PointZ	65
	Definition	66
	Syntax	66
	Return Type	66
(85)	ST_Relate	66
	Definition	66
	Syntax	66
	Return Type	66
	Examples	66
(86)	ST_SetSRID	67
	Definition	67
	Syntax	67
	Return Type	67
(87)	ST_Simplify	67
	Definition	67
	Syntax	67
	Return Type	67
(88)	ST_StartPoint	68
	Definition	68
	Syntax	68
	Return Type	68
	Examples	68
(89)	ST_SymmetricDiff	68
	Definition	68
	Syntax	68
	Return Type	68
	Examples	68
(90)	ST_Touches	69
	Definition	69
	Syntax	69
	Return Type	69
	Examples	69
(91)	ST_Transform	70
	Definition	70
	Syntax	70
	Return Type	70
	Examples	70
(92)	ST_Union	70
	Definition	70

	Syntax . . . . .	71
	Return Type . . . . .	71
	Examples . . . . .	71
(93)	ST_Union . . . . .	71
	Definition . . . . .	71
	Syntax . . . . .	71
	Return Type . . . . .	71
	Examples . . . . .	71
(94)	ST_Within . . . . .	72
	Definition . . . . .	72
	Syntax . . . . .	72
	Return Type . . . . .	72
	Examples . . . . .	72
(95)	ST_X . . . . .	72
	Definition . . . . .	72
	Syntax . . . . .	73
	Return Type . . . . .	73
	Examples . . . . .	73
(96)	ST_Y . . . . .	73
	Definition . . . . .	73
	Syntax . . . . .	73
	Return Type . . . . .	73
	Examples . . . . .	73
(97)	ST_Z . . . . .	73
	Definition . . . . .	74
	Syntax . . . . .	74
	Return Type . . . . .	74
	Examples . . . . .	74

## GIS Extensions for Dremio - SQL Function Reference

### Author

Brian Holman

## Legal Disclaimer

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 screenshot displays the Dremio SQL Editor interface. At the top, there is a search bar labeled "Search Spaces and Datasets". Below it, a toolbar contains buttons for "New Query", "Data", "Preview", and "Run". The main area is the "SQL Editor", which contains the following SQL query:

```
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'
```

To the right of the SQL Editor is a "Functions" panel with a search bar labeled "Search functions...". It lists various GIS functions, including:

- 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

At the bottom of the interface, there is a "Column filter" section showing two fields: "place\_wkt" and "area\_in\_acres". The "area\_in\_acres" field is selected, and its value is displayed as "28235.74996942792".

Figure 1: DAC with GIS extensions

## Third-Party Libraries

The **GIS Extensions** allow Dremio to perform standard GIS functions within Dremio SQL with 72 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) H3\_AsText

#### Definition

Returns a Hex representation of the H3 value as a string.

#### Syntax

H3\_AsText(bigint h3Value)

#### Return Type

string

#### Examples

Query	Result
SELECT H3_AsText(H3_FromGeomPoint(ST_Point(40.4168, -3.7038), 4))	'847b59dfffffffff'

### (2) H3\_Boundary

#### Definition

Returns a polygon geography representing the H3 cell.

### Syntax

H3\_Boundary(bigint h3Value)

### Return Type

binary

### Examples

Query	Result
SELECT ST_AsText(H3_Boundary(H3_FromLongLat(40.4168, -3.7038, 15)))	'POLYGON ((-3.703802360352346 40.41680913267208, -3.7038075007518416 40.41680558484906, -3.703806130063667 40.41680018598506, -3.7037996189769617 40.41679833494421, -3.7037944785779335 40.41680188276699, -3.7037958492651386 40.416807281630845, -3.703802360352346 40.41680913267208))'

## (3) H3\_Center

### Definition

Returns the center of the H3 cell as a point. It will throw an error if the *h3Value* is not valid as an H3 Value.

### Syntax

H3\_Center(bigint h3Value)

### Return Type

binary

### Examples



Query	Result
SELECT ST_AsText(H3_Center(H3_FromText('847b59dfffffffff')))	'POINT (40.305476423174326 -3.743203325561687)'

## (4) H3\_Compact

### Definition

Returns an array with the indexes of a set of hexagons across multiple resolutions that represent the same area as the input set of hexagons.

### Syntax

H3\_Compact(bigint h3Value)

### Return Type

bigint[]

### Examples

Query	Result
SELECT H3_Compact(H3_Uncompact(H3_Wrap(H3_FromText('847b59dfffffffff')),5))	[596645165859340300]

## (5) H3\_Distance

### Definition

Returns the grid distance between two hexagon indexes. This function may fail to find the distance between two indexes if they are very far apart or on opposite sides of a pentagon.

### Syntax

H3\_Distance(bigint h3Value1, bigint h3Value2)

### Return Type

numeric

### Examples

Query	Result
SELECT H3_Distance(H3_FromText('847b591fffffffff'), H3_FromText('847b59bfffffffff'))	1

## (6) H3\_FromGeomPoint

### Definition

Returns the H3 cell index that the point belongs to in the required *resolution*. It will return **null** for non-point geometry and throw an error for resolution outside the valid range [0,15].

### Syntax

H3\_FromGeomPoint(binary pointGeom, number resolution)

### Return Type

bigint

### Examples

Query	Result
SELECT H3_AsText(H3_FromGeomPoint(ST_Point(40.4168, -3.7038), 4))	'847b59dfffffffff'

## (7) H3\_FromGeomPoly

### Definition

Returns the smallest H3 cell that completely encloses the polygon with a dynamic resolution. It will return `null` if there isn't a value to fit the polygon and throw an error if a non-polygon geometry is supplied.

### Syntax

`H3_FromGeomPoly(binary polyGeom)`

### Return Type

`bigint`

### Examples

Query	Result
<pre>SELECT H3_FromGeomPoly(ST_GeomFromText('POLYGON ((-111.69808887205492 40.238737891364416, -111.69837535651395 40.23873740478143, -111.6983745544178 40.2384602506758, -111.69808807598443 40.23846160818078, -111.69808887205492 40.238737891364416))',true))</pre>	<pre>622175171754917887</pre>

## (8) H3\_FromLongLat

### Definition

Returns the H3 cell index specified by *lon* and *lat* at the specified *resolution*. It will throw an error for resolution outside the valid range [0,15].

### Syntax

`H3_FromLongLat(number lon, number lat, number resolution)`

### Return Type

`bigint`

## Examples

Query	Result
SELECT H3_AsText(H3_FromLongLat(40.4168, -3.7038, 4))	'84390cbfffffffff'

## (9) H3\_FromText

### Definition

Converts from String representation of H3 cell value to the bigint representation. It will throw an error if the hex representation is not valid as an H3 Value.

### Syntax

H3\_FromText(string h3Text)

### Return Type

bigint

## Examples

Query	Result
SELECT ST_AsText(H3_Center(H3_FromText('847b59dfffffffff')))	'POINT (40.305476423174326 -3.743203325561687)'

## (10) H3\_HexRing

### Definition

Returns all cell indexes in a hollow hexagonal ring centered at the origin in no particular order. Unlike H3\_Kring, this function will throw an exception if there is a pentagon anywhere in the ring.

### Syntax

H3\_HexRing(bigint h3Origin, int ringSize)

### Return Type

bigint[]

### Examples

Query	Result
SELECT H3_Hexring(H3_FromText('837b59ffffffff'), 1)	[592141849699811300,592141506102427600,592141712260857900,59212487

## (11) H3\_IsPentagon

### Definition

Returns *true* if given H3 index is a pentagon. Returns *false* otherwise, even on invalid input.

### Syntax

H3\_IsPentagon(bigint h3Value)

### Return Type

boolean

### Examples

Query	Result
SELECT H3_IsPentagon(H3_FromText('837b59ffffffff'))	false
SELECT H3_IsPentagon(H3_FromText('8075ffffffff'))	true

## (12) H3\_IsValid

### Definition

Returns *true* when the given index is valid, *false* otherwise.

### Syntax

H3\_IsValid(bigint h3Value)

### Return Type

boolean

### Examples

Query	Result
SELECT H3_IsValid(8675309)	false
SELECT H3_IsValid(H3_FromText('837b59fffffffff'))	true

## (13) H3\_KRing

### Definition

Returns all cell indexes in a filled hexagonal k-ring centered at the origin in no particular order.

### Syntax

H3\_KRing(bigint h3Origin, int ringSize)

### Return Type

bigint[]

### Examples

Query	Result
SELECT H3_KRing(H3_FromText('837b59fffffffff'), 1)	[592141574821904400,592141506102427600,592141712260857900,59212487

## (14) H3\_KRing\_Distances

### Definition

Returns all cell indexes and their distances in a filled hexagonal k-ring centered at the origin in no particular order.

### Syntax

```
H3_KRing_Distances(bigint h3Origin, int ringSize)
```

### Return Type

```
struct{index, distance}
```

### Examples

Query	Result
<pre>SELECT H3_KRingDistances(H3_FromText('837b59fffffffff'), 1)</pre>	<pre>[{"index":592141574821904383,"distance":0}, {"index":59214150610242</pre>

## (15) H3\_Polyfill

### Definition

Returns an array with all the H3 cell indexes for the given polygon or multipolygon including automatically handling the inner holes.

### Syntax

```
H3_Polyfillbinary geometry, number resolution
```

### Return Type

```
bigint[]
```

### Examples

Query	Result
SELECT H3_Polyfill(ST_GeomFromText('POLYGON ((30 10, 40 40, 20 40, 10 20, 30 10))',false), 1)	[582059465512058900,582072659651592200,582068261605081100,58208145

## (16) H3\_Resolution

### Definition

Returns the H3 cell resolution as an integer. It will throw an error if the *h3Value* is not valid as an H3 Value.

### Syntax

H3\_Resolution(bigint h3Value)

### Return Type

integer

### Examples

Query	Result
SELECT H3_Resolution(H3_FromText('847b59dfffffff'))	4

## (17) H3\_ToCenterChild

### Definition

Returns the center child (finer) index for the given hexagon at the given resolution.

### Syntax

H3\_ToCenterChild(bigint h3Value, number childResolution)

### Return Type

bigint



## Examples

Query	Result
<pre>SELECT H3_AsText(H3_ToCenterChild(H3_FromText('837b59ffffffff'), 4))</pre>	<pre>'847b591fffffffff'</pre>

## (18) H3\_ToChildren

### Definition

Returns an array with the indexes of the children/descendents of the given hexagon at the given resolution.

### Syntax

H3\_ToChildren(bigint h3Value, number childResolution)

### Return Type

bigint[]

## Examples

Query	Result
<pre>WITH H3Children AS (SELECT FLATTEN(H3_ToChildren(H3_FromText('837b59ffffffff'), 4)) AS H3Values) SELECT H3_AsText(H3Values) FROM H3Children</pre>	<pre>'847b591fffffffff' '847b593fffffffff' '847b595fffffffff' '847b597fffffffff' '847b599fffffffff' '847b59bfffffffff' '847b59dfffffffff'</pre>

## (19) H3\_ToParent

### Definition

Returns the H3 cell index of the parent of the given hexagon at the given resolution.

### Syntax

H3\_ToParent(bigint h3Value, number resolution)

### Return Type

bigint

### Examples

Query	Result
SELECT H3_AsText(H3_ToParent(H3_FromText('847b59dffffffff'), 3))	'837b59fffffffff'

## (20) H3\_Uncompact

### Definition

Returns an array with the indexes of a set of hexagons of the same *resolution* that represent the same area as the compacted input hexagons.

### Syntax

H3\_Uncompact(bigint h3Value, number resolution)

### Return Type

bigint[]

### Examples

Query	Result
SELECT H3_Uncompact(H3_Wrap(H3_FromText('847b59dffffffff'),5)	[601148757970518000,601148759044259800,601148760118001700,60114876

## (21) H3\_Wrap

### Definition

Takes a single H3 value and wraps it in a list

### Syntax

H3\_Wrap(bigint h3Value)

### Return Type

bigint[]

### Examples

Query	Result
SELECT H3_Wrap(H3_FromText('847b59dfffffffff'))[0]	596645165859340287

## (22) ST\_AggrConvexHull

### Definition

Creates a single geometry that is a convex hull of a geometry that resulted from a union of all aggregate input geometries.

### Syntax

ST\_AggrConvexHull(binary geometry)

### Return Type

binary

### Examples

Query	Result
WITH GEOMLIST AS (SELECT ST_GeomFromText('polygon ((40 40, 40 60, 60 60, 60 40, 40 40))', true) AS GEOM1, ST_GeomFromText('polygon ((20 30, 30 30, 30 40, 20 40, 20 30))', true) AS GEOM2) SELECT ST_AsText(ST_AggrConvexHull(GEOM)) FROM GEOMLIST UNPIVOT ("GEOM" for "COL" in (GEOM1, GEOM2))	'POLYGON ((20 30, 30 30, 60 40, 60 60, 40 60, 20 40, 20 30))'

## (23) ST\_AggrIntersection

### Definition

Returns a single geometry that is an intersection of all aggregate input geometries.

### Syntax

ST\_AggrIntersection(binary geometry)

### Return Type

binary

### Examples

Query	Result
WITH GEOMLIST AS (SELECT ST_GeomFromText('polygon ((5 5, 12 5, 12 10, 5 10, 5 5))', true) AS GEOM1, ST_GeomFromText('polygon ((10 8, 14 8, 14 15, 10 15, 10 8))', true) AS GEOM2, ST_GeomFromText('polygon ((6 8, 20 8, 20 20, 6 20, 6 8))', true) AS GEOM3) SELECT ST_AsText(ST_AggrIntersection(GEOM)) FROM GEOMLIST UNPIVOT ("GEOM" for "COL" in (GEOM1, GEOM2, GEOM3))	'POLYGON ((10 8, 12 8, 12 10, 10 10, 10 8))'

## (24) ST\_AggrUnion

### Definition

Returns a single geometry that is the union of all aggregate input geometries.

### Syntax

ST\_AggrUnion(binary geometry)

### Return Type

binary

### Examples

Query	Result
WITH GEOMLIST AS (SELECT ST_GeomFromText('polygon ((20 30, 30 30, 30 40, 20 40, 20 30))', true) AS GEOM1, ST_GeomFromText('polygon ((40 40, 40 60, 60 60, 60 40, 40 40))', true) AS GEOM2) SELECT ST_AsText(ST_AggrUnion(GEOM)) FROM GEOMLIST UNPIVOT ("GEOM" for "COL" in (GEOM1, GEOM2))	'MULTIPOLYGON (((20 30, 30 30, 30 40, 20 40, 20 30)), ((40 40, 60 40, 60 60, 40 60, 40 40)))'

## (25) ST\_Area

### Definition

Returns the area of polygon or multipolygon

### Syntax

ST\_Area(binary geometry)

### Return Type

number

### Examples

Query	Result
SELECT ST_Area(ST_GeomFromText('POLYGON ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))',true))	24.0

## (26) ST\_AsGeoJSON

### Definition

Returns the GeoJSON representation of *geometry*.

### Syntax

ST\_AsGeoJSON(binary geometry)

### Return Type

string

### Examples

Query	Result
SELECT ST_AsGeoJSON(ST_Point(1, 2))	'{"type":"Point","coordinates":[1,2],"crs":{"type":"name","properties":{}}}'

## (27) ST\_AsText

### Definition

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

### Syntax

ST\_AsText(binary geometry)

### Return Type

string

## Examples

Query	Result
SELECT ST_AsText(ST_Point(1, 2))	'POINT (1 2)'

## (28) ST\_Boundary

### Definition

Returns the closure of the combinatorial boundary of this Geometry.

### Syntax

ST\_Boundary(binary geometry)

### Return Type

binary

## Examples

Query	Result
SELECT ST_AsText(ST_Boundary(ST_GeomFromText('LINESTRING (0 1, 1 0)',true)))	'MULTIPOINT ((0 1), (1 0))'
SELECT ST_AsText(ST_Boundary(ST_GeomFromText('POLYGON ((1 1, 4 1, 1 4), (1 1, 4 1, 1 4))',true)))	'MULTILINESTRING ((1 1, 4 1, 1 4, 1 1))'

## (29) 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

## (30) 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)',true)))	'POINT(2 3)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('MULTIPOINT ((0 0), (1 1), (1 -1), (6 0))',true)))	'POINT(2 0)'



Query	Result
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('linestring (0 0, 6 0)',true)))	'POINT(3 0)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((0 0, 0 8, 8 8, 8 0, 0 0))',true)))	'POINT(4 4)'
SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((1 1, 5 1, 3 4))',true)))	'POINT(3 2)'

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

### (32) ST\_ConvexHull

### Definition

Computes the convex hull of *geometry*. The convex hull is the smallest convex geometry that encloses all geometries in the input. One can think of the convex hull as the geometry obtained by wrapping an rubber band around a set of geometries.

### Syntax

ST\_ConvexHull(binary geometry)

### Return Type

binary

### Examples

Query	Result
<pre>SELECT ST_AsText(ST_ConvexHull(ST_GeomFromText('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))',true)))</pre>	<pre>'POLYGON ((0 0, 8 0, 0 8, 0 0))'</pre>

## (33) 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)',true))	3

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

## (35) ST\_Densify

**Definition**

Densifies a MultiPath (polygons and polylines) *geometry* by *maxLength* so that no segments are longer than given threshold value.

**Syntax**

ST\_Densify(binary geometry, number maxLength)

**Return Type**

binary

**Examples**

Query	Result
SELECT ST_AsText(ST_Densify(ST_GeomFromText('POLYGON ((0 0, 8 0, 0 8, 0 0))',true),4))	'POLYGON ((0 0, 4 0, 8 0, 5.333 2.667, 2.667 5.333, 0 8, 0 4, 0 0))'

**(36) 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)',true), 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))',true), ST_GeomFromText('POLYGON ((0 0, 0 5, 5 5, 5 0))',true)))	'POLYGON ((5 0, 10 0, 10 10, 0 10, 0 5, 5 5, 5 0))'

### (37) 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)',true))	1
SELECT ST_Dimension(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))',true))	2

### (38) 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)',true), ST_GeomFromText('LINESTRING (1 1, 1 0)',true))	true
SELECT ST_Disjoint(ST_GeomFromText('LINESTRING (0 0, 1 1)',true), ST_GeomFromText('LINESTRING (1 0, 0 1)',true))	false

**(39) 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

## (40) 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))',true), 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))',true), ST_Point (10.02,20.01), 100)	false

## (41) ST\_EndPoint

### Definition

Returns the last point of a Linestring.

### Syntax

ST\_EndPoint(binary geometry)

**Return Type**

binary

**Examples**

Query	Result
<pre>SELECT ST_AsText(ST_EndPoint(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)',true)))</pre>	'POINT(3.0 2.2)'

**(42) 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
<pre>SELECT ST_AsText(ST_Envelope(ST_GeomFromText('LINESTRING (0 0, 2 2)'),true)))</pre>	'POLYGON ((0 0, 2 0, 2 2, 0 2, 0 0))'
<pre>SELECT ST_AsText(ST_Envelope(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0)'),true)))</pre>	'POLYGON ((2 0, 3 0, 3 3, 2 3, 2 0))'

**(43) ST\_EnvIntersects**



**Definition**

Returns true if the envelopes of *geometry1* and *geometry2* intersect, otherwise returns false.

**Syntax**

ST\_EnvIntersects(binary geometry1, binary geometry2)

**Return Type**

boolean

**Examples**

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

**(44) 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)',true),ST_GeomFromText('LINESTRING (1 1, 0 0)',true))	true
SELECT ST_Equals(ST_GeomFromText('LINESTRING (0 0, 1 1)',true),ST_GeomFromText('LINESTRING (1 0, 0 1)',true))	false
SELECT ST_Equals(ST_GeomFromText('LINESTRING (0 0, 3 3)',true),ST_GeomFromText('LINESTRING (3 3, 2 2, 1 1, 0 0)',true))	true

## (45) 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 1))',true)))	'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))',true)))	'LINESTRING (0 0, 8 0, 0 8, 0 0)'

## (46) 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
SELECT ST_AsText(ST_Generalize(ST_GeomFromText('POLYGON ((0 0, 1 1, 2 0, 3 2, 4 1, 5 0, 5 10, 0 10))',true), 2, true))	'POLYGON ((0 0, 5 0, 5 10, 0 10, 0 0))'

## (47) 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))',true))/4047 AS utah_acreage</pre>	5.416484897473004E7

## (48) 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 ST_GeodesicLengthWGS84(ST_GeomFromText('MultiLineString((0.0 80.0, 0.3 80.4))', true, 4326))</pre>	45026.96274781222

## (49) 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
SELECT ST_AsText(ST_GeometryN(ST_GeomFromText('MULTIPOINT (10 40, 40 30, 20 20, 30 10)',true), 3))	'POINT (20 20)'
SELECT ST_AsText(ST_GeometryN(ST_GeomFromText('MULTILINESTRING ((2 4, 10 10), (20 20, 7 8))',true), 2))	'LINESTRING (20 20, 7 8)'

## (50) 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)',true))	'ST_LINESTRING'
SELECT ST_GeometryType(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))',true))	'ST_POLYGON'

## (51) 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 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)'

## (52) ST\_GeomFromGeoJSON

### Definition

Constructs a geometry from GeoJSON.

### Syntax

ST\_GeomFromGeoJSON(string geoJsonString)

### Return Type

binary

### Examples

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

## (53) ST\_GeomFromText

### Definition

Takes a well-known text representation and returns a geometry object. Set *ignoreErrors* to *true* to ignore bad data or *false* to fail and show the bad WKT value.

### Syntax

ST\_GeomFromText(string wktString, boolean ignoreErrors)

### Return Type

binary

---

## (54) ST\_GeomFromText

**Definition**

Takes a well-known text representation and a spatial reference ID and returns a geometry object. Set *ignoreErrors* to *true* to ignore bad data or *false* to fail and show the bad WKT value.

**Syntax**

ST\_GeomFromText(string wktString, boolean ignoreErrors, number SRID)

**Return Type**

binary

---

**(55) ST\_GeomFromWKB****Definition**

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

**Syntax**

ST\_GeomFromWKB(binary wkbValue)

**Return Type**

binary

---

**(56) 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

---

**(57) ST\_GeoSize****Definition**

Takes a geometry object and returns its size in bytes.

**Syntax**

ST\_GeoSize(binary geometry)

**Return Type**

number

---

**(58) 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))',true), 1))	'LINESTRING (1 1, 1 5, 5 1, 1 1)'

## (59) 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)',true), ST_GeomFromText('LINESTRING(0 3, 0 1, 1 0, 3 0)',true)))	'MULTILINESTRING ((1 0, 2 0), (0 2, 0 1))'
SELECT ST_AsText(ST_Intersection(ST_GeomFromText('POLYGON ((2 0, 2 3, 3 0))',true), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))',true)))	'POLYGON ((2 1, 2.6666666666666667 1, 2 3, 2 1))'

## (60) 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)',true), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))',true))	true
SELECT ST_Intersects(ST_GeomFromText('LINESTRING (8 7, 7 8)',true), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))',true))	false

**(61) 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))',true))	false
SELECT ST_Is3D(ST_GeomFromText('LINESTRING (0 0, 3 4, 0 4, 0 0)',true))	false
SELECT ST_Is3D(ST_Point(3, 4))	false
SELECT ST_Is3D(ST_PointZ(3, 4, 2))	true

## (62) 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))	true
SELECT ST_IsClosed(ST_GeomFromText('LINESTRING(0 0, 3 4)',true))	false

## (63) 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))	true

**(64) 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))	true
SELECT ST_IsMeasured(ST_GeomFromText('POINT ZM (1 1 5 60)',true))	true

## (65) 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))	true
SELECT ST_IsRing(ST_GeomFromText('LINESTRING (0 0, 1 1, 1 2, 2 1, 1 1, 0 0)',true))	false
SELECT ST_IsRing(ST_GeomFromText('LINESTRING (0 0, 3 4)',true))	false

## (66) 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)',true))	false

## (67) ST\_JSONPath

### Definition

Extract a portion of *jsonData* as a string by following the specified path in the JSON Object from *jsonPath*. **Example JSON Path Syntax (similar to XPath for XML):** `‘/data[Array]’ ‘/data[Array][1]/id[String]’ ‘/data[Array][1]/likes[Object]’ ‘/data[Array][1]/likes[Object]/summary[Object]/total_count[String]’ ‘/data[Array][3]’ ‘/data[Array][id=131272076894593_1420960724592382]/likes[Object]/summary[Object]/total_count’ ‘fbids[String]’ ‘quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/enterpriseValue[Object]/fmt[String]’ ‘quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/result[4][0]/defaultKeyStatistics[6]/sharesOutstanding[6]/raw[1]’ ‘quoteSummary[6]/result[Array]’ ‘quoteSummary[6]/result[Array][0]’ ‘quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/lastSplitDate[Object]/raw1[Long]’ ‘quoteSummary[Object]/result[Array][0]/defaultKeyStatistics[Object]/result[Array][0]/defaultKeyStatistics[Object]/result[4][0]/defaultKeyStatistics[6]/sharesOutstanding[6]/raw[1]’`

### Syntax

ST\_JSONPath(string jsonPath, string jsonData)

### Return Type

string

## Examples

Query	Result
SELECT ST_JSONPath('/coordinates[Array][0]',ST_AsGeoJSON(ST_Envelope(ST_GeomFromText('LINESTRING (-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)'),true)))	'[[[-114.0500520000997,36.99897700038832],[-109.04517199998776,36.99897700038832],[-109.05002599989996,41.000691000389395],[-111.04681499981234,40.997875000031286],[-111.04671399965133,42.00170200004732],[-114.0500520000997,42.00170200004732],[-114.0500520000997,36.99897700038832]]]'

## (68) 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)',true))	5.0
SELECT ST_Length(ST_GeomFromText('MULTILINESTRING ((1 0, 2 0), (0 2, 0 1))',true))	2.0

## (69) 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)',true))	80.0
SELECT ST_M(ST_GeomFromText('POINT ZM (1 1 5 60)',true))	60.0

**(70) 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)',true))	2.0
SELECT ST_MaxM(ST_GeomFromText('POINT M (1.5 2.5 3)',true))	3.0

## (71) 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)',true))	3.0
SELECT ST_MaxX(ST_GeomFromText('POINT M (1.5 2.5 3)',true))	1.5

## (72) 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)',true))	2.5
SELECT ST_MaxY(ST_GeomFromText('POINT M (1.5 2.5 3)',true))	2.5

**(73) 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)',true))	2.0
SELECT ST_MaxZ(ST_GeomFromText('LINESTRING Z (1.5 2.5 3, 3.0 2.2 4)',true))	4.0

## (74) 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)',true))	1.0
SELECT ST_MinM(ST_GeomFromText('POINT M (1.5 2.5 3)',true))	3.0

## (75) 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)',true))	1.25
SELECT ST_MinX(ST_GeomFromText('POINT M (1.75 2.5 3)',true))	1.75

## (76) 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)',true))	2.2
SELECT ST_MinY(ST_GeomFromText('POINT M (1.5 2.25 3)',true))	2.25

## (77) 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)',true))	1.0
SELECT ST_MinZ(ST_GeomFromText('LINESTRING Z (1.5 2.5 3, 3.0 2.2 4)',true))	3.0

**(78) 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))',true))	4

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

## (79) 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))',true))	1

## (80) 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)',true))	2
SELECT ST_NumPoints((ST_GeomFromText('POLYGON ((0 0, 10 0, 0 10, 0 0))',true)))	4

## (81) 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))',true), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))',true))	true
SELECT ST_Overlaps(ST_GeomFromText('POLYGON ((2 0, 2 1, 3 1))',true), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))',true))	false



## (82) ST\_Point

### Definition

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

### Syntax

ST\_Point(number lon, number lat)

### Return Type

binary

---

## (83) 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
<pre>SELECT ST_AsText(ST_PointN(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)',true), 2))</pre>	<pre>'POINT (3 2.2)'</pre>

## (84) 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

---

## (85) 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))',true), ST_GeomFromText('POLYGON ((1 1, 1 4, 4, 4 1))',true), '****T****')	true
SELECT ST_Relate(ST_GeomFromText('POLYGON ((2 0, 2 1, 3 1))',true), ST_GeomFromText('POLYGON ((1 1, 1 4, 4, 4 1))',true), 'T*****')	false

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

## (86) ST\_SetSRID

### Definition

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

### Syntax

ST\_SetSRID(binary geometry, number SRID)

### Return Type

binary

---

## (87) 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

---

## (88) ST\_StartPoint

### Definition

Returns the first point of a Linestring.

### Syntax

ST\_StartPoint(binary geometry)

### Return Type

binary

### Examples

Query	Result
<pre>SELECT ST_AsText(ST_StartPoint(ST_GeomFromText('LINESTRING (1.5 2.5, 3.0 2.2)',true)))</pre>	<pre>'POINT(1.5 2.5)'</pre>

## (89) 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)',true), ST_GeomFromText('LINESTRING (1 2, 3 2)',true)))	'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))',true), ST_GeomFromText('POLYGON ((1 1, 3 1, 3 3, 1 3, 1 1))',true)))	'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)))'

## (90) 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))	true
SELECT ST_Touches(ST_Point(8, 8), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))',true))	false

## (91) 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
<pre>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))', true), 4326, 3857))</pre>	<pre>'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))'</pre>

## (92) 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))',true), ST_GeomFromText('POLYGON ((4 1, 4 4, 4 8, 8 1))',true)))	'POLYGON ((1 1, 4 1, 8 1, 4 8, 4 4, 1 4, 1 1))'

**(93) ST\_Union****Definition**

Returns a geometry as the union of the supplied geometry.

**Syntax**

ST\_Union(binary geometry)

**Return Type**

binary

**Examples**

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

## (94) 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))	true
SELECT ST_Within(ST_Point(8, 8), ST_GeomFromText('POLYGON ((1 1, 1 4, 4 4, 4 1))',true))	false

## (95) 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

**(96) 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)',true))	7.0

**(97) 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)',true))	9.0