

GIS Extensions for Dremio - SQL Functions Reference

Brian K. Holman

Abstract

Complete Reference to the 72 GIS-related functions in the *GIS Extensions for Dremio* implementation including syntax, return type, and examples.

Contents

GIS Extensions for Dremio - SQL Function Reference	13
Author	13
Legal Disclaimer	13
Third-Party Libraries	13
(1) H3_AsText	15
Definition	15
Syntax	15
Return Type	15
Examples	15
(2) H3_Boundary	15
Definition	15
Syntax	15
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	17
Examples	17
(6) H3_FromGeomPoint	18
Definition	18
Syntax	18
Return Type	18
Examples	18
(7) H3_FromLongLat	18
Definition	18
Syntax	18
Return Type	19
Examples	19
(8) H3_FromText	19
Definition	19
Syntax	19
Return Type	19
Examples	19
(9) H3_HexRing	19
Definition	19
Syntax	20
Return Type	20
Examples	20
(10) H3_IsPentagon	20
Definition	20
Syntax	20
Return Type	20
Examples	20
(11) H3_IsValid	20
Definition	21
Syntax	21
Return Type	21
Examples	21
(12) H3_KRing	21
Definition	21
Syntax	21

	Return Type	21
	Examples	21
(13)	H3_KRing_Distances	22
	Definition	22
	Syntax	22
	Return Type	22
	Examples	22
(14)	H3_Polyfill	22
	Definition	22
	Syntax	22
	Return Type	22
	Examples	23
(15)	H3_Resolution	23
	Definition	23
	Syntax	23
	Return Type	23
	Examples	23
(16)	H3_ToChildren	23
	Definition	23
	Syntax	23
	Return Type	24
	Examples	24
(17)	H3_ToParent	24
	Definition	24
	Syntax	24
	Return Type	24
	Examples	24
(18)	H3_Uncompact	24
	Definition	25
	Syntax	25
	Return Type	25
	Examples	25
(19)	H3_Wrap	25
	Definition	25
	Syntax	25
	Return Type	25
	Examples	25
(20)	ST_AggrConvexHull	26
	Definition	26

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

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

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

Examples	40
(44) ST_Generalize	41
Definition	41
Syntax	41
Return Type	41
Examples	41
(45) ST_GeodesicAreaWGS84	41
Definition	41
Syntax	42
Return Type	42
Examples	42
(46) ST_GeodesicLengthWGS84	42
Definition	42
Syntax	42
Return Type	42
Examples	42
(47) ST_GeometryN	43
Definition	43
Syntax	43
Return Type	43
Examples	43
(48) ST_GeometryType	43
Definition	43
Syntax	44
Return Type	44
Examples	44
(49) ST_GeomFromEWKB	44
Definition	44
Syntax	44
Return Type	44
Examples	44
(50) ST_GeomFromGeoJSON	45
Definition	45
Syntax	45
Return Type	45
Examples	45
(51) ST_GeomFromText	45
Definition	45
Syntax	46

	Return Type	46
(52)	ST_GeomFromText	46
	Definition	46
	Syntax	46
	Return Type	46
(53)	ST_GeomFromWKB	46
	Definition	46
	Syntax	46
	Return Type	47
(54)	ST_GeomFromWKB	47
	Definition	47
	Syntax	47
	Return Type	47
(55)	ST_GeoSize	47
	Definition	47
	Syntax	47
	Return Type	47
(56)	ST_InteriorRingN	48
	Definition	48
	Syntax	48
	Return Type	48
	Examples	48
(57)	ST_Intersection	48
	Definition	48
	Syntax	48
	Return Type	48
	Examples	48
(58)	ST_Intersects	49
	Definition	49
	Syntax	49
	Return Type	49
	Examples	49
(59)	ST_Is3D	50
	Definition	50
	Syntax	50
	Return Type	50
	Examples	50
(60)	ST_IsClosed	50
	Definition	50

	Syntax	50
	Return Type	50
	Examples	51
(61)	ST_IsEmpty	51
	Definition	51
	Syntax	51
	Return Type	51
	Examples	51
(62)	ST_IsMeasured	51
	Definition	51
	Syntax	52
	Return Type	52
	Examples	52
(63)	ST_IsRing	52
	Definition	52
	Syntax	52
	Return Type	52
	Examples	52
(64)	ST_IsSimple	53
	Definition	53
	Syntax	53
	Return Type	53
	Examples	53
(65)	ST_JSONPath	53
	Definition	53
	Syntax	53
	Return Type	54
	Examples	54
(66)	ST_Length	54
	Definition	54
	Syntax	55
	Return Type	55
	Examples	55
(67)	ST_M	55
	Definition	55
	Syntax	55
	Return Type	55
	Examples	55
(68)	ST_MaxM	56

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

(76) ST_NumGeometries	60
Definition	60
Syntax	61
Return Type	61
Examples	61
(77) ST_NumInteriorRing	61
Definition	61
Syntax	61
Return Type	61
Examples	61
(78) ST_NumPoints	62
Definition	62
Syntax	62
Return Type	62
Examples	62
(79) ST_Overlaps	62
Definition	62
Syntax	63
Return Type	63
Examples	63
(80) ST_Point	63
Definition	63
Syntax	63
Return Type	63
(81) ST_PointN	63
Definition	63
Syntax	64
Return Type	64
Examples	64
(82) ST_PointZ	64
Definition	64
Syntax	64
Return Type	64
(83) ST_Relate	64
Definition	64
Syntax	65
Return Type	65
Examples	65
(84) ST_SetSRID	65

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

	Syntax	70
	Return Type	70
	Examples	70
(93)	ST_X	71
	Definition	71
	Syntax	71
	Return Type	71
	Examples	71
(94)	ST_Y	71
	Definition	71
	Syntax	71
	Return Type	72
	Examples	72
(95)	ST_Z	72
	Definition	72
	Syntax	72
	Return Type	72
	Examples	72

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.

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.

Search Spaces and Datasets

New Query Data Preview Run

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

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
<code>SELECT H3_AsText(H3_FromGeomPoint(ST_Point(40.4168, -3.7038), 4))</code>	<code>'847b59dfffffffff'</code>

(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_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'

(8) 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)'

(9) 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

(10) 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('8075ffffffffffff'))	true

(11) 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

(12) 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('837b59ffffffff'), 1)	[592141574821904400,592141506102427600,592141712260857900,59212487

(13) 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
SELECT H3_KRingDistances(H3_FromText('837b59ffffffff'), 1)	[{"index":592141574821904383,"distance":0},{ "index":59214150610242

(14) 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_Polyfill(binary 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

(15) 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('847b59dfffffffff'))	4

(16) 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
WITH H3Children AS (SELECT FLATTEN(H3_ToChildren(H3_FromText('837b59fffffffff'), 4)) AS H3Values) SELECT H3_AsText(H3Values) FROM H3Children	'847b591fffffffff' '847b593fffffffff' '847b595fffffffff' '847b597fffffffff' '847b599fffffffff' '847b59bfffffffff' '847b59dfffffffff'

(17) 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('847b59dfffffffff'), 3))	'837b59fffffffff'

(18) 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

(19) 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
<code>SELECT H3_Wrap(H3_FromText('847b59dfffffff'))[0]</code>	596645165859340287

(20) 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
<code>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))</code>	<code>'POLYGON ((20 30, 30 30, 60 40, 60 60, 40 60, 20 40, 20 30))'</code>

(21) 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))'

(22) 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)))'

(23) 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))'))	24.0

(24) 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":{}}}'

(25) 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)'

(26) 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)'))))	'MULTIPOINT ((0 1), (1 0))'
SELECT ST_AsText(ST_Boundary(ST_GeomFromText('POLYGON ((1 1, 4 1, 1 4, 1 1)) ((1 1, 4 1, 1 4))'))))	'MULTILINESTRING ((1 1, 4 1, 1 4, 1 1))'

(27) 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
<code>SELECT ST_Buffer(ST_Point(0, 0), 1)</code>	polygon approximating a unit circle

(28) 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
<code>SELECT ST_AsText(ST_Centroid(ST_GeomFromText('point (2 3)')))</code>	<code>'POINT(2 3)'</code>
<code>SELECT ST_AsText(ST_Centroid(ST_GeomFromText('MULTIPOINT ((0 0), (1 1), (1 -1), (6 0))')))</code>	<code>'POINT(2 0)'</code>
<code>SELECT ST_AsText(ST_Centroid(ST_GeomFromText('linestring (0 0, 6 0)')))</code>	<code>'POINT(3 0)'</code>
<code>SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((0 0, 0 8, 8 8, 8 0, 0 0))')))</code>	<code>'POINT(4 4)'</code>
<code>SELECT ST_AsText(ST_Centroid(ST_GeomFromText('POLYGON ((1 1, 5 1, 3 4))')))</code>	<code>'POINT(3 2)'</code>

(29) 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

(30) 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
SELECT ST_AsText(ST_ConvexHull(ST_GeomFromText('polygon ((0 0, 8 0, 0 8, 0 0), (1 1, 1 5, 5 1, 1 1))')))	'POLYGON ((0 0, 8 0, 0 8, 0 0))'

(31) 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

(32) 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

(33) 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))'),4))	'POLYGON ((0 0, 4 0, 8 0, 5.333 2.667, 2.667 5.333, 0 8, 0 4, 0 0))'

(34) 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))'

(35) 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

(36) 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

(37) 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

(38) 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

(39) 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)')))</pre>	<pre>'POINT(3.0 2.2)'</pre>

(40) 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))'

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

(42) 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

(43) 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))')))	'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)'

(44) 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))'), 2, true))	'POLYGON ((0 0, 5 0, 5 10, 0 10, 0 0))'

(45) 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
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	5.416484897473004E7

(46) 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
SELECT ST_GeodesicLengthWGS84(ST_GeomFromText('MultiLineString((0.0 80.0, 0.3 80.4))', 4326))	45026.96274781222

(47) 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)'), 3))	'POINT (20 20)'
SELECT ST_AsText(ST_GeometryN(ST_GeomFromText('MULTILINESTRING ((2 4, 10 10), (20 20, 7 8))'), 2))	'LINESTRING (20 20, 7 8)'

(48) 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'

(49) 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)'

(50) 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)'

(51) 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

(52) 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

(53) ST_GeomFromWKB**Definition**

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

Syntax

ST_GeomFromWKB(binary wkbValue)

Return Type

binary

(54) 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

(55) ST_GeoSize**Definition**

Takes a geometry object and returns its size in bytes.

Syntax

ST_GeoSize(binary geometry)

Return Type

number

(56) 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
<pre>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))</pre>	<pre>'LINESTRING (1 1, 1 5, 5 1, 1 1)'</pre>

(57) 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 0))'), ST_GeomFromText('POLYGON ((1 1, 4 1, 4 4, 1 4))'))))	'POLYGON ((2 1, 2.6666666666666667 1, 2 3, 2 1))'

(58) 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

(59) 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

(60) 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

(61) 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

(62) 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

(63) 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

Query	Result
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

(64) 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

(65) 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
<pre>SELECT ST_JSONPath('/coordinates[Array][0]',ST_AsGeoJSON(ST_Envelope(the_geom))) FROM utah_county_taxparcels SELECT ST_JSONPath('/crs[Object]/properties[Object]/name',ST_AsGeoJSON(ST_Envelope(the_geom))) FROM utah_county_taxparcels</pre>	
<pre>' 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]/forwardPE[Object]/raw[Double]' 'quoteSummary[6]/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]/sharesOutstanding[Object]/raw[Integer]'</pre>	

(66) 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

(67) 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

(68) 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

(69) 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

(70) 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

(71) 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

(72) 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

(73) 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

(74) 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

(75) 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

(76) 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

(77) 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

(78) 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

(79) 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

(80) 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

(81) 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)'), 2))</pre>	<pre>'POINT (3 2.2)'</pre>

(82) 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

(83) 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

(84) ST_SetSRID

Definition

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

Syntax

ST_SetSRID(binary geometry, number SRID)

Return Type

binary

(85) 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

(86) 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)'))))</pre>	'POINT(1.5 2.5)'

(87) 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
<pre>SELECT ST_AsText(ST_SymmetricDiff(ST_GeomFromText('LINESTRING (0 2, 2 2)'), ST_GeomFromText('LINESTRING (1 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))')))) --> '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)))'</pre>	'MULTILINESTRING ((0 2, 1 2), (2 2, 3 2))'

(88) 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

(89) 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)) '

(90) 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))'

(91) 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))'), 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))'

(92) 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

(93) 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

(94) 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

(95) 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