PostGIS 2.0 Cheatsheet

New in this release [1] Enhanced in this release [2] Requires GEOS 3.3 or higher[g3.3] 2.5/3D support[3d] SQL-MM[mm] Supports geography [G]

PostgreSQL PostGIS Geometry/Geography/Box Types

box2d A box composed of x min, ymin, xmax, ymax. Often used to return the 2d enclosing box of a geometry.

box3d A box composed of x min, ymin, zmin, xmax, ymax, zmax. Often used to return the 3d extent of a geometry or collection of geometries.

geometry Planar spatial data type.

geometry_dump A spatial datatype with two fields - geom (holding a geometry object) and path[] (a 1-d array holding the position of the geometry within the dumped object.)

geography Ellipsoidal spatial data type.

Management Functions

AddGeometryColumn[2 3d] Adds a geometry column to an existing table of attributes. By default uses type modifier to define rather than constraints. Pass in false for use_typmod to get old check constraint based behavior

```
1. table_name, column_name, srid, type, dimension,
   use_typmod=true
2. schema_name, table_name, column_name, srid, type,
   dimension, use_typmod=true
3. catalog_name, schema_name, table_name, column_name,
   srid, type, dimension, use_typmod=true
```

DropGeometryColumn[3d] Removes a geometry column from a spatial table.

```
1. table_name, column_name
2. schema_name, table_name, column_name
3. catalog_name, schema_name, table_name, column_name
```

DropGeometryTable Drops a table and all its references in geometry_columns.

```
1. table_name
2. schema_name, table_name
3. catalog_name, schema_name, table_name
```

PostGIS_Full_Version () Reports full postgis version and build configuration infos.

PostGIS_GEOS_Version () Returns the version number of the GEOS library.

PostGIS_LibXML_Version () Returns the version number of the libxml2 library.

PostGIS_Lib_Build_Date () Returns build date of the PostGIS library.

PostGIS_Lib_Version () Returns the version number of the PostGIS library.

PostGIS_PROJ_Version () Returns the version number of the PROJ4 library.

PostGIS_Scripts_Build_Date () Returns build date of the PostGIS scripts.

PostGIS_Scripts_Installed () Returns version of the postgis scripts installed in this database.

PostGIS_Scripts_Released () Returns the version number of the postgis.sql script released with the installed postgis lib.

PostGIS_Version () Returns PostGIS version number and compile-time options.

Populate_Geometry_Columns[2] Ensures geometry columns are defined with type modifiers or have appropriate spatial constraints This ensures they will be registered correctly in geometry_columns view. By default will convert all geometry columns with no type modifier to ones with type modifiers. To get old behavior set use_typmod=false

```
1. use_typmod=true
2. relation_oid, use_typmod=true
```

UpdateGeometrySRID[3d] Updates the SRID of all features in a geometry column, geometry_columns metadata and srid. If it was enforced with constraints, the constraints will be updated with new srid constraint. If the old was enforced by type definition, the type definition will be changed.

```
1. table_name, column_name, srid
2. schema_name, table_name, column_name, srid
3. catalog_name, schema_name, table_name, column_name,
   srid
```

Geometry Constructors

ST_BdPolyFromText (WKT, srid) Construct a Polygon given an arbitrary collection of closed linestrings as a MultiLineString Well-Known text representation.

ST_BdMPolyFromText (WKT, srid) Construct a MultiPolygon given an arbitrary collection of closed linestrings as a MultiLineString text representation Well-Known text representation.

ST_GeogFromText[G] (EWKT) Return a specified geography value from Well-Known Text representation or extended (WKT).

Operators

&&[2 G] Returns TRUE if A's 2D bounding box intersects B's 2D bounding box.

```
1. A, B
2. A, B
```

&&&[1 3d] (A, B) Returns TRUE if A's 3D bounding box intersects B's 3D bounding box.

&< (A, B) Returns TRUE if A's bounding box overlaps or is to the left of B's.

&<| (A, B) Returns TRUE if A's bounding box overlaps or is below B's.

&> (A, B) Returns TRUE if A' bounding box overlaps or is to the right of B's.

<< (A, B) Returns TRUE if A's bounding box is strictly to the left of B's.

<<| (A, B) Returns TRUE if A's bounding box is strictly below B's.

=[G] Returns TRUE if A's bounding box is the same as B's. Uses double precision bounding box.

```
1. A, B
2. A, B
```

>> (A, B) Returns TRUE if A's bounding box is strictly to the right of B's.

@ (A, B) Returns TRUE if A's bounding box is contained by B's.

|&> (A, B) Returns TRUE if A's bounding box overlaps or is above B's.

|>> (A, B) Returns TRUE if A's bounding box is strictly above B's.

~ (A, B) Returns TRUE if A's bounding box contains B's.

~= (A, B) Returns TRUE if A's bounding box is the same as B's.

<->[1] (A, B) Returns the distance between two points. For point / point checks it uses floating point accuracy (as opposed to the double precision accuracy of the underlying point geometry). For other geometry types the distance between the floating point bounding box centroids is returned. Useful for doing distance ordering and nearest neighbor limits using KNN gist functionality.

<#>[1] (A, B) Returns the distance between bounding box of 2 geometries. For point / point checks it's almost the same as distance (though may be different since the bounding box is at floating point accuracy and geometries are double precision). Useful for doing distance ordering and nearest neighbor limits using KNN gist functionality.

Spatial Relationships and Measurements

ST_3DClosestPoint[1 3d] (g1, g2) Returns the 3-dimensional point on g1 that is closest to g2. This is the first point of the 3D shortest line.

ST_3DDistance[1 mm 3d] (g1, g2) For geometry type Returns the 3-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units.

ST_3DDWithin[1 mm 3d] (g1, g2, distance_of_srid) For 3d (z) geometry type Returns true if two geometries 3d distance is within number of units.

ST_3DDFullyWithin[1 3d] (g1, g2, distance) Returns true if all of the 3D geometries are within the specified distance of one another.

ST_3DIntersects[1 mm 3d] (geomA, geomB) Returns TRUE if the Geometries "spatially intersect" in 3d - only for points and linestrings

ST_3DLongestLine[1 3d] (g1, g2) Returns the 3-dimensional longest line between two geometries

ST_3DMaxDistance[1 3d] (g1, g2) For geometry type Returns the 3-dimensional cartesian maximum distance (based on spatial ref) between two geometries in projected units.

ST_3DShortestLine[1 3d] (g1, g2) Returns the 3-dimensional shortest line between two geometries

ST_Area[2 mm G] Returns the area of the surface if it is a polygon or multi-polygon. For "geometry" type area is in SRID units. For "geography" area is in square meters.

```
1. g1
2. geog, use_spheroid=true
```

ST_GeographyFromText[G] (EWKT) Return a specified geography value from Well-Known Text representation or extended (WKT).

ST_GeogFromWKB[G] (geom) Creates a geography instance from a Well-Known Binary geometry representation (WKB) or extended Well Known Binary (EWKB).

ST_GeomCollFromText[mm] Makes a collection Geometry from collection WKT with the given SRID. If SRID is not give, it defaults to -1.

```
1. WKT, srid
2. WKT
```

ST_GeomFromEWKB[2 3d] (EWKB) Return a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

ST_GeomFromEWKT[2 3d] (EWKT) Return a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

ST_GeometryFromText[mm] Return a specified ST_Geometry value from Well-Known Text representation (WKT). This is an alias name for ST_GeomFromText

```
1. WKT
2. WKT, srid
```

ST_GeomFromGML[2 3d] Takes as input GML representation of geometry and outputs a PostGIS geometry object

```
1. geomgml
2. geomgml, srid
```

ST_GeomFromGeoJSON[1 3d] (geomjson) Takes as input a geojson representation of a geometry and outputs a PostGIS geometry object

ST_GeomFromKML[3d] (geomkml) Takes as input KML representation of geometry and outputs a PostGIS geometry object

ST_GMLToSQL[2 mm] Return a specified ST_Geometry value from GML representation. This is an alias name for ST_GeomFromGML

```
1. geomgml
2. geomgml, srid
```

ST_GeomFromText[mm] Return a specified ST_Geometry value from Well-Known Text representation (WKT).

```
1. WKT
2. WKT, srid
```

ST_GeomFromWKB[mm] Creates a geometry instance from a Well-Known Binary geometry representation (WKB) and optional SRID.

```
1. geom
2. geom, srid
```

ST_LineFromMultiPoint[3d] (aMultiPoint) Creates a LineString from a MultiPoint geometry.

ST_LineFromText[mm] Makes a Geometry from WKT representation with the given SRID. If SRID is not given, it defaults to -1.

```
1. WKT
2. WKT, srid
```

ST_LineFromWKB[mm] Makes a LINESTRING from WKB with the given SRID

```
1. WKB
2. WKB, srid
```

ST_LinestringFromWKB[mm] Makes a geometry from WKB with the given SRID.

```
1. WKB
2. WKB, srid
```

ST_MakeBox2D (pointLowLeft, pointUpRight) Creates a BOX2D defined by the given point geometries.

ST_3DMakeBox[3d] (point3DLowLeftBottom, point3DUpRightTop) Creates a BOX3D defined by the given 3d point geometries.

ST_MakeLine[1 3d] Creates a Linestring from point or line geometries.

```
1. geoms
2. geom1, geom2
3. geoms_array
```

ST_MakeEnvelope[2] (xmin, ymin, xmax, ymax, srid=unknown) Creates a rectangular Polygon formed from the given minimums and maximums. Input values must be in SRS specified by the SRID.

ST_MakePolygon[3d] Creates a Polygon formed by the given shell. Input geometries must be closed LINESTRINGS.

```
1. linestring
2. outerlinestring, interiorlinestrings
```

ST_Azimuth[2 G] Returns the angle in radians from the horizontal of the vector defined by pointA and pointB. Angle is computed clockwise from down-to-up: on the clock: 12=0; 3=PI/2; 6=PI; 9=3PI/2.

```
1. pointA, pointB
2. pointA, pointB
```

ST_Centroid[mm] (g1) Returns the geometric center of a geometry.

ST_ClosestPoint (g1, g2) Returns the 2-dimensional point on g1 that is closest to g2. This is the first point of the shortest line.

ST_Contains[mm] (geomA, geomB) Returns true if and only if no points of B lie in the exterior of A, and at least one point of the interior of B lies in the interior of A.

ST_ContainsProperly (geomA, geomB) Returns true if B intersects the interior of A but not the boundary (or exterior). A does not contain properly itself, but does contain itself.

ST_Covers[G] Returns 1 (TRUE) if no point in Geometry B is outside Geometry A

```
1. geomA, geomB
2. geogpolyA, geogpointB
```

ST_CoveredBy[G] Returns 1 (TRUE) if no point in Geometry/Geography A is outside Geometry/Geography B

```
1. geomA, geomB
2. geogA, geogB
```

ST_Crosses[mm] (g1, g2) Returns TRUE if the supplied geometries have some, but not all, interior points in common.

ST_LineCrossingDirection (linestringA, linestringB) Given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behavior. 0 is no crossing.

ST_Disjoint[mm] (A, B) Returns TRUE if the Geometries do not "spatially intersect" - if they do not share any space together.

ST_Distance[mm G] For geometry type Returns the 2-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units. For geography type defaults to return spheroidal minimum distance between two geographies in meters.

```
1. g1, g2
2. gg1, gg2
3. gg1, gg2, use_spheroid
```

ST_HausdorffDistance Returns the Hausdorff distance between two geometries. Basically a measure of how similar or dissimilar 2 geometries are. Units are in the units of the spatial reference system of the geometries.

```
1. g1, g2
2. g1, g2, densifyFrac
```

ST_MaxDistance (g1, g2) Returns the 2-dimensional largest distance between two geometries in projected units.

ST_Distance_Sphere (geomlonlatA, geomlonlatB) Returns minimum distance in meters between two lon/lat geometries. Uses a spherical earth and radius of 6370986 meters. Faster than ST_Distance_Spheroid , but less accurate. PostGIS versions prior to 1.5 only implemented for points.

ST_Distance_Spheroid (geomlonlatA, geomlonlatB, measurement_spheroid) Returns the minimum distance between two lon/lat geometries given a particular spheroid. PostGIS versions prior to 1.5 only support points.

ST_DFullyWithin (g1, g2, distance) Returns true if all of the geometries are within the specified distance of one another

ST_DWithin[G] Returns true if the geometries are within the specified distance of one another. For geometry units are in those of spatial reference and For geography units are in meters and measurement is defaulted to use_spheroid=true (measure around spheroid), for faster check, use_spheroid=false to measure along sphere.

```
1. g1, g2, distance_of_srid
2. gg1, gg2, distance_meters
3. gg1, gg2, distance_meters, use_spheroid
```

ST_Equals[mm] (A, B) Returns true if the given geometries represent the same geometry. Directionality is ignored.

ST_HasArc[3d] (geomA) Returns true if a geometry or geometry collection contains a circular string

Geometry Accessors

GeometryType[2 3d] (geomA) Returns the type of the geometry as a string. Eg: 'LINESTRING', 'POLYGON', 'MULTIPOINT', etc.

ST_Boundary[mm 3d] (geomA) Returns the closure of the combinatorial boundary of this Geometry.

ST_CoordDim[mm 3d] (geomA) Return the coordinate dimension of the ST_Geometry value.

ST_Dimension[2 mm] (g) The inherent dimension of this Geometry object, which must be less than or equal to the coordinate dimension.

ST_EndPoint[mm 3d] (g) Returns the last point of a LINESTRING geometry as a POINT.

ST_Envelope[mm] (g1) Returns a geometry representing the double precision (float8) bounding box of the supplied geometry.

ST_ExteriorRing[mm 3d] (a_polygon) Returns a line string representing the exterior ring of the POLYGON geometry. Return NULL if the geometry is not a polygon. Will not work with MULTIPOLYGON

ST_GeometryN[2 mm 3d] (geomA, n) Return the 1-based Nth geometry if the geometry is a GEOMETRYCOLLECTION, (MULTI)POINT, (MULTI)LINESTRING, MULTICURVE or (MULTI)POLYGON, POLYHEDRALSURFACE Otherwise, return NULL.

ST_GeometryType[2 mm 3d] (g1) Return the geometry type of the ST_Geometry value.

ST_InteriorRingN[mm 3d] (a_polygon, n) Return the Nth interior linestring ring of the polygon geometry. Return NULL if the geometry is not a polygon or the given N is out of range.

ST_IsClosed[2 mm 3d] (g) Returns TRUE if the LINESTRING's start and end points are coincident. For Polyhedral surface is closed (volumetric).

ST_IsCollection[3d] (g) Returns TRUE if the argument is a collection (MULTI*, GEOMETRYCOLLECTION, ...)

ST_IsEmpty[mm] (geomA) Returns true if this Geometry is an empty geometrycollection, polygon, point etc.

ST_IsRing[mm] (g) Returns TRUE if this LINESTRING is both closed and simple.

ST_IsSimple[mm 3d] (geomA) Returns (TRUE) if this Geometry has no anomalous geometric points, such as self intersection or self tangency.

ST_IsValid[mm g3.3] Returns true if the ST_Geometry is well formed.

```
1. g
2. g, flags
```

ST_IsValidReason[1 g3.3] Returns text stating if a geometry is valid or not and if not valid, a reason why.

```
1. geomA
2. geomA, flags
```

ST_IsValidDetail[1 g3.3] Returns a valid_detail (valid,reason,location) row stating if a geometry is valid or not and if not valid, a reason why and a location where.

```
1. geom
2. geom, flags
```

ST_M[mm 3d] (a_point) Return the M coordinate of the point, or NULL if not available. Input must be a point.

ST_NDims[3d] (g1) Returns coordinate dimension of the geometry as a small int. Values are: 2,3 or 4.

ST_NPoints[2 3d] (g1) Return the number of points (vertexes) in a geometry.

ST_NRings[3d] (geomA) If the geometry is a polygon or multi-polygon returns the number of rings.

ST_NumGeometries[2 mm 3d] (geom) If geometry is a GEOMETRYCOLLECTION (or MULTI*) return the number of geometries, for single geometries will return 1, otherwise return NULL.

ST_NumInteriorRings[mm] (a_polygon) Return the number of interior rings of the first polygon in the geometry. This will work with both POLYGON and MULTIPOLYGON types but only looks at the first polygon. Return NULL if there is no polygon in the geometry.

ST_NumInteriorRing[mm] (a_polygon) Return the number of interior rings of the first polygon in the geometry. Synonym to ST_NumInteriorRings.

ST_NumPatches[1 mm 3d] (g1) Return the number of faces on a Polyhedral Surface. Will return null for non-polyhedral geometries.

ST_NumPoints[mm] (g1) Return the number of points in an ST_LineString or ST_CircularString value.

ST_Intersects[mm G] Returns TRUE if the Geometries/Geography "spatially intersect in 2D" - (share any portion of space) and FALSE if they don't (they are Disjoint). For geography -- tolerance is 0.00001 meters (so any points that close are considered to intersect)

```
1. geomA, geomB
2. geogA, geogB
```

ST_Length[mm G] Returns the 2d length of the geometry if it is a linestring or multilinestring. geometry are in units of spatial reference and geography are in meters (default spheroid)

```
1. a_2dlinestring
2. geog, use_spheroid=true
```

ST_Length2D (a_2dlinestring) Returns the 2-dimensional length of the geometry if it is a linestring or multi-linestring. This is an alias for ST_Length

ST_3DLength[3d] (a_3dlinestring) Returns the 3-dimensional or 2-dimensional length of the geometry if it is a linestring or multi-linestring.

ST_Length_Spheroid[3d] (a_linestring, a_spheroid) Calculates the 2D or 3D length of a linestring/multilinestring on an ellipsoid. This is useful if the coordinates of the geometry are in longitude/latitude and a length is desired without reprojection.

ST_Length2D_Spheroid (a_linestring, a_spheroid) Calculates the 2D length of a linestring/multilinestring on an ellipsoid. This is useful if the coordinates of the geometry are in longitude/latitude and a length is desired without reprojection.

ST_3DLength_Spheroid[3d] (a_linestring, a_spheroid) Calculates the length of a geometry on an ellipsoid, taking the elevation into account. This is just an alias for ST_Length_Spheroid.

ST_LongestLine (g1, g2) Returns the 2-dimensional longest line points of two geometries. The function will only return the first longest line if more than one, that the function finds. The line returned will always start in g1 and end in g2. The length of the line this function returns will always be the same as st_maxdistance returns for g1 and g2.

ST_OrderingEquals[mm] (A, B) Returns true if the given geometries represent the same geometry and points are in the same directional order.

ST_Overlaps[mm] (A, B) Returns TRUE if the Geometries share space, are of the same dimension, but are not completely contained by each other.

ST_Perimeter[mm G] Return the length measurement of the boundary of an ST_Surface or ST_MultiSurface geometry or geography. (Polygon, Multipolygon). geometry measurement is in units of spatial reference and geography is in meters.

```
1. g1
2. geog, use_spheroid=true
```

ST_Perimeter2D (geomA) Returns the 2-dimensional perimeter of the geometry, if it is a polygon or multi-polygon. This is currently an alias for ST_Perimeter.

ST_3DPerimeter[3d] (geomA) Returns the 3-dimensional perimeter of the geometry, if it is a polygon or multi-polygon.

ST_PointOnSurface[mm 3d] (g1) Returns a POINT guaranteed to lie on the surface.

ST_Project[1 G] (g1, distance, azimuth) Returns a POINT projected from a start point using a distance in meters and bearing (azimuth) in radians.

ST_Relate[2 mm] Returns true if this Geometry is spatially related to anotherGeometry, by testing for intersections between the Interior, Boundary and Exterior of the two geometries as specified by the values in the intersectionMatrixPattern. If no intersectionMatrixPattern is passed in, then returns the maximum intersectionMatrixPattern that relates the 2 geometries.

```
1. geomA, geomB, intersectionMatrixPattern
2. geomA, geomB
3. geomA, geomB, BoundaryNodeRule
```

ST_RelateMatch[1 g3.3] (intersectionMatrix, intersectionMatrixPattern) Returns true if intersectionMattrixPattern1 implies intersectionMatrixPattern2

ST_ShortestLine (g1, g2) Returns the 2-dimensional shortest line between two geometries

ST_PatchN[1 mm 3d] (geomA, n) Return the 1-based Nth geometry (face) if the geometry is a POLYHEDRALSURFACE, POLYHEDRALSURFACEM. Otherwise, return NULL.

ST_PointN[mm 3d] (a_linestring, n) Return the Nth point in the first linestring or circular linestring in the geometry. Return NULL if there is no linestring in the geometry.

ST_SRID[mm] (g1) Returns the spatial reference identifier for the ST_Geometry as defined in spatial_ref_sys table.

ST_StartPoint[mm 3d] (geomA) Returns the first point of a LINESTRING geometry as a POINT.

ST_Summary[G] Returns a text summary of the contents of the geometry.

```
1. g
2. g
```

ST_X[mm 3d] (a_point) Return the X coordinate of the point, or NULL if not available. Input must be a point.

ST_XMax[3d] (aGeomorBox2DorBox3D) Returns X maxima of a bounding box 2d or 3d or a geometry.

ST_XMin[3d] (aGeomorBox2DorBox3D) Returns X minima of a bounding box 2d or 3d or a geometry.

ST_Y[mm 3d] (a_point) Return the Y coordinate of the point, or NULL if not available. Input must be a point.

ST_YMax[3d] (aGeomorBox2DorBox3D) Returns Y maxima of a bounding box 2d or 3d or a geometry.

ST_YMin[3d] (aGeomorBox2DorBox3D) Returns Y minima of a bounding box 2d or 3d or a geometry.

ST_Z[mm 3d] (a_point) Return the Z coordinate of the point, or NULL if not available. Input must be a point.

ST_ZMax[3d] (aGeomorBox2DorBox3D) Returns Z minima of a bounding box 2d or 3d or a geometry.

ST_Zmflag[3d] (geomA) Returns ZM (dimension semantic) flag of the geometries as a small int. Values are: 0=2d, 1=3dm, 2=3dz, 3=4d.

ST_ZMin[3d] (aGeomorBox2DorBox3D) Returns Z minima of a bounding box 2d or 3d or a geometry.

Geometry Editors

ST_AddPoint[3d] Adds a point to a LineString before point (0-based index).

```
1. linestring, point
2. linestring, point, position
```

ST_Affine[2 3d] Applies a 3d affine transformation to the geometry to do things like translate, rotate, scale in one step.

```
1. geomA, a, b, c, d, e, f, g, h, i, xoff, yoff, zoff
2. geomA, a, b, d, e, xoff, yoff
```

ST_Force_2D[2 3d] (geomA) Forces the geometries into a "2-dimensional mode" so that all output representations will only have the X and Y coordinates.

ST_Force_3D[2 3d] (geomA) Forces the geometries into XYZ mode. This is an alias for ST_Force_3DZ.

ST_Force_3DZ[2 3d] (geomA) Forces the geometries into XYZ mode. This is a synonym for ST_Force_3D.

ST_Force_3DM (geomA) Forces the geometries into XYM mode.

ST_Force_4D[3d] (geomA) Forces the geometries into XYZM mode.

ST_Force_Collection[2 3d] (geomA) Converts the geometry into a GEOMETRYCOLLECTION.

ST_ForceRHR[2 3d] (g) Forces the orientation of the vertices in a polygon to follow the Right-Hand-Rule.

ST_LineMerge (amultilinestring) Returns a (set of) LineString(s) formed by sewing together a MULTILINESTRING.

ST_CollectionExtract (collection, type) Given a (multi)geometry, returns a (multi)geometry consisting only of elements of the specified type.

ST_CollectionHomogenize[1] (collection) Given a geometry collection, returns the "simplest" representation of the contents.

ST_Multi (g1) Returns the geometry as a MULTI* geometry. If the geometry is already a MULTI*, it is returned unchanged.

ST_RemovePoint[3d] (linestring, offset) Removes point from a linestring. Offset is 0-based.

ST_Reverse (g1) Returns the geometry with vertex order reversed.

ST_Touches[mm] (g1, g2) Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.

ST_Within[mm] (A, B) Returns true if the geometry A is completely inside geometry B

Geometry Processing

ST_Buffer[mm G] (T) For geometry: Returns a geometry that represents points whose distance from this Geometry is less than or equal to distanc Calculations are in the Spatial Reference System of this Geometry. For geography: Uses a planar transform wrapper. Introduced in 1.5 support f different end cap and mitre settings to control shape. buffer_style options quad_segs=#,endcap=round|flat|square,join=round|mitre|bevel,mitre_limi

```
1. g1, radius_of_buffer
2. g1, radius_of_buffer, num_seg_quarter_circle
3. g1, radius_of_buffer, buffer_style_parameters
4. g1, radius_of_buffer_in_meters
```

ST_BuildArea (A) Creates an areal geometry formed by the constituer linework of given geometry

ST_Collect[3d] Return a specified ST_Geometry value from a collection other geometries.

```
1. g1field
2. g1, g2
3. g1_array
```

ST_ConcaveHull[1] (geomA, target_percent, allow_holes=false) The concave hull of a geometry represents a possibly concave geometry that encloses all geometries within the set. You can think of it as shrink wrapp

ST_ConvexHull[mm 3d] (geomA) The convex hull of a geometry represe the minimum convex geometry that encloses all geometries within the set.

ST_CurveToLine[mm 3d] Converts a CIRCULARSTRING/CURVEDPOLY to a LINESTRING/POLYGON

```
1. curveGeom
2. curveGeom, segments_per_qtr_circle
```

ST_Difference[mm 3d] (geomA, geomB) Returns a geometry that repres that part of geometry A that does not intersect with geometry B.

ST_Dump[2 3d] (g1) Returns a set of geometry_dump (geom,path) rows make up a geometry g1.

ST_DumpPoints[2 3d] (geom) Returns a set of geometry_dump (geom,p rows of all points that make up a geometry.

ST_DumpRings[3d] (a_polygon) Returns a set of geometry_dump rows, representing the exterior and interior rings of a polygon.

ST_FlipCoordinates[1 3d] (geom) Returns a version of the given geome with X and Y axis flipped. Useful for people who have built latitude/longitu features and need to fix them.

ST_Intersection[mm G] (T) Returns a geometry that represents the shar portion of geomA and geomB. The geography implementation does a transform to geometry to do the intersection and then transform back to WGS84.

```
1. geomA, geomB
2. geogA, geogB
```

ST_LineToCurve[3d] (geomANoncircular) Converts a LINESTRING/POLYGON to a CIRCULARSTRING, CURVED POLYGON

ST_MakeValid[1 3d] (input) Attempts to make an invalid geometry valid w loosing vertices.

ST_MemUnion[3d] (geomfield) Same as ST_Union, only memory-friendl (uses less memory and more processor time).

ST_MinimumBoundingCircle (geomA, num_segs_per_qt_circ=48) Returns the smallest circle polygon that can fully contain a geometry. Def uses 48 segments per quarter circle.

ST_Polygonize Aggregate. Creates a GeometryCollection containing possible polygons formed from the constituent linework of a set of geome

```
1. geomfield
2. geom_array
```

ST_Node[1 g3.3 3d] (geom) Node a set of linestrings.

ST_OffsetCurve[1 g3.3] (line, signed_distance, style_parameters='') Ret an offset line at a given distance and side from an input line. Useful for computing parallel lines about a center line

ST_RemoveRepeatedPoints[1 3d] (geom) Returns a version of the give geometry with duplicated points removed.

ST_Rotate[2 3d] Rotate a geometry rotRadians counter-clockwise about an origin.

```
1. geomA, rotRadians
2. geomA, rotRadians, x0, y0
3. geomA, rotRadians, pointOrigin
```

ST_RotateX[2 3d] (geomA, rotRadians) Rotate a geometry rotRadians about the X axis.

ST_RotateY[2 3d] (geomA, rotRadians) Rotate a geometry rotRadians about the Y axis.

ST_RotateZ[2 3d] (geomA, rotRadians) Rotate a geometry rotRadians about the Z axis.

ST_Scale[2 3d] Scales the geometry to a new size by multiplying the ordinates with the parameters. Ie: ST_Scale(geom, Xfactor, Yfactor, Zfactor).

```
1. geomA, XFactor, YFactor, ZFactor
2. geomA, XFactor, YFactor
```

ST_Segmentize (geomA, max_length) Return a modified geometry having no segment longer than the given distance. Distance computation is performed in 2d only.

ST_SetPoint[3d] (linestring, zerobasedposition, point) Replace point N of linestring with given point. Index is 0-based.

ST_SetSRID (geom, srid) Sets the SRID on a geometry to a particular integer value.

ST_SnapToGrid[3d] Snap all points of the input geometry to a regular grid.

```
1. geomA, originX, originY, sizeX, sizeY
2. geomA, sizeX, sizeY
3. geomA, size
4. geomA, pointOrigin, sizeX, sizeY, sizeZ, sizeM
```

ST_Snap[1 g3.3] (input, reference, tolerance) Snap segments and vertices of input geometry to vertices of a reference geometry.

ST_Transform[2 mm] (g1, srid) Returns a new geometry with its coordinates transformed to the SRID referenced by the integer parameter.

ST_Translate[3d] Translates the geometry to a new location using the numeric parameters as offsets. Ie: ST_Translate(geom, X, Y) or ST_Translate(geom, X, Y,Z).

```
1. g1, deltax, deltay
2. g1, deltax, deltay, deltaz
```

ST_TransScale[3d] (geomA, deltaX, deltaY, XFactor, YFactor) Translates the geometry using the deltaX and deltaY args, then scales it using the XFactor, YFactor args, working in 2D only.

Geometry Outputs

ST_AsBinary[2 mm G 3d] Return the Well-Known Binary (WKB) representation of the geometry/geography without SRID meta data.

```
1. g1
2. g1, NDR_or_XDR
3. g1
4. g1, NDR_or_XDR
```

ST_AsEWKB[2 3d] Return the Well-Known Binary (WKB) representation of the geometry with SRID meta data.

```
1. g1
2. g1, NDR_or_XDR
```

ST_AsEWKT[2 G 3d] Return the Well-Known Text (WKT) representation of the geometry with SRID meta data.

```
1. g1
2. g1
```

ST_AsGeoJSON[G 3d] Return the geometry as a GeoJSON element.

```
1. geom, maxdecimaldigits=15, options=0
2. geog, maxdecimaldigits=15, options=0
3. gj_version, geom, maxdecimaldigits=15, options=0
4. gj_version, geog, maxdecimaldigits=15, options=0
```

ST_AsGML[2 G 3d] Return the geometry as a GML version 2 or 3 element.

```
1. geom, maxdecimaldigits=15, options=0
2. geog, maxdecimaldigits=15, options=0
3. version, geom, maxdecimaldigits=15, options=0,
   nprefix=null
4. version, geog, maxdecimaldigits=15, options=0,
   nprefix=null
```

ST_SharedPaths[1 g3.3] (lineal1, lineal2) Returns a collection containing paths shared by the two input linestrings/multilinestrings.

ST_Shift_Longitude[2 3d] (geomA) Reads every point/vertex in every component of every feature in a geometry, and if the longitude coordinate <0, adds 360 to it. The result would be a 0-360 version of the data to be plotted in a 180 centric map

ST_Simplify (geomA, tolerance) Returns a "simplified" version of the g geometry using the Douglas-Peucker algorithm.

ST_SimplifyPreserveTopology (geomA, tolerance) Returns a "simplifi version of the given geometry using the Douglas-Peucker algorithm. Will avoid creating derived geometries (polygons in particular) that are invalid

ST_Split[1] (input, blade) Returns a collection of geometries resulting by splitting a geometry.

ST_SymDifference[mm 3d] (geomA, geomB) Returns a geometry that represents the portions of A and B that do not intersect. It is called a symmetric difference because ST_SymDifference(A,B) = ST_SymDifference(B,A).

ST_Union[mm] Returns a geometry that represents the point set union of Geometries.

```
1. g1field
2. g1, g2
3. g1_array
```

ST_UnaryUnion[1 g3.3 3d] (geom) Like ST_Union, but working at the geometry component level.

Linear Referencing

ST_Line_Interpolate_Point[3d] (a_linestring, a_fraction) Returns a point interpolated along a line. Second argument is a float8 between 0 and 1 representing fraction of total length of linestring the point has to be located.

ST_Line_Locate_Point (a_linestring, a_point) Returns a float between 0 and 1 representing the location of the closest point on LineString to the given Point, as a fraction of total 2d line length.

ST_Line_Substring[3d] (a_linestring, startfraction, endfraction) Return a linestring being a substring of the input one starting and ending at the given fractions of total 2d length. Second and third arguments are float8 values between 0 and 1.

ST_LocateAlong (ageom_with_measure, a_measure, offset) Return a derived geometry collection value with elements that match the specified measure. Polygonal elements are not supported.

ST_LocateBetween (geomA, measure_start, measure_end, offset) Return a derived geometry collection value with elements that match the specified range of measures inclusively. Polygonal elements are not supported.

ST_LocateBetweenElevations[3d] (geom_mline, elevation_start, elevation_end) Return a derived geometry (collection) value with elements that intersect the specified range of elevations inclusively. Only 3D, 4D LINESTRINGS and MULTILINESTRINGS are supported.

ST_InterpolatePoint[1 3d] (line, point) Return the value of the measure dimension of a geometry at the point closed to the provided point.

ST_AddMeasure[3d] (geom_mline, measure_start, measure_end) Return a derived geometry with measure elements linearly interpolated between the start and end points. If the geometry has no measure dimension, one is added. If the geometry has a measure dimension, it is over-written with new values. Only LINESTRINGS and MULTILINESTRINGS are supported.

Long Transactions Support

AddAuth (auth_token) Add an authorization token to be used in current transaction.

CheckAuth Creates trigger on a table to prevent/allow updates and deletes of rows based on authorization token.

```
1. a_schema_name, a_table_name,
   a_key_column_name
2. a_table_name, a_key_column_name
```

DisableLongTransactions () Disable long transaction support. This function removes the long transaction support metadata tables, and drops all triggers attached to lock-checked tables.

EnableLongTransactions () Enable long transaction support. This function creates the required metadata tables, needs to be called once before using the other functions in this section. Calling it twice is harmless.

ST_AsHEXEWKB[3d] Returns a Geometry in HEXEWKB format (as text) using either little-endian (NDR) or big-endian (XDR) encoding.

```
1. g1, NDRorXDR
2. g1
```

ST_AsKML[2 G 3d] Return the geometry as a KML element. Several variants. Default version=2, default precision=15

```
1. geom, maxdecimaldigits=15
2. geog, maxdecimaldigits=15
3. version, geom, maxdecimaldigits=15, nprefix=NULL
4. version, geog, maxdecimaldigits=15, nprefix=NULL
```

ST_AsSVG[G] Returns a Geometry in SVG path data given a geometry or geography object.

```
1. geom, rel=0, maxdecimaldigits=15
2. geog, rel=0, maxdecimaldigits=15
```

ST_AsX3D[1 3d] (g1, maxdecimaldigits=15, options=0) Returns a Geometry in X3D xml node element format: ISO-IEC-19776-1.2-X3DEncodings-XML

ST_GeoHash (geom, maxchars=full_precision_of_point) Return a GeoHash representation (geohash.org) of the geometry.

ST_AsText[mm G] Return the Well-Known Text (WKT) representation of the geometry/geography without SRID metadata.

```
1. g1
2. g1
```

ST_AsLatLonText[1] Return the Degrees, Minutes, Seconds representation of the given point.

```
1. pt
2. pt, format
```

LockRow Set lock/authorization for specific row in table

```
1. a_schema_name, a_table_name, a_row_key,
   an_auth_token, expire_dt
2. a_table_name, a_row_key, an_auth_token,
   expire_dt
3. a_table_name, a_row_key, an_auth_token
```

UnlockRows (auth_token) Remove all locks held by specified authorization id. Returns the number of locks released.

Miscellaneous Functions

ST_Accum[2 3d] (geomfield) Aggregate. Constructs an array of geometries.

Box2D[2] (geomA) Returns a BOX2D representing the maximum extents of the geometry.

Box3D[2 3d] (geomA) Returns a BOX3D representing the maximum extents of the geometry.

ST_Estimated_Extent Return the 'estimated' extent of the given spatial table. The estimated is taken from the geometry column's statistics. The current schema will be used if not specified.

```
1. schema_name, table_name, geocolumn_name
2. table_name, geocolumn_name
```

ST_Expand[2] Returns bounding box expanded in all directions from the bounding box of the input geometry. Uses double-precision

```
1. g1, units_to_expand
2. g1, units_to_expand
3. g1, units_to_expand
```

ST_Extent[2] (geomfield) an aggregate function that returns the bounding box that bounds rows of geometries.

ST_3DExtent[2 3d] (geomfield) an aggregate function that returns the box3D bounding box that bounds rows of geometries.

Find_SRID (a_schema_name, a_table_name, a_geomfield_name) The syntax is find_srid(db_schema, table,column) and the function returns the integer SRID of the specified column by searching through the GEOMETRY_COLUMNS table.

ST_Mem_Size[3d] (geomA) Returns the amount of space (in bytes) the geometry takes.

ST_Point_Inside_Circle (a_point, center_x, center_y, radius) Is the point geometry insert circle defined by center_x, center_y, radius

Exceptional Functions

PostGIS_AddBBox (geomA) Add bounding box to the geometry.

PostGIS_DropBBox (geomA) Drop the bounding box cache from the geometry.

PostGIS_HasBBox (geomA) Returns TRUE if the bbox of this geometry is cached, FALSE otherwise.