### PostGIS 2.0 Cheatsheet

New in this release: 1. Enhanced in this release: 2. Requires GEOS 3.3 or higher. 3. 2.5/3D support.

#### PostgreSQL PostGIS Geometry/Geography/Box Types

- **box2d** - A box composed of xmin, ymin, xmax, ymax. Often used to return the 2d bounding box of a geometry.
- **box3d** - A box composed of xmin, ymin, zmin, xmax, ymax, zmax. Often used to return the 3d extent of a geometry or collection of geometries.
- **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** 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 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 info.

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

- **PostGIS_Lib_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 postgres scripts installed in this database.

- **PostGIS_Scripts_Release** () - 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.

#### Spatial Relationships and Measurements

- **ST_3DClosestPoint** 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** 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** 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** 3d - (g1, g2, distance_of_srid) For 3d (z) geometry type Returns true if all of the 3D geometries are within the specified distance of one another.

- **ST_3DIntersects** 3d - (geomA, geomB) Returns TRUE if the Geometries "spatially intersect" in 3d - only for points and linestrings.

- **ST_3DLongestLine** 3d - (g1, g2) Returns the 3-dimensional longest line between two geometries.

- **ST_3DMaxDistance** 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** 3d - (g1, g2) Returns the 3-dimensional shortest line between two geometries.

#### PostGIS Full Version

- 1

- 2

- 3

- G

### Operators

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&amp; &amp; G</td>
<td>Returns TRUE if A's 2D bounding box intersects B's 2D bounding box.</td>
</tr>
<tr>
<td>&amp;&lt; G</td>
<td>Returns TRUE if A's bounding box overlaps or is to the left of B's.</td>
</tr>
<tr>
<td>&amp;&lt; (A, B)</td>
<td>Returns TRUE if A's bounding box overlaps or is below B's.</td>
</tr>
<tr>
<td>&amp;&gt; (A, B)</td>
<td>Returns TRUE if A's bounding box overlaps or is to the right of B's.</td>
</tr>
<tr>
<td>&lt;&lt;= (A, B)</td>
<td>Returns TRUE if A's bounding box is strictly below B's.</td>
</tr>
<tr>
<td>&lt;= (A, B)</td>
<td>Returns TRUE if A's bounding box is the same as B's. Uses double precision bounding box.</td>
</tr>
<tr>
<td>&gt;&gt; (A, B)</td>
<td>Returns TRUE if A's bounding box is strictly to the right of B's.</td>
</tr>
<tr>
<td>@ (A, B)</td>
<td>Returns TRUE if A's bounding box is contained by B's.</td>
</tr>
<tr>
<td>&amp;&gt; (A, B)</td>
<td>Returns TRUE if A's bounding box overlaps or is above B's.</td>
</tr>
<tr>
<td>&gt;= (A, B)</td>
<td>Returns TRUE if A's bounding box is strictly above B's.</td>
</tr>
<tr>
<td>= (A, B)</td>
<td>Returns TRUE if A's bounding box contains B's.</td>
</tr>
<tr>
<td>=~ (A, B)</td>
<td>Returns TRUE if A's bounding box is the same as B's.</td>
</tr>
<tr>
<td>&lt;-&gt; (A, B)</td>
<td>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.</td>
</tr>
<tr>
<td>&lt;= (g1, g2, distance)</td>
<td>Returns true if all of the geometries &quot;spatially intersect&quot; in 3d - only for points and linestrings.</td>
</tr>
<tr>
<td>&lt;#&gt; (g1, g2)</td>
<td>Returns the 3-dimensional shortest line between two geometries.</td>
</tr>
<tr>
<td>&lt;#&gt; (g1, g2, distance_of_srid)</td>
<td>For 3d (z) 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.</td>
</tr>
</tbody>
</table>

### 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** (EWKT) - Return a specified geography value from Well-Known Text representation or extended (WKT).
### PostGIS 2.0 Cheat Sheet v2.0.2

#### ST_GeographyFromText(G) (EWKT)
Returns 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
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
Returns a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

#### ST_GeomFromEWKT
Returns a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

#### ST_GeometryFromText
Returns a specified ST_Geometry value from Well-Known Text representation (WKT).

1. WKT
2. WKT, srid

#### ST_GeomFromGML
Takes as input GML representation of geometry and outputs a PostGIS geometry object

1. geomxml
2. geomxml, srid

#### ST_GeomFromGeoJSON
Takes as input a geojson representation of a geometry and outputs a PostGIS geometry object

#### ST_GeomFromKML
Takes as input KML representation of geometry and outputs a PostGIS geometry object

#### ST_GMLToSQL
Returns a specified ST_Geometry value from GML representation. This is an alias name for ST_GeomFromGML

1. geomxml
2. geomxml, srid

#### ST_GeomFromText
Returns 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_GeomFromWKB
Creates a geometry instance from a Well-Known Binary geometry representation (WKB) and optional SRID.

1. geom
2. geom, srid

#### ST_GeomFromWKB
Creates a geometry instance from a Well-Known Binary geometry representation (WKB) or extended Well Known Binary (EWKB).

#### ST_GeomFromWKB
Returns a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

#### ST_GeomFromWKT
Returns a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

#### ST_GeomFromText
Returns 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_GeomFromWKB
Takes as input GML representation of geometry and outputs a PostGIS geometry object

1. geomxml
2. geomxml, srid

#### ST_GeomFromWKB
Takes as input GML representation of geometry and outputs a PostGIS geometry object

#### ST_GeomFromWKB
Returns a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

#### ST_GeomFromWKT
Returns a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

#### ST_GeometryFromText
Returns 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
Returns a specified ST_Geometry value from GML representation. This is an alias name for ST_GeomFromGML

1. geomxml
2. geomxml, srid

#### ST_GeomFromText
Returns a specified ST_Geometry value from Well-Known Text representation (WKT).

1. WKT
2. WKT, srid

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

1. geom
2. geom, srid

#### ST_GeomFromWKB
Creates a geometry instance from a Well-Known Binary geometry representation (WKB) or extended Well Known Binary (EWKB).

#### ST_GeomFromWKB
Returns a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

#### ST_GeomFromWKT
Returns a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

#### ST_GeometryFromText
Returns 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_GeomFromWKB
Takes as input GML representation of geometry and outputs a PostGIS geometry object

1. geomxml
2. geomxml, srid

### ST_Azimuth
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
Returns the geometric center of a geometry.

#### ST_ClosestPoint(gg1, gg2)
Returns the 2-dimensional point on gg1 that is closest to gg2. This is the first point of the shortest line.

#### ST_Contains
Returns true if ggA is within ggB. ggA is fully contained in ggB, or otherwise stated ggB contains ggA. ggA and ggB intersect at least at one point. ggB contains ggA but ggA is not contained in ggB.

1. ggA, ggB
2. geomA, geomB

#### ST_CoveredBy
Returns true if a geometry or geometry array 1 contains 2. If ggB contains ggA, returns a number between -3 and 3 denoting what kind of covering behavior.

1. gg1, gg2
2. geom1, geom2, use_spheroid

#### ST_CoveredBy
Returns true if a geometry or geometry array contains a circular string.

1. geomA, geomB
2. geospatialArray, geospatialPointB

#### ST_Crosses
Returns true if the supplied geometries have some, but not all, interior points in common.

1. geom1, geom2
2. geomA, geomB, distance_meters, use_spheroid

#### ST_CrossingDirection
Given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behavior. 0 is no crossing.

1. geom1, geom2
2. g1, g2, distance_of_srid

#### ST_Distance_Spheroid
Returns the minimum distance between two geometries in meters. For geometry type Returns the 2-dimensional cartesian minimum distance between two geometries in projected units. For geography type Returns the 2-dimensional spherical minimum distance between two geometries in meters. Faster than ST_Distance_Spheroid , but less accurate. PostGIS versions prior to 1.5 only implemented for points.

1. g1, g2
2. use_spheroid

#### ST_Distance_Spheroid
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. use_3d

#### ST_Distance_Spheroid
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, distanceofsriz

#### ST_DWithin
Returns true if all of the geometries are within the specified distance of one another.

1. g1, g2, distance

#### ST_DWithin
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=use_spheroid (measure around spheroid), for faster check, use_spheroid=false to measure along sphere.

1. g1, g2, distance
2. g1, g2, distanceofsriz

#### ST_EqualWKB
Returns true if the given geometries represent the same geometry. Directionality is ignored.

1. g1, g2
2. geom1, geom2, geom3

#### ST_HasArc
Returns true if a geometry or geometry collection contains a circular string.
### Geometry Type
- `GeometryType2 3d (geomA)` Returns the type of the geometry as a string. E.g: 'LINESTRING', 'POLYGON', 'MULTIPOINT', etc.

### Boundary
- `ST_Boundary3d (geomA)` Returns the closure of the combinatorial boundary of this Geometry.

### Coordinate Dimension
- `ST_CoordDim3d (geomA)` Return the coordinate dimension of the ST_Geometry value.

### Dimension
- `ST_Dimension2 (g)` The inherent dimension of this Geometry object, which must be less than or equal to the coordinate dimension.

### End Point
- `ST_EndPoint3d (g)` Returns the last point of a LINESTRING geometry as a POINT.

### Envelope
- `ST_Envelope(g1)` Returns a geometry representing the double precision (float8) bounding box of the supplied geometry.

### Exterior Ring
- `ST_ExteriorRing3d (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.

### Geometry N
- `ST_GeometryN2 (geomA, n)` Return the Nth geometry of the ST_Geometry value.

### Geometry Type
- `ST_GeometryType2 3d (g1)` Return the geometry type of the ST_Geometry value.

### Interior Ring
- `ST_InteriorRing3d (a_polygon, n)` Return the nth interior ring of the polygon geometry. Return NULL if the geometry is not a polygon or the given N is out of range.

### Is Closed
- `ST_IsClosed3d (g)` Returns TRUE if the LINESTRING's start and end points are coincident. For Polyhedral surface is closed (volumetric).

### Is Collection
- `ST_IsCollection3d (g)` Returns TRUE if the argument is a collection (MULTI*, GEOMETRYCOLLECTION, ...) not valid, a reason why.

### Is Empty
- `ST_IsEmpty(g)` Returns true if this Geometry is an empty geometrycollection, polygon, point etc.

### Is Empty
- `ST_IsEmpty(g)` Returns true if this Geometry is an empty geometrycollection, polygon, point etc.

### Is Listring
- `ST_IsListring(g)` Returns true if the argument is a collection (MULTI*, GEOMETRYCOLLECTION, ...) not valid, a reason why.

### Is Simple
- `ST_IsSimple2 (g)` Returns TRUE if this Geometry has no anomalous geometric points, such as self intersection or self tangency.

### Is Valid
- `ST_IsValid3d (g)` Returns true if the ST_Geometry is well formed.

### M
- `ST_M3d (a_point)` Return the M coordinate of the point, or NULL if not available. Input must be a point.

### N Dims
- `ST_NDims3d (g1)` Returns coordinate dimension of the geometry as a small int. Values are: 2 or 4.

### N Points
- `ST_NPoints2 (g)` Returns the number of points (vertexes) in a geometry. This will work with both POLYGON and MULTIPOLYGON.

### Num Geometries
- `ST_NumGeometries2 mm 3d (geom)` If geometry is a GEOMETRYCOLLECTION or MULTIGEOMETRY, return the number of geometries, for single geometry will return 1, otherwise return NULL.

### Num Interior Rings
- `ST_NumInteriorRings2 (a_polygon)` Return the number of interior rings of the first polygon in the geometry. This will work with both POLYGON and MULTIPOLYGON but only looks at the first polygon. Return NULL if there is no polygon in the geometry.

### Num Interior Rings
- `ST_NumInteriorRings2 (a_polygon)` Return the number of interior rings of the first polygon in the geometry. Synonym to ST_NumInteriorRings.

### Num Patches
- `ST_NumPatches1 mm 3d (g1)` Return the number of faces on a Polyhedral Surface. Will return NULL for non-polyhedral geometries.

### Num Points
- `ST_NumPoints2 (g1)` Return the number of points in an ST_LineString or ST_CircularString value.

### Shortest Line
- `ST_ShortestLine2 g1, geomA, geomB` Calculates the minimum 2-dimensional line between two geometries.

### Related
- `ST_Relate3d (a_polygon, a_polygon)` Calculates the minimum 3-dimensional line between two geometries.

### Shortest Polygon
- `ST_ShortestPolygon2 geomA, geomB` Calculates the minimum Polygon line between two geometries.

### Related
- `ST_Relation2 (g1)` Returns true if Geom1 and Geom2 are the same geometry and points are in the same directional order.

### Overlaps
- `ST_Overlaps2 (A, B)` Returns true if geometries share space, are of the same dimension, but are not completely contained by each other.

### Overlaps
- `ST_Overlaps2 (A, B)` Returns true if geometries share the same representation and points are in the same directional order.

### Shortest Line
- `ST_ShortestLine2 (g1, g2)` Returns the 2-dimensional shortest line between two geometries.

### Length
- `ST_Length3d (g)` Returns the length of a geometry on an ellipsoid. geometry measurement is in units of spatial reference and geography is in meters (default spheroid).

### Length
- `ST_Length3d (g)` Returns the length of a geometry on an ellipsoid. geometry measurement is in units of spatial reference and geography is in meters (default spheroid).

### Length 2D
- `ST_Length2D (g)` Returns the 2-dimensional length of a linestring or multilinestring. This is an alias for ST_Length.

### Length 3D
- `ST_Length3D (g)` Returns the 3-dimensional length of a linestring or multilinestring. This is an alias for ST_Length.

### Spheroid
- `ST_LengthSpheroid2d (a_2dlinestring)` 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.

### Spheroid
- `ST_LengthSpheroid2d (a_2dlinestring)` 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.

### Longest Line
- `ST_LongestLine2 (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.

### Ordering Equals
- `ST_OrderingEquals2 (A, B)` Returns true if the given geometries represent the same geometry and points are in the same directional order.

### Point On Surface
- `ST_PointOnSurface3d (g1)` Returns a POINT guaranteed to lie on the surface.

### Project
- `ST_Project2 (g1, distance, azimuth)` Returns a POINT projected from a start point using a distance in meters and bearing (azimuth) in radians.

### Related
- `ST_Relate)` Returns true if this Geometry is spatially related to another Geometry, 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 two geometries.

### Project
- `ST_Project2 (g1, distance, azimuth)` Returns a POINT projected from a start point using a distance in meters and bearing (azimuth) in radians.

### Shortest Line
- `ST_ShortestLine2 (g1, g2)` Returns the 2-dimensional shortest line between two geometries.
ST_Patch  Returns a geometry that represents all of a geometry's POLYHEDRALSURFACE.

ST_Point  Returns the Nth point in the first LineString in a GEOMETRYCOLLECTION.

ST_SRID  Returns the spatial reference identifier for the ST_Geometry as defined in spatial_ref_sys table.

ST_StartPoint  Returns the first point of a LINESTRING or a POINT.

ST_Summary  Returns a text summary of the contents of the geometry. 

Geometry Editors

ST_AddPoint  Adds a point to a LineString before point (0-based index).

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

ST_Force_2D  Forces the geometries into a 2-dimensional mode so that all output representations will only have the X and Y coordinates.

ST_Force_3D  Forces the geometries into XYZ mode. This is an alias for ST_Force_3DZ.

ST_Force_3DZ  Forces the geometries into XYZ mode. This is a synonym for ST_Force_3D.

ST_Force_4D  Forces the geometries into XYZM mode.

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

ST_CollectionHomogenize  Given a geometry collection, returns the "simplest" representation of the contents.

ST_MakeValid  Forces the geometries into XY2M mode.

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

ST_CollectionHomogenize  Given a (multi)geometry, returns the "simplest" representation of the contents.

ST_MakeValid  Forces the geometries into XY2M mode.

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

ST_CollectionHomogenize  Given a geometry collection, returns the "simplest" representation of the contents.

ST_MakeValid  Forces the geometries into XY2M mode.

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

ST_CollectionHomogenize  Given a (multi)geometry, returns the "simplest" representation of the contents.

ST_MakeValid  Forces the geometries into XY2M mode.

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

ST_CollectionHomogenize  Given a geometry collection, returns the "simplest" representation of the contents.

ST_MakeValid  Forces the geometries into XY2M mode.

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

ST_CollectionHomogenize  Given a geometry collection, returns the "simplest" representation of the contents.
<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ST_Rotate</code></td>
<td>Rotate a geometry counterclockwise about an origin.</td>
<td>1. <code>geomA</code>, <code>rotationRadians</code> 2. <code>geomA</code>, <code>rotationRadians</code>, <code>x0</code>, <code>y0</code> 3. <code>geomA</code>, <code>rotationRadians</code>, <code>pointOrigin</code></td>
</tr>
<tr>
<td><code>ST_RotateX</code></td>
<td>Rotate a geometry <code>rotationRadians</code> about the X axis.</td>
<td><code>geomA</code>, <code>rotationRadians</code></td>
</tr>
<tr>
<td><code>ST_RotateY</code></td>
<td>Rotate a geometry <code>rotationRadians</code> about the Y axis.</td>
<td><code>geomA</code>, <code>rotationRadians</code></td>
</tr>
<tr>
<td><code>ST_RotateZ</code></td>
<td>Rotate a geometry <code>rotationRadians</code> about the Z axis.</td>
<td><code>geomA</code>, <code>rotationRadians</code></td>
</tr>
<tr>
<td><code>ST_Scale</code></td>
<td>Scales the geometry to a new size by multiplying the ordinates with the parameters.</td>
<td>1. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code> 2. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code>, <code>ZFactor</code></td>
</tr>
<tr>
<td><code>ST_Translate</code></td>
<td>Translates the geometry to a new location using the numeric parameters as offsets.</td>
<td>1. <code>geomA</code>, <code>deltax</code>, <code>deltay</code>, <code>deltaz</code> 2. <code>geomA</code>, <code>deltax</code>, <code>deltay</code></td>
</tr>
<tr>
<td><code>ST_Segmentsize</code></td>
<td>Returns a modified geometry having no segment longer than the given distance.</td>
<td><code>geomA</code>, <code>max_length</code></td>
</tr>
<tr>
<td><code>ST_SnapToGrid</code></td>
<td>Snap all points of the input geometry to a regular grid.</td>
<td><code>geomA</code>, <code>originX</code>, <code>originY</code>, <code>sizeX</code>, <code>sizeY</code></td>
</tr>
<tr>
<td><code>ST_Snap</code></td>
<td>Snap segments and vertices of input geometry to vertices of a reference geometry.</td>
<td>1. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code> 2. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code>, <code>ZFactor</code></td>
</tr>
<tr>
<td><code>ST_Transform</code></td>
<td>Returns a new geometry with its coordinates transformed to the SRID referenced by the integer parameter.</td>
<td><code>g1</code>, <code>srid</code></td>
</tr>
<tr>
<td><code>ST_Translate</code></td>
<td>Translates the geometry to a new location using the numeric parameters as offsets.</td>
<td><code>g1</code>, <code>deltax</code>, <code>deltay</code>, <code>deltaz</code></td>
</tr>
<tr>
<td><code>ST_Translate</code></td>
<td>Translates the geometry using the <code>deltax</code> and <code>deltay</code> arguments, then scales it using the <code>XFactor</code> and <code>YFactor</code> arguments, working in 2D only.</td>
<td><code>geomA</code>, <code>deltax</code>, <code>deltay</code>, <code>deltaz</code></td>
</tr>
</tbody>
</table>

### Linear Referencing

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ST_Line_Union</code></td>
<td>Returns a geometry that represents the point set union of Geometries.</td>
<td>1. <code>g1</code>, <code>g2</code> 2. <code>g1_array</code></td>
</tr>
<tr>
<td><code>ST_Simplify</code></td>
<td>Returns a &quot;simplified&quot; version of the geometry using the Douglas-Peucker algorithm.</td>
<td><code>geomA</code>, <code>tolerance</code></td>
</tr>
<tr>
<td><code>ST_SimplifyPreserveTopology</code></td>
<td>Returns a &quot;simplified&quot; version of the given geometry using the Douglas-Peucker algorithm. Will avoid creating derived geometries (polygons in particular) that are invalid.</td>
<td><code>geomA</code>, <code>tolerance</code></td>
</tr>
<tr>
<td><code>ST_Segmentsize</code></td>
<td>Return a modified geometry having no segment longer than the given distance.</td>
<td><code>geomA</code>, <code>max_length</code></td>
</tr>
<tr>
<td><code>ST_SnapToGrid</code></td>
<td>Snap all points of the input geometry to a regular grid.</td>
<td><code>geomA</code>, <code>originX</code>, <code>originY</code>, <code>sizeX</code>, <code>sizeY</code></td>
</tr>
<tr>
<td><code>ST_Snap</code></td>
<td>Snap segments and vertices of input geometry to vertices of a reference geometry.</td>
<td>1. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code> 2. <code>geomA</code>, <code>XFactor</code>, <code>YFactor</code>, <code>ZFactor</code></td>
</tr>
<tr>
<td><code>ST_Transform</code></td>
<td>Returns a new geometry with its coordinates transformed to the SRID referenced by the integer parameter.</td>
<td><code>g1</code>, <code>srid</code></td>
</tr>
<tr>
<td><code>ST_Translate</code></td>
<td>Translates the geometry to a new location using the numeric parameters as offsets.</td>
<td><code>g1</code>, <code>deltax</code>, <code>deltay</code>, <code>deltaz</code></td>
</tr>
<tr>
<td><code>ST_Translate</code></td>
<td>Translates the geometry using the <code>deltax</code> and <code>deltay</code> arguments, then scales it using the <code>XFactor</code> and <code>YFactor</code> arguments, working in 2D only.</td>
<td><code>geomA</code>, <code>deltax</code>, <code>deltay</code>, <code>deltaz</code></td>
</tr>
</tbody>
</table>

### Geometry Outputs

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ST_AsBinary</code></td>
<td>Return the Well-Known Binary (WKB) representation of the geometry.</td>
<td>1. <code>g1</code> 2. <code>g1</code>, <code>NDR_or_XDR</code> 3. <code>g1</code></td>
</tr>
<tr>
<td><code>ST_AsEWKB</code></td>
<td>Return the Well-Known Binary (WKB) representation of the geometry with SRID meta data.</td>
<td>1. <code>g1</code> 2. <code>g1</code>, <code>NDR_or_XDR</code></td>
</tr>
<tr>
<td><code>ST_AsEWKT</code></td>
<td>Return the Well-Known Text (WKT) representation of the geometry with SRID meta data.</td>
<td>1. <code>g1</code></td>
</tr>
<tr>
<td><code>ST_AsGeoJSON</code></td>
<td>Return the geometry as a GeoJSON element.</td>
<td>1. <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code> 2. <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code> 3. <code>gj_version</code>, <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code> 4. <code>gj_version</code>, <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code></td>
</tr>
<tr>
<td><code>ST_AsGML</code></td>
<td>Return the geometry as a GML version 2 or 3 element.</td>
<td>1. <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code> 2. <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code> 3. <code>version</code>, <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code>, <code>nprefix=null</code> 4. <code>version</code>, <code>geom</code>, <code>maxdecimaldigits=15</code>, <code>options=0</code>, <code>nprefix=null</code></td>
</tr>
</tbody>
</table>

### Long Transactions Support

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AddAuth</code></td>
<td>Add an authorization token to be used in current transaction.</td>
<td><code>auth_token</code></td>
</tr>
<tr>
<td><code>CheckAuth</code></td>
<td>Creates trigger on a table to prevent/allow updates and deletes of rows based on authorization token.</td>
<td><code>a_schema_name</code>, <code>a_table_name</code>, <code>a_key_column_name</code></td>
</tr>
<tr>
<td><code>DisableLongTransactions</code></td>
<td>Disables long transaction support.</td>
<td>()</td>
</tr>
<tr>
<td><code>EnableLongTransactions</code></td>
<td>Enables long transaction support.</td>
<td>()</td>
</tr>
</tbody>
</table>
### Miscellaneous Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ST_AsHEXEWKB</strong>&lt;sup&gt;3d&lt;/sup&gt;</td>
<td>Returns a Geometry in HEXEWKB format (as text) using either little-endian (NDR) or big-endian (XDR) encoding.</td>
<td>1. <code>g1</code>, NDROrXDR 2. <code>g1</code></td>
</tr>
<tr>
<td><strong>ST_AsKML</strong>&lt;sup&gt;G&lt;/sup&gt;</td>
<td>Return the geometry as a KML element. Several variants. Default version=2, default precision=15</td>
<td>1. <code>geom</code>, maxdecimaldigits=15 2. <code>geog</code>, maxdecimaldigits=15 3. <code>version</code>, <code>geom</code>, maxdecimaldigits=15, <code>nprefix=NULL</code> 4. <code>version</code>, <code>geog</code>, maxdecimaldigits=15, <code>nprefix=NULL</code></td>
</tr>
<tr>
<td><strong>ST_AsSVG</strong></td>
<td>Returns a Geometry in SVG path data given a geometry or geography object.</td>
<td>1. <code>geom</code>, <code>rel=0</code>, maxdecimaldigits=15 2. <code>geog</code>, <code>rel=0</code>, maxdecimaldigits=15</td>
</tr>
<tr>
<td><strong>ST_AsLatLonText</strong>&lt;sup&gt;1&lt;/sup&gt;</td>
<td>Return the Degrees, Minutes, Seconds representation of the given point.</td>
<td>1. <code>pt</code> 2. <code>pt</code>, <code>format</code></td>
</tr>
<tr>
<td><strong>LockRow</strong></td>
<td>Set lock/authorization for specific row in table</td>
<td>1. <code>a_schema_name</code>, <code>a_table_name</code>, <code>a_row_key</code>, <code>an_auth_token</code>, <code>expire_dt</code> 2. <code>a_table_name</code>, <code>a_row_key</code>, <code>an_auth_token</code>, <code>expire_dt</code> 3. <code>a_table_name</code>, <code>a_row_key</code>, <code>an_auth_token</code></td>
</tr>
<tr>
<td><strong>UnlockRows</strong></td>
<td>Remove all locks held by specified authorization id. Returns the number of locks released.</td>
<td><code>(auth_token)</code></td>
</tr>
</tbody>
</table>

### Exceptional Functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Arguments</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Find_SRID</strong>&lt;sup&gt;1&lt;/sup&gt;</td>
<td>The syntax is <code>find_srid(db_schema, table,column)</code> and the function returns the integer SRID of the specified column by searching through the <code>GEOMETRY_COLUMNS</code> table.</td>
<td><code>(a_schema_name, a_table_name, a_geomfield_name)</code></td>
</tr>
<tr>
<td><strong>ST_Accum</strong>&lt;sup&gt;2d&lt;/sup&gt;</td>
<td>(geomfield) Aggregate. Constructs an array of geometries.</td>
<td></td>
</tr>
<tr>
<td><strong>Box2D</strong>&lt;sup&gt;2&lt;/sup&gt;</td>
<td>(geomA) Returns a BOX2D representing the maximum extents of the geometry.</td>
<td></td>
</tr>
<tr>
<td><strong>Box3D</strong>&lt;sup&gt;3d&lt;/sup&gt;</td>
<td>(geomA) Returns a BOX3D representing the maximum extents of the geometry.</td>
<td></td>
</tr>
<tr>
<td><strong>ST_Estimated_Extent</strong></td>
<td>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.</td>
<td>1. <code>schema_name</code>, <code>table_name</code>, <code>geocolumn_name</code> 2. <code>table_name</code>, <code>geocolumn_name</code></td>
</tr>
<tr>
<td><strong>ST_Expand</strong>&lt;sup&gt;2d&lt;/sup&gt;</td>
<td>Returns bounding box expanded in all directions from the bounding box of the input geometry. Uses double-precision</td>
<td>1. <code>g1</code>, <code>units_to_expand</code> 2. <code>g1</code>, <code>units_to_expand</code> 3. <code>g1</code>, <code>units_to_expand</code></td>
</tr>
<tr>
<td><strong>ST_Extent</strong>&lt;sup&gt;2d&lt;/sup&gt;</td>
<td>(geomfield) an aggregate function that returns the bounding box that bounds rows of geometries.</td>
<td></td>
</tr>
<tr>
<td><strong>ST_3DExtent</strong>&lt;sup&gt;3d&lt;/sup&gt;</td>
<td>(geomfield) an aggregate function that returns the box3D bounding box that bounds rows of geometries.</td>
<td></td>
</tr>
<tr>
<td><strong>Find_SRID</strong>&lt;sup&gt;2d&lt;/sup&gt;</td>
<td>(a_schema_name, a_table_name, a_geomfield_name) The syntax is <code>find_srid(db_schema, table, column)</code> and the function returns the integer SRID of the specified column by searching through the <code>GEOMETRY_COLUMNS</code> table.</td>
<td></td>
</tr>
<tr>
<td><strong>ST_Mem_Size</strong>&lt;sup&gt;3d&lt;/sup&gt;</td>
<td>(geomA) Returns the amount of space (in bytes) the geometry takes.</td>
<td></td>
</tr>
<tr>
<td><strong>ST_Point_Inside_Circle</strong>&lt;sup&gt;1&lt;/sup&gt;</td>
<td>(a_point, center_x, center_y, radius) Is the point geometry insert circle defined by center_x, center_y, radius</td>
<td></td>
</tr>
</tbody>
</table>

---

This work is licensed under a Creative Commons Attribution . Feel free to use this material for private or commercial purposes, but we ask that you please retain the [PostGIS in Action](http://www.postgis.us) PostGIS in Action website link.

PostGIS in Action  BostonGIS.com  Paragon Corporation  Postgres OnLine Journal