

## Post GIS Day 2011 Commemorative Playing Cards



<http://www.postgis.org>

Celebrate this Post GIS day with these versatile Post GIS day commemorative playing cards. The number of games and fun-filled hours you can have with these cards is priceless. Here is a small listing of the infinite number of games you can play with Post GIS cards:

- **Name that thing** In this game you have the descriptions face up and have the opponent guess the name of the function, type, or operator.
- **What does it do?** In this game you have the name of the thing face up and have the opponent describe what the thing does or is for. Your friends and even strangers you tricked into playing this game will be amazed at your mastery of the 400 some-odd functions PostGIS provides. *To be able to exercise all 400 some-odd functions, you need to be running PostGIS 2.0+*
- **Post GIS war game** This game requires no knowledge of PostGIS what-so-ever. In this game, you play with the descriptions face up. Even your kids will like this game, and may learn how to use PostGIS better than you. There are two joker cards -- the "What Is Post GIS?" and "What does Post GIS?". Any player that is dealt either of these cards wins - period. For other cards the order of precedence is: <sup>1</sup> - Is super and beats anything else except another <sup>1</sup> or joker card. In the event of multiple <sup>1</sup>, the one that happens alphabetically first trumps the others. Symbols always trump letters.  
<sup>2</sup> - Second favorite, alphabetical rules apply (is beaten by a joker, <sup>1</sup>)  
<sup>mm</sup> - third highest ranking  
 All other cards precedence by alphabetical order.
- **Post GIS in a language I don't understand** To celebrate the ubiquity of PostGIS, you can create Post GIS playing cards in a language you don't understand. Here is what you do. Go to <http://translate.google.com> and paste in the URL to this page in the first text box (make sure it is set to English), in the **To:** drop down, pick a language you do not know, but preferably you have friends that speak that language and can laugh at your grammar and pronunciation. In no time you'll be able to impress your friends living far far away with your command of their language. **Warning: because of the great number of functions PostGIS has to offer, Google (or any other translator) may refuse to translate all cards leaving you with a mix of some other language and English cards.**
- **Post GIS in a language I do understand** Similar to the I don't understand game, except you pick a non-english language that you do understand. Enjoy many moments of laughter reading machine generated translations that are sorta accurate but often comical.
- **The Scotch and Milk moment, the beginning of all brilliant ideas** You realize there are 66 pages each of which has 6 cards. You realize you are a grown-up and grown-ups look silly cutting out cards from paper unless if accompanied by a minor. You have a kid staring at you wondering why this day is so special. *Eureka Moment* Pour yourself a glass of scotch and the kid a glass of milk and whip out the old scissors, glue, and print outs. **Serving suggestion:** It might be a good idea to pour the Scotch in a clear glass so you don't hand out the wrong glass to the kids. After the second helping, it might be prudent to stay away from the scissors.
- Invent your own Post GIS card game. The possibilities are only limited by your imagination.

<p><b>WHAT IS POST GIS?</b></p>	<p><b>POSTGIS</b></p> 
<p><b>WHAT DOES POST GIS?</b></p>	<p><b>POSTGIS</b></p> 
<p><b>box2d</b></p>	<p>A box composed of x min, ymin, xmax, ymax. Often used to return the 2d enclosing box of a geometry.</p>
<p><b>box3d</b></p>	<p>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.</p>
<p><b>geometry</b></p>	<p>Planar spatial data type.</p>
<p><b>geometry_dump</b></p>	<p>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.)</p>

<b>geography</b>	Ellipsoidal spatial data type.
<b>AddGeometryColumn<sup>2 3d</sup></b>	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_tpmmod to get old check constraint based behavior
<b>DropGeometryColumn<sup>3d</sup></b>	Removes a geometry column from a spatial table.
<b>DropGeometryTable</b>	Drops a table and all its references in geometry_columns.
<b>PostGIS_Full_Version</b>	Reports full postgis version and build configuration infos.
<b>PostGIS_GEOS_Version</b>	Returns the version number of the GEOS library.

<b>PostGIS_LibXML_Version</b>	Returns the version number of the libxml2 library.
<b>PostGIS_Lib_Build_Date</b>	Returns build date of the PostGIS library.
<b>PostGIS_Lib_Version</b>	Returns the version number of the PostGIS library.
<b>PostGIS_PROJ_Version</b>	Returns the version number of the PROJ4 library.
<b>PostGIS_Scripts_Build_Date</b>	Returns build date of the PostGIS scripts.
<b>PostGIS_Scripts_Installed</b>	Returns version of the postgis scripts installed in this database.

<b>PostGIS_Scripts_Released</b>	Returns the version number of the postgis.sql script released with the installed postgis lib.
<b>PostGIS_Uses_Stats</b>	Returns TRUE if STATS usage has been enabled.
<b>PostGIS_Version</b>	Returns PostGIS version number and compile-time options.
<b>Populate_Geometry_Columns<sup>2</sup></b>	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
<b>UpdateGeometrySRID<sup>3d</sup></b>	Updates the SRID of all features in a geometry column, geometry_columns metadata and srid table constraint
<b>ST_BdPolyFromText</b>	Construct a Polygon given an arbitrary collection of closed linestrings as a MultiLineString Well-Known text representation.

<b>ST_BdMPolyFromText</b>	Construct a MultiPolygon given an arbitrary collection of closed linestrings as a MultiLineString text representation Well-Known text representation.
<b>ST_GeogFromText<sup>G</sup></b>	Return a specified geography value from Well-Known Text representation or extended (WKT).
<b>ST_GeographyFromText<sup>G</sup></b>	Return a specified geography value from Well-Known Text representation or extended (WKT).
<b>ST_GeogFromWKB<sup>G</sup></b>	Creates a geography instance from a Well-Known Binary geometry representation (WKB) or extended Well Known Binary (EWKB).
<b>ST_GeomCollFromText<sup>mm</sup></b>	Makes a collection Geometry from collection WKT with the given SRID. If SRID is not give, it defaults to -1.
<b>ST_GeomFromEWKB<sup>2 3d</sup></b>	Return a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).

<b>ST_GeomFromEWKT<sup>2 3d</sup></b>	Return a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).
<b>ST_GeometryFromText<sup>mm</sup></b>	Return a specified ST_Geometry value from Well-Known Text representation (WKT). This is an alias name for ST_GeomFromText
<b>ST_GeomFromGML<sup>2 3d</sup></b>	Takes as input GML representation of geometry and outputs a PostGIS geometry object
<b>ST_GeomFromGeoJSON<sup>1 3d</sup></b>	Takes as input a geojson representation of a geometry and outputs a PostGIS geometry object
<b>ST_GeomFromKML<sup>3d</sup></b>	Takes as input KML representation of geometry and outputs a PostGIS geometry object
<b>ST_GMLToSQL<sup>2 mm</sup></b>	Return a specified ST_Geometry value from GML representation. This is an alias name for ST_GeomFromGML

<b>ST_GeomFromText<sup>mm</sup></b>	Return a specified ST_Geometry value from Well-Known Text representation (WKT).
<b>ST_GeomFromWKB<sup>mm</sup></b>	Creates a geometry instance from a Well-Known Binary geometry representation (WKB) and optional SRID.
<b>ST_LineFromMultiPoint<sup>3d</sup></b>	Creates a LineString from a MultiPoint geometry.
<b>ST_LineFromText<sup>mm</sup></b>	Makes a Geometry from WKT representation with the given SRID. If SRID is not given, it defaults to -1.
<b>ST_LineFromWKB<sup>mm</sup></b>	Makes a LINESTRING from WKB with the given SRID
<b>ST_LinestringFromWKB<sup>mm</sup></b>	Makes a geometry from WKB with the given SRID.



<b>ST_MakeBox2D</b>	Creates a BOX2D defined by the given point geometries.
<b>ST_3DMakeBox<sup>3d</sup></b>	Creates a BOX3D defined by the given 3d point geometries.
<b>ST_MakeLine<sup>3d</sup></b>	Creates a Linestring from point geometries.
<b>ST_MakeEnvelope<sup>2</sup></b>	Creates a rectangular Polygon formed from the given minimums and maximums. Input values must be in SRS specified by the SRID.
<b>ST_MakePolygon<sup>3d</sup></b>	Creates a Polygon formed by the given shell. Input geometries must be closed LINESRINGS.
<b>ST_MakePoint<sup>3d</sup></b>	Creates a 2D,3DZ or 4D point geometry.

<b>ST_MakePointM</b>	Creates a point geometry with an x y and m coordinate.
<b>ST_MLineFromText<sup>mm</sup></b>	Return a specified ST_MultiLineString value from WKT representation.
<b>ST_MPointFromText<sup>mm</sup></b>	Makes a Geometry from WKT with the given SRID. If SRID is not give, it defaults to -1.
<b>ST_MPolyFromText<sup>mm</sup></b>	Makes a MultiPolygon Geometry from WKT with the given SRID. If SRID is not give, it defaults to -1.
<b>ST_Point<sup>mm</sup></b>	Returns an ST_Point with the given coordinate values. OGC alias for ST_MakePoint.
<b>ST_PointFromText<sup>mm</sup></b>	Makes a point Geometry from WKT with the given SRID. If SRID is not given, it defaults to unknown.

<b>ST_PointFromWKB<sup>mm 3d</sup></b>	Makes a geometry from WKB with the given SRID
<b>ST_Polygon<sup>mm 3d</sup></b>	Returns a polygon built from the specified linestring and SRID.
<b>ST_PolygonFromText<sup>mm</sup></b>	Makes a Geometry from WKT with the given SRID. If SRID is not give, it defaults to -1.
<b>ST_WKBToSQL<sup>mm</sup></b>	Return a specified ST_Geometry value from Well-Known Binary representation (WKB). This is an alias name for ST_GeomFromWKB that takes no srId
<b>ST_WKTToSQL<sup>mm</sup></b>	Return a specified ST_Geometry value from Well-Known Text representation (WKT). This is an alias name for ST_GeomFromText
<b>GeometryType<sup>2 3d</sup></b>	Returns the type of the geometry as a string. Eg: 'LINESTRING', 'POLYGON', 'MULTIPOINT', etc.

<b>ST_Boundary<sup>mm 3d</sup></b>	Returns the closure of the combinatorial boundary of this Geometry.
<b>ST_CoordDim<sup>mm 3d</sup></b>	Return the coordinate dimension of the ST_Geometry value.
<b>ST_Dimension<sup>2 mm</sup></b>	The inherent dimension of this Geometry object, which must be less than or equal to the coordinate dimension.
<b>ST_EndPoint<sup>mm 3d</sup></b>	Returns the last point of a LINESTRING geometry as a POINT.
<b>ST_Envelope<sup>mm</sup></b>	Returns a geometry representing the double precision (float8) bounding box of the supplied geometry.
<b>ST_ExteriorRing<sup>mm 3d</sup></b>	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

<b>ST_GeometryN</b> <sup>2 mm 3d</sup>	Return the 1-based Nth geometry if the geometry is a GEOMETRYCOLLECTION, (MULTI)POINT, (MULTI)LINestring, MULTICURVE or (MULTI)POLYGON, POLYHEDRALSURFACE Otherwise, return NULL.
<b>ST_GeometryType</b> <sup>2 mm 3d</sup>	Return the geometry type of the ST_Geometry value.
<b>ST_InteriorRingN</b> <sup>mm 3d</sup>	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.
<b>ST_IsClosed</b> <sup>2 mm 3d</sup>	Returns TRUE if the LINestring's start and end points are coincident. For Polyhedral surface is closed (volumetric).
<b>ST_IsCollection</b> <sup>3d</sup>	Returns TRUE if the argument is a collection (MULTI*, GEOMETRYCOLLECTION, ...)
<b>ST_IsEmpty</b> <sup>mm</sup>	Returns true if this Geometry is an empty geometrycollection, polygon, point etc.

<b>ST_IsRing<sup>mm</sup></b>	Returns TRUE if this LINESTRING is both closed and simple.
<b>ST_IsSimple<sup>mm 3d</sup></b>	Returns (TRUE) if this Geometry has no anomalous geometric points, such as self intersection or self tangency.
<b>ST_IsValid<sup>mm g3.3</sup></b>	Returns true if the ST_Geometry is well formed.
<b>ST_IsValidReason<sup>1 g3.3</sup></b>	Returns text stating if a geometry is valid or not and if not valid, a reason why.
<b>ST_IsValidDetail<sup>1 g3.3</sup></b>	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.
<b>ST_M<sup>mm 3d</sup></b>	Return the M coordinate of the point, or NULL if not available. Input must be a point.

<b>ST_NDims</b> <sup>3d</sup>	Returns coordinate dimension of the geometry as a small int. Values are: 2,3 or 4.
<b>ST_NPoints</b> <sup>2 3d</sup>	Return the number of points (vertexes) in a geometry.
<b>ST_NRings</b> <sup>3d</sup>	If the geometry is a polygon or multi-polygon returns the number of rings.
<b>ST_NumGeometries</b> <sup>2 mm 3d</sup>	If geometry is a GEOMETRYCOLLECTION (or MULTI*) return the number of geometries, for single geometries will return 1, otherwise return NULL.
<b>ST_NumInteriorRings</b> <sup>mm</sup>	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.
<b>ST_NumInteriorRing</b> <sup>mm</sup>	Return the number of interior rings of the first polygon in the geometry. Synonym to ST_NumInteriorRings.

<b>ST_NumPatches</b> <sup>1 mm 3d</sup>	Return the number of faces on a Polyhedral Surface. Will return null for non-polyhedral geometries.
<b>ST_NumPoints</b> <sup>mm</sup>	Return the number of points in an ST_LineString or ST_CircularString value.
<b>ST_PatchN</b> <sup>1 mm 3d</sup>	Return the 1-based Nth geometry (face) if the geometry is a POLYHEDRALSURFACE, POLYHEDRALSURFACEM. Otherwise, return NULL.
<b>ST_PointN</b> <sup>mm 3d</sup>	Return the Nth point in the first linestring or circular linestring in the geometry. Return NULL if there is no linestring in the geometry.
<b>ST_SRID</b> <sup>mm</sup>	Returns the spatial reference identifier for the ST_Geometry as defined in spatial_ref_sys table.
<b>ST_StartPoint</b> <sup>mm 3d</sup>	Returns the first point of a LINESTRING geometry as a POINT.



<b>ST_Summary<sup>3d</sup></b>	Returns a text summary of the contents of the ST_Geometry.
<b>ST_X<sup>mm 3d</sup></b>	Return the X coordinate of the point, or NULL if not available. Input must be a point.
<b>ST_XMax<sup>3d</sup></b>	Returns X maxima of a bounding box 2d or 3d or a geometry.
<b>ST_XMin<sup>3d</sup></b>	Returns X minima of a bounding box 2d or 3d or a geometry.
<b>ST_Y<sup>mm 3d</sup></b>	Return the Y coordinate of the point, or NULL if not available. Input must be a point.
<b>ST_YMax<sup>3d</sup></b>	Returns Y maxima of a bounding box 2d or 3d or a geometry.

<b>ST_YMin<sup>3d</sup></b>	Returns Y minima of a bounding box 2d or 3d or a geometry.
<b>ST_Z<sup>mm</sup>3d</b>	Return the Z coordinate of the point, or NULL if not available. Input must be a point.
<b>ST_ZMax<sup>3d</sup></b>	Returns Z minima of a bounding box 2d or 3d or a geometry.
<b>ST_Zmflag<sup>3d</sup></b>	Returns ZM (dimension semantic) flag of the geometries as a small int. Values are: 0=2d, 1=3dm, 2=3dz, 3=4d.
<b>ST_ZMin<sup>3d</sup></b>	Returns Z minima of a bounding box 2d or 3d or a geometry.
<b>ST_AddPoint<sup>3d</sup></b>	Adds a point to a LineString before point position (0-based index).

<b>ST_Affine<sup>2 3d</sup></b>	Applies a 3d affine transformation to the geometry to do things like translate, rotate, scale in one step.
<b>ST_Force_2D<sup>2 3d</sup></b>	Forces the geometries into a "2-dimensional mode" so that all output representations will only have the X and Y coordinates.
<b>ST_Force_3D<sup>2 3d</sup></b>	Forces the geometries into XYZ mode. This is an alias for ST_Force_3DZ.
<b>ST_Force_3DZ<sup>2 3d</sup></b>	Forces the geometries into XYZ mode. This is a synonym for ST_Force_3D.
<b>ST_Force_3DM</b>	Forces the geometries into XYM mode.
<b>ST_Force_4D<sup>3d</sup></b>	Forces the geometries into XYZM mode.

<b>ST_Force_Collection</b> <sup>2 3d</sup>	Converts the geometry into a GEOMETRYCOLLECTION.
<b>ST_ForceRHR</b> <sup>2 3d</sup>	Forces the orientation of the vertices in a polygon to follow the Right-Hand-Rule.
<b>ST_LineMerge</b>	Returns a (set of) LineString(s) formed by sewing together a MULTILINESTRING.
<b>ST_CollectionExtract</b>	Given a GEOMETRYCOLLECTION, returns a MULTI* geometry consisting only of the specified type. Sub-geometries that are not the specified type are ignored. If there are no sub-geometries of the right type, an EMPTY collection will be returned. Only points, lines and polygons are supported. Type numbers are 1 == POINT, 2 == LINESTRING, 3 == POLYGON.
<b>ST_Multi</b>	Returns the geometry as a MULTI* geometry. If the geometry is already a MULTI*, it is returned unchanged.
<b>ST_RemovePoint</b> <sup>3d</sup>	Removes point from a linestring. Offset is 0-based.

<b>ST_Reverse</b>	Returns the geometry with vertex order reversed.
<b>ST_Rotate<sup>2 3d</sup></b>	This is a synonym for ST_RotateZ
<b>ST_RotateX<sup>2 3d</sup></b>	Rotate a geometry rotRadians about the X axis.
<b>ST_RotateY<sup>2 3d</sup></b>	Rotate a geometry rotRadians about the Y axis.
<b>ST_RotateZ<sup>2 3d</sup></b>	Rotate a geometry rotRadians about the Z axis.
<b>ST_Scale<sup>2 3d</sup></b>	Scales the geometry to a new size by multiplying the ordinates with the parameters. I.e: ST_Scale(geom, Xfactor, Yfactor, Zfactor).

<b>ST_Segmentize</b>	Return a modified geometry having no segment longer than the given distance. Distance computation is performed in 2d only.
<b>ST_SetPoint<sup>3d</sup></b>	Replace point N of linestring with given point. Index is 0-based.
<b>ST_SetSRID</b>	Sets the SRID on a geometry to a particular integer value.
<b>ST_SnapToGrid<sup>3d</sup></b>	Snap all points of the input geometry to a regular grid.
<b>ST_Snap<sup>1 g3.3</sup></b>	Snap segments and vertices of input geometry to vertices of a reference geometry.
<b>ST_Transform<sup>2 mm</sup></b>	Returns a new geometry with its coordinates transformed to the SRID referenced by the integer parameter.

<b>ST_Translate<sup>3d</sup></b>	Translates the geometry to a new location using the numeric parameters as offsets. I.e: ST_Translate(geom, X, Y) or ST_Translate(geom, X, Y,Z).
<b>ST_TransScale<sup>3d</sup></b>	Translates the geometry using the deltaX and deltaY args, then scales it using the XFactor, YFactor args, working in 2D only.
<b>ST_AsBinary<sup>2 mm G</sup></b>	Return the Well-Known Binary (WKB) representation of the geometry/geography without SRID meta data.
<b>ST_AsEWKB<sup>2 3d</sup></b>	Return the Well-Known Binary (WKB) representation of the geometry with SRID meta data.
<b>ST_AsEWKT<sup>2 3d</sup></b>	Return the Well-Known Text (WKT) representation of the geometry with SRID meta data.
<b>ST_AsGeoJSON<sup>G 3d</sup></b>	Return the geometry as a GeoJSON element.

<b>ST_AsGML<sup>2 G 3d</sup></b>	Return the geometry as a GML version 2 or 3 element.
<b>ST_AsHEXEWKB<sup>3d</sup></b>	Returns a Geometry in HEXEWKB format (as text) using either little-endian (NDR) or big-endian (XDR) encoding.
<b>ST_AsKML<sup>2 G 3d</sup></b>	Return the geometry as a KML element. Several variants. Default version=2, default precision=15
<b>ST_AsSVG<sup>G</sup></b>	Returns a Geometry in SVG path data given a geometry or geography object.
<b>ST_AsX3D<sup>1 3d</sup></b>	Returns a Geometry in X3D xml node element format: ISO-IEC-19776-1.2-X3DEncodings-XML
<b>ST_GeoHash</b>	Return a GeoHash representation (geohash.org) of the geometry.



<p><b>ST_AsText<sup>mm G</sup></b></p>	<p>Return the Well-Known Text (WKT) representation of the geometry/geography without SRID metadata.</p>
<p><b>ST_AsLatLonText<sup>1</sup></b></p>	<p>Return the Degrees, Minutes, Seconds representation of the given point.</p>
<p><b>&amp;&amp;<sup>2 G</sup></b></p>	<p>Returns TRUE if A's 2D bounding box intersects B's 2D bounding box.</p>
<p><b>&amp;&amp;&amp;<sup>1 3d</sup></b></p>	<p>Returns TRUE if A's 3D bounding box intersects B's 3D bounding box.</p>
<p><b>&amp;&lt;</b></p>	<p>Returns TRUE if A's bounding box overlaps or is to the left of B's.</p>
<p><b>&amp;&lt; </b></p>	<p>Returns TRUE if A's bounding box overlaps or is below B's.</p>

<b>&amp;&gt;</b>	Returns TRUE if A' bounding box overlaps or is to the right of B's.
<b>&lt;&lt;</b>	Returns TRUE if A's bounding box is strictly to the left of B's.
<b>&lt;&lt; </b>	Returns TRUE if A's bounding box is strictly below B's.
<b>=G</b>	Returns TRUE if A's bounding box is the same as B's. Uses double precision bounding box.
<b>&gt;&gt;</b>	Returns TRUE if A's bounding box is strictly to the right of B's.
<b>@</b>	Returns TRUE if A's bounding box is contained by B's.

&>	Returns TRUE if A's bounding box overlaps or is above B's.
>>	Returns TRUE if A's bounding box is strictly above B's.
~	Returns TRUE if A's bounding box contains B's.
~=	Returns TRUE if A's bounding box is the same as B's.
↔ <sup>1</sup>	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.
↔ <sup>#</sup> <sup>1</sup>	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.

<b>ST_3DClosestPoint</b> <sup>1 3d</sup>	Returns the 3-dimensional point on g1 that is closest to g2. This is the first point of the 3D shortest line.
<b>ST_3DDistance</b> <sup>1 mm 3d</sup>	For geometry type Returns the 3-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units.
<b>ST_3DDWithin</b> <sup>1 mm 3d</sup>	For 3d (z) geometry type Returns true if two geometries 3d distance is within number of units.
<b>ST_3DDFullyWithin</b> <sup>1 3d</sup>	Returns true if all of the 3D geometries are within the specified distance of one another.
<b>ST_3DIntersects</b> <sup>1 mm 3d</sup>	Returns TRUE if the Geometries "spatially intersect" in 3d - only for points and linestrings
<b>ST_3DLongestLine</b> <sup>1 3d</sup>	Returns the 3-dimensional longest line between two geometries

<b>ST_3DMaxDistance</b> <sup>1 3d</sup>	For geometry type Returns the 3-dimensional cartesian maximum distance (based on spatial ref) between two geometries in projected units.
<b>ST_3DShortestLine</b> <sup>1 3d</sup>	Returns the 3-dimensional shortest line between two geometries
<b>ST_Area</b> <sup>2 mm G</sup>	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.
<b>ST_Azimuth</b> <sup>2 G</sup>	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/4.
<b>ST_Centroid</b> <sup>mm</sup>	Returns the geometric center of a geometry.
<b>ST_ClosestPoint</b>	Returns the 2-dimensional point on g1 that is closest to g2. This is the first point of the shortest line.

<b>ST_Contains<sup>mm</sup></b>	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.
<b>ST_ContainsProperly</b>	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.
<b>ST_Covers<sup>G</sup></b>	Returns 1 (TRUE) if no point in Geometry B is outside Geometry A
<b>ST_CoveredBy<sup>G</sup></b>	Returns 1 (TRUE) if no point in Geometry/Geography A is outside Geometry/Geography B
<b>ST_Crosses<sup>mm</sup></b>	Returns TRUE if the supplied geometries have some, but not all, interior points in common.
<b>ST_LineCrossingDirection</b>	Given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behavior. 0 is no crossing.

<b>ST_Disjoint<sup>mm</sup></b>	Returns TRUE if the Geometries do not "spatially intersect" - if they do not share any space together.
<b>ST_Distance<sup>mm G</sup></b>	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.
<b>ST_HausdorffDistance</b>	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.
<b>ST_MaxDistance</b>	Returns the 2-dimensional largest distance between two geometries in projected units.
<b>ST_Distance_Sphere</b>	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.
<b>ST_Distance_Spheroid</b>	Returns the minimum distance between two lon/lat geometries given a particular spheroid. PostGIS versions prior to 1.5 only support points.

<b>ST_DFullyWithin</b>	Returns true if all of the geometries are within the specified distance of one another
<b>ST_DWithin<sup>G</sup></b>	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.
<b>ST_Equals<sup>mm</sup></b>	Returns true if the given geometries represent the same geometry. Directionality is ignored.
<b>ST_HasArc<sup>3d</sup></b>	Returns true if a geometry or geometry collection contains a circular string
<b>ST_Intersects<sup>mm G</sup></b>	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)
<b>ST_Length<sup>mm G</sup></b>	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)



<b>ST_Length2D</b>	Returns the 2-dimensional length of the geometry if it is a linestring or multi-linestring. This is an alias for ST_Length
<b>ST_3DLength<sup>3d</sup></b>	Returns the 3-dimensional or 2-dimensional length of the geometry if it is a linestring or multi-linestring.
<b>ST_Length_Spheroid<sup>3d</sup></b>	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.
<b>ST_Length2D_Spheroid</b>	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.
<b>ST_3DLength_Spheroid<sup>3d</sup></b>	Calculates the length of a geometry on an ellipsoid, taking the elevation into account. This is just an alias for ST_Length_Spheroid.
<b>ST_LongestLine</b>	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.

<b>ST_OrderingEquals<sup>mm</sup></b>	Returns true if the given geometries represent the same geometry and points are in the same directional order.
<b>ST_Overlaps<sup>mm</sup></b>	Returns TRUE if the Geometries share space, are of the same dimension, but are not completely contained by each other.
<b>ST_Perimeter<sup>mm</sup></b>	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.
<b>ST_Perimeter2D</b>	Returns the 2-dimensional perimeter of the geometry, if it is a polygon or multi-polygon. This is currently an alias for ST_Perimeter.
<b>ST_3DPerimeter<sup>3d</sup></b>	Returns the 3-dimensional perimeter of the geometry, if it is a polygon or multi-polygon.
<b>ST_PointOnSurface<sup>mm 3d</sup></b>	Returns a POINT guaranteed to lie on the surface.

<b>ST_Project<sup>G</sup></b>	Returns a POINT projected from a start point using a bearing and distance.
<b>ST_Relate<sup>2 mm</sup></b>	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.
<b>ST_RelateMatch<sup>1 g3.3</sup></b>	Returns true if intersectionMatrixPattern1 implies intersectionMatrixPattern2
<b>ST_ShortestLine</b>	Returns the 2-dimensional shortest line between two geometries
<b>ST_Touches<sup>mm</sup></b>	Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.
<b>ST_Within<sup>mm</sup></b>	Returns true if the geometry A is completely inside geometry B

<p><b>ST_Buffer<sup>mm G</sup></b></p>	<p>(T) For geometry: Returns a geometry that represents all points whose distance from this Geometry is less than or equal to distance. Calculations are in the Spatial Reference System of this Geometry. For geography: Uses a planar transform wrapper. Introduced in 1.5 support for different end cap and mitre settings to control shape. <code>buffer_style</code> options: <code>quad_segs=#,endcap=round flat square,join=round mitre bevel,mitre_limit=</code></p>
<p><b>ST_BuildArea</b></p>	<p>Creates an areal geometry formed by the constituent linework of given geometry</p>
<p><b>ST_Collect<sup>3d</sup></b></p>	<p>Return a specified ST_Geometry value from a collection of other geometries.</p>
<p><b>ST_ConcaveHull<sup>1</sup></b></p>	<p>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 wrapping.</p>
<p><b>ST_ConvexHull<sup>mm 3d</sup></b></p>	<p>The convex hull of a geometry represents the minimum convex geometry that encloses all geometries within the set.</p>
<p><b>ST_CurveToLine<sup>mm 3d</sup></b></p>	<p>Converts a CIRCULARSTRING/CURVEDPOLYGON to a LINESTRING/POLYGON</p>

<b>ST_Difference</b> <sup>mm 3d</sup>	Returns a geometry that represents that part of geometry A that does not intersect with geometry B.
<b>ST_Dump</b> <sup>2 3d</sup>	Returns a set of geometry_dump (geom,path) rows, that make up a geometry g1.
<b>ST_DumpPoints</b> <sup>2 3d</sup>	Returns a set of geometry_dump (geom,path) rows of all points that make up a geometry.
<b>ST_DumpRings</b> <sup>3d</sup>	Returns a set of geometry_dump rows, representing the exterior and interior rings of a polygon.
<b>ST_FlipCoordinates</b> <sup>1 3d</sup>	Returns a version of the given geometry with X and Y axis flipped. Useful for people who have built latitude/longitude features and need to fix them.
<b>ST_Intersection</b> <sup>mm G</sup>	(T) Returns a geometry that represents the shared portion of geomA and geomB. The geography implementation does a transform to geometry to do the intersection and then transform back to WGS84.

<b>ST_LineToCurve<sup>3d</sup></b>	Converts a LINESTRING/POLYGON to a CIRCULARSTRING, CURVED POLYGON
<b>ST_MakeValid<sup>1 3d</sup></b>	Attempts to make an invalid geometry valid w/out losing vertices.
<b>ST_MemUnion<sup>3d</sup></b>	Same as ST_Union, only memory-friendly (uses less memory and more processor time).
<b>ST_MinimumBoundingCircle</b>	Returns the smallest circle polygon that can fully contain a geometry. Default uses 48 segments per quarter circle.
<b>ST_Polygonize</b>	Aggregate. Creates a GeometryCollection containing possible polygons formed from the constituent linework of a set of geometries.
<b>ST_Node<sup>1 g3.3 3d</sup></b>	Node a set of linestrings.

<b>ST_OffsetCurve</b> <sup>1 g3.3</sup>	Return an offset line at a given distance and side from an input line. Useful for computing parallel lines about a center line
<b>ST_RemoveRepeatedPoints</b> <sup>1 3d</sup>	Returns a version of the given geometry with duplicated points removed.
<b>ST_SharedPaths</b> <sup>1 g3.3</sup>	Returns a collection containing paths shared by the two input linestrings/multilinestrings.
<b>ST_Shift_Longitude</b> <sup>2 3d</sup>	Reads every point/vertex in every component of every feature in a geometry, and if the longitude coordinate is 0, adds 360 to it. The result would be a 0-360 version of the data to be plotted in a 180 centric map
<b>ST_Simplify</b>	Returns a "simplified" version of the given geometry using the Douglas-Peucker algorithm.
<b>ST_SimplifyPreserveTopology</b>	Returns a "simplified" version of the given geometry using the Douglas-Peucker algorithm. Will avoid creating derived geometries (polygons in particular) that are invalid.

<b>ST_Split<sup>1</sup></b>	Returns a collection of geometries resulting by splitting a geometry.
<b>ST_SymDifference<sup>mm 3d</sup></b>	Returns a geometry that represents the portions of A and B that do not intersect. It is called a symmetric difference because $ST\_SymDifference(A,B) = ST\_SymDifference(B,A)$ .
<b>ST_Union<sup>mm</sup></b>	Returns a geometry that represents the point set union of the Geometries.
<b>ST_UnaryUnion<sup>1 g3.3 3d</sup></b>	Like ST_Union, but working at the geometry component level.
<b>ST_Line_Interpolate_Point<sup>3d</sup></b>	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.
<b>ST_Line_Locate_Point</b>	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.



<b>ST_Line_Substring<sup>3d</sup></b>	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.
<b>ST_Locate_Along_Measure</b>	Return a derived geometry collection value with elements that match the specified measure. Polygonal elements are not supported.
<b>ST_Locate_Between_Measures</b>	Return a derived geometry collection value with elements that match the specified range of measures inclusively. Polygonal elements are not supported.
<b>ST_LocateBetweenElevations<sup>3d</sup></b>	Return a derived geometry (collection) value with elements that intersect the specified range of elevations inclusively. Only 3D, 4D LINESTRINGS and MULTILINESTRINGS are supported.
<b>ST_AddMeasure<sup>3d</sup></b>	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.
<b>AddAuth</b>	Add an authorization token to be used in current transaction.

<b>CheckAuth</b>	Creates trigger on a table to prevent/allow updates and deletes of rows based on authorization token.
<b>DisableLongTransactions</b>	Disable long transaction support. This function removes the long transaction support metadata tables, and drops all triggers attached to lock-checked tables.
<b>EnableLongTransactions</b>	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.
<b>LockRow</b>	Set lock/authorization for specific row in table
<b>UnlockRows</b>	Remove all locks held by specified authorization id. Returns the number of locks released.
<b>ST_Accum<sup>2 3d</sup></b>	Aggregate. Constructs an array of geometries.

<b>Box2D<sup>2</sup></b>	Returns a BOX2D representing the maximum extents of the geometry.
<b>Box3D<sup>2 3d</sup></b>	Returns a BOX3D representing the maximum extents of the geometry.
<b>ST_Estimated_Extent</b>	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.
<b>ST_Expand<sup>2</sup></b>	Returns bounding box expanded in all directions from the bounding box of the input geometry. Uses double-precision
<b>ST_Extent<sup>2</sup></b>	an aggregate function that returns the bounding box that bounds rows of geometries.
<b>ST_3DExtent<sup>2 3d</sup></b>	an aggregate function that returns the box3D bounding box that bounds rows of geometries.

<b>Find_SRID</b>	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.
<b>ST_Mem_Size<sup>3d</sup></b>	Returns the amount of space (in bytes) the geometry takes.
<b>ST_Point_Inside_Circle</b>	Is the point geometry inside circle defined by <code>center_x</code> , <code>center_y</code> , <code>radius</code>
<b>PostGIS_AddBBox</b>	Add bounding box to the geometry.
<b>PostGIS_DropBBox</b>	Drop the bounding box cache from the geometry.
<b>PostGIS_HasBBox</b>	Returns <code>TRUE</code> if the <code>bbox</code> of this geometry is cached, <code>FALSE</code> otherwise.

<b>geomval</b>	A spatial datatype with two fields - geom (holding a geometry object) and val (holding a double precision pixel value from a raster band)
<b>histogram</b>	A composite type used as record output of the ST_Histogram and ST_ApproxHistogram functions.
<b>raster</b>	raster spatial data type.
<b>reclassarg</b>	A composite type used as input into the ST_Reclass function defining the behavior of reclassification.
<b>summarystats</b>	A composite type used as output of the ST_SummaryStats function.
<b>AddRasterConstraints<sup>1</sup></b>	Adds raster constraints to a loaded raster table for a specific column that constrains spatial ref, scaling, blocksize, alignment, bands, band type and a flag to denote if raster column is regularly blocked. The table must be loaded with data for the constraints to be inferred. Returns true if the constraint setting was accomplished and if issues a notice.

<b>DropRasterConstraints<sup>1</sup></b>	Drops PostGIS raster constraints that refer to a raster table column. Useful if you need to reload data or update your raster column data.
<b>PostGIS_Raster_Lib_Build_Date</b>	Reports full raster library build date
<b>PostGIS_Raster_Lib_Version</b>	Reports full raster version and build configuration infos.
<b>ST_GDALDrivers<sup>1</sup></b>	Returns a list of raster formats supported by your lib gdal. These are the formats you can output your raster using ST_AsGDALRaster.
<b>ST_AddBand</b>	Returns a raster with the new band(s) of given type added with given initial value in the given index location. If no index is specified, the band is added to the end.
<b>ST_AsRaster<sup>1</sup></b>	Converts a PostGIS geometry to a PostGIS raster.

<b>ST_Band<sup>1</sup></b>	Returns one or more bands of an existing raster as a new raster. Useful for building new rasters from existing rasters.
<b>ST_MakeEmptyRaster</b>	Returns an empty raster (having no bands) of given dimensions, with upperleft x y, pixel size expressed as scalex, scaley, skewx, skewy and reference system (srid). If a raster is passed in, returns a new raster with same meta data properties. If srid is left out, the spatial ref is set to unknown.
<b>ST_GeoReference</b>	Returns the georeference meta data in GDAL or ESRI format as commonly seen in a world file. Default is GDAL.
<b>ST_Height</b>	Returns the height of the raster in pixels?
<b>ST_MetaData</b>	Returns basic meta data about a raster object such as skew, rotation, upper,lower left etc.
<b>ST_NumBands</b>	Returns the number of bands in the raster object.

<b>ST_PixelHeight</b>	Returns the pixel height in geometric units of the spatial reference system
<b>ST_PixelWidth</b>	Returns the pixel width in geometric units of the spatial reference system
<b>ST_ScaleX</b>	Returns the x size of pixels in units of coordinate reference system?
<b>ST_ScaleY</b>	Returns the y size of pixels in units of coordinate reference system?
<b>ST_Raster2WorldCoordX</b>	Returns the geometric x coordinate upper left of a raster, column and row. Numbering of columns and rows starts at 1.
<b>ST_Raster2WorldCoordY</b>	Returns the geometric y coordinate upper left corner of a raster, column and row. Numbering of columns and rows starts at 1.



<b>ST_Rotation</b>	Returns the rotation of the raster
<b>ST_SkewX</b>	Returns the georeference X skew (or rotation parameter)
<b>ST_SkewY</b>	Returns the georeference Y skew (or rotation parameter)
<b>ST_SRID</b>	Returns the spatial reference identifier of the raster as defined in spatial_ref_sys table.
<b>ST_UpperLeftX</b>	Returns the upper left x coordinate of raster in projected spatial ref.
<b>ST_UpperLeftY</b>	Returns the upper left y coordinate of raster in projected spatial ref.

<b>ST_Width</b>	Returns the width of the raster in pixels?
<b>ST_World2RasterCoordX</b>	Returns the column in the raster of the point geometry (pt) or a x y world coordinate (xw, yw) represented in world spatial reference system of raster.
<b>ST_World2RasterCoordY</b>	Returns the row in the raster of the point geometry (pt) or a x y world coordinate (xw, yw) represented in world spatial reference system of raster.
<b>ST_IsEmpty</b>	Returns true if the raster is empty (width = 0 and height = 0). Otherwise, returns false.
<b>ST_BandMetaData</b>	Returns basic meta data for a specific raster band. band num 1 is assumed if none-specified.
<b>ST_BandNoDataValue</b>	Returns the value in a given band that represents no data. If no band num 1 is assumed.

<b>ST_BandIsNoData<sup>1</sup></b>	Returns true if the band is filled with only nodata values.
<b>ST_BandPath</b>	Returns system file path to a band stored in file system. If no bandnum specified, 1 is assumed.
<b>ST_BandPixelType</b>	Returns the type of pixel for given band. If no bandnum specified, 1 is assumed.
<b>ST_HasNoBand<sup>1</sup></b>	Returns true if there is no band with given band number. If no band number is specified, then band number 1 is assumed.
<b>ST_PixelAsPolygon</b>	Returns the geometry that bounds the pixel (for now a rectangle) for a particular band, row, column. If no band specified band num 1 is assumed.
<b>ST_Value<sup>2</sup></b>	Returns the value of a given band in a given columnx, rowy pixel or at a particular geometric point. Band numbers start at 1 and assumed to be 1 if not specified. If exclude_nodata_value is set to false, then all pixels include nodata pixels are considered to intersect and return value. If exclude_nodata_value is not passed in then reads it from metadata of raster.

<b>ST_SetValue</b>	Returns modified raster resulting from setting the value of a given band in a given columnx, rowy pixel or at a pixel that intersects a particular geometric point. Band numbers start at 1 and assumed to be 1 if not specified.
<b>ST_SetGeoReference</b>	Set Georeference 6 georeference parameters in a single call. Numbers should be separated by white space. Accepts inputs in GDAL or ESRI format. Default is GDAL.
<b>ST_SetRotation</b>	Set the rotation of the raster.
<b>ST_SetScale</b>	Sets the x and y size of pixels in units of coordinate reference system. Number units/pixel width/height
<b>ST_SetSkew</b>	Sets the georeference X and Y skew (or rotation parameter). If only one is passed in sets x and y to same number.
<b>ST_SetSRID</b>	Sets the SRID of a raster to a particular integer srid defined in the spatial_ref_sys table.

<b>ST_SetUpperLeft</b>	Sets the value of the upper left corner of the pixel to projected x,y coordinates
<b>ST_Resample<sup>1</sup></b>	Warp a raster in a known spatial reference system using a specified resampling algorithm and set of raster attributes or reference raster. If using a reference raster, the new raster will have the same srid, skew, scale and alignment. Uses NearestNeighbor if no algorithm is specified Options: NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline, Lanczos.
<b>ST_Rescale<sup>1</sup></b>	Rescales a raster that is in a known spatial ref and using specified algorithm. Uses NearestNeighbor if no algorithm is specified Options: NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline, Lanczos. the scalex, scaley, scalexy are the desired new ratio between pixel coordinates and geometry coordinates
<b>ST_Transform<sup>1</sup></b>	Reprojects a raster in a known spatial reference system to another known spatial reference system using specified resampling algorithm. Uses NearestNeighbor if no algorithm is specified Options: NearestNeighbor, Bilinear, Cubic, CubicSpline, Lanczos.
<b>ST_SetBandNoDataValue</b>	Sets the value for the given band that represents no data. Band 1 is assumed if no band is specified. To mark a band as having no nodata value, set the nodata value = NULL
<b>ST_SetBandIsNoData<sup>1</sup></b>	Sets the isnodata flag of the band to TRUE. You may want to call this function if ST_BandIsNoData(rast, band) != ST_BandIsNodata(rast, band, TRUE). This is, if the isnodata flag is dirty. Band 1 is assumed if no band is specified.

<b>ST_Count<sup>1</sup></b>	Returns the number of pixels in a given band of a raster or raster coverage. If no band is specified defaults to band 1. If <code>exclude_nodata_value</code> is set to true, will only count pixels that are not equal to the nodata value.
<b>ST_Histogram<sup>1</sup></b>	Returns a set of histogram summarizing a raster or raster coverage data distribution separate bin ranges. Number of bins are autocomputed if not specified.
<b>ST_Quantile<sup>1</sup></b>	Compute quantiles in the context of the sample or population. Thus, a value could be examined to be at the raster's 25%, 50%, 75% percentile.
<b>ST_SummaryStats<sup>1</sup></b>	Returns summary stats consisting of count,sum,mean,stddev,min,max for a given raster band of a raster or raster coverage. Band 1 is assumed is no band is specified.
<b>ST_ValueCount<sup>1</sup></b>	Returns a set of records containing a pixel band value and count of the number of pixels in a given band of a raster (or a raster coverage) that have a given set of values. If no band is specified defaults to band 1. By default nodata value pixels are not counted. and all other values in the pixel are output and pixel band values are rounded to the nearest integer.
<b>ST_AsBinary</b>	Return the Well-Known Binary (WKB) representation of the raster without SRID meta data.

<b>ST_AsGDALRaster<sup>1</sup></b>	Return the raster tile in the designated GDAL Raster format. Raster formats are one of those supported by your compiled library. Use ST_GDALRasters() to get a list of formats supported by your library.
<b>ST_AsJPEG<sup>1</sup></b>	Return the raster tile selected bands as a single Joint Photographic Exports Group (JPEG) image (byte array). If no band is specified and 1 or more than 3 bands, then only the first band is used. If only 3 bands then all 3 bands are used and mapped to RGB.
<b>ST_AsPNG<sup>1</sup></b>	Return the raster tile selected bands as a single portable network graphics (PNG) image (byte array). If 1,3, or 4 bands in raster and no bands are specified, then all bands are used. If more 2 or more than 4 bands and no bands specified, then only band 1 is used. Bands are mapped to RGB or RGBA space.
<b>ST_AsTIFF<sup>1</sup></b>	Return the raster tile selected bands as a single TIFF image (byte array). If no band is specified, then will try to use all bands?.
<b>Box3D</b>	Returns the box 3d representation of the enclosing box of the raster
<b>ST_Clip<sup>1</sup></b>	Returns a single band raster that is a band clipped by the input geometry. If no band is specified band 1 is assumed. If trimraster setting is not specified, false is assumed meaning the output raster has the same extent as the input raster.

<b>ST_ConvexHull</b>	Return the convex hull geometry of the raster including pixel values equal to BandNoDataValue. For regular shaped and non-skewed rasters, this gives the same answer as ST_Envelope so only useful for irregularly shaped or skewed rasters.
<b>ST_DumpAsPolygons</b>	Returns a set of geomval (geom,val) rows, from a given raster band. If no band number is specified, band num defaults to 1.
<b>ST_Envelope</b>	Returns the polygon representation of the extent of the raster.
<b>ST_HillShade<sup>1</sup></b>	Returns the hypothetical illumination of a raster band using provided azimuth, altitude, brightness, and elevation scale inputs. Useful for visualizing terrain.
<b>ST_Intersection<sup>2</sup></b>	Returns a set of geometry-pixelvalue pairs resulting from intersection of a raster band with a geometry when operation is geometry,raster. If operation is raster,geometry then a raster is returned that is result of the input raster clipped by geometry. If no band number is specified, band 1 is assumed.
<b>ST_MapAlgebraExpr<sup>1</sup></b>	1 raster band version: Creates a new one band raster formed by applying a valid PostgreSQL algebraic operation on the input raster band and of pixeltypes provided. band 1 is assumed if no band is specified.



<b>ST_MapAlgebraExpr<sup>1</sup></b>	2 raster band version: Creates a new one band raster formed by applying a valid PostgreSQL algebraic operation on the 2 input raster bands and of pixeltype provided. band 1 of each raster is assumed if no band numbers are specified. The resulting raster will be aligned (scale, skew and pixel corners) on the grid defined by the first raster. The resulting raster will have the extent defined by the "extenttype" parameter. Values for "extenttype" can be: INTERSECTION, UNION, FIRST, SECOND
<b>ST_MapAlgebraFct<sup>1</sup></b>	1 band version - Creates a new one band raster formed by applying a valid PostgreSQL function on the input raster band and of pixeltype provided. Band 1 is assumed if no band is specified.
<b>ST_MapAlgebraFct<sup>1</sup></b>	2 band version - Creates a new one band raster formed by applying a valid PostgreSQL function on the 2 input raster bands and of pixeltype provided. Band 1 is assumed if no band is specified. Extent type defaults to INTERSECTION if not specified
<b>ST_MapAlgebraFctNgb<sup>1</sup></b>	1-band version: Map Algebra Nearest Neighbor using user-defined PostgreSQL function. Return a raster which values are the result of a PLPGSQL user function involving a neighborhood of values from the input raster band.
<b>ST_Polygon</b>	Returns a polygon geometry formed by the union of pixels that have a pixel value that is not no data value. If no band number is specified, band num defaults to 1.
<b>ST_Reclass<sup>1</sup></b>	Creates a new raster composed of band types reclassified from original. The nband is the band to be changed. If nband is not specified assumed to be 1. All other bands are returned unchanged. Use case: convert a 16BUI band to a 8 8BUI and so forth for simpler rendering as viewable formats.

<b>ST_Union<sup>1</sup></b>	Returns the union of a set of raster tiles into a single raster composed of 1 band. If no band is specified for unioning, band num 1 is assumed. The resulting raster's extent is the extent of the whole set. In the case of intersection, the resulting value is defined by p_expression which is one of the following: LAST - the default when none is specified, MEAN, SUM, FIRST, MAX, MIN
<b>&amp;&amp;</b>	Returns TRUE if A's bounding box overlaps B's.
<b>&amp;&lt;</b>	Returns TRUE if A's bounding box is to the left of B's.
<b>&amp;&gt;</b>	Returns TRUE if A's bounding box is to the right of B's.
<b>ST_Intersects<sup>2</sup></b>	If band numbers are omitted only considers convex hull of raster. Returns true only if rast pixel in a band with non-nodata band value intersects with a geometry/raster.
<b>ST_SameAlignment<sup>1</sup></b>	Returns true if rasters have same skew,scale, spatial ref and false if they don't with notice detailing issue.

<b>getfaceedges_returntype</b>	A composite type that consists of a sequence number and edge number. This is the return type for ST_GetFaceEdges
<b>topogeometry</b>	A composite type that refers to a topology geometry in a specific topology, layer, having specific type (1:[multi]point, 2:[multi]line, 3:[multi]poly, 4:collection) with specific identifier id in the topology. The id uniquely defines the topogeometry in the topology.
<b>validatetopology_returntype</b>	A composite type that consists of an error message and id1 and id2 to denote location of error. This is the return type for ValidateTopology
<b>TopoElement</b>	An array of 2 integers generally used to identify a TopoGeometry component.
<b>topoelementarray</b>	An array of element_id,element_type values. a bidimensional array of integers: '{{id,type}, {id,type}, ...}'
<b>AddTopoGeometryColumn</b>	Adds a topogeometry column to an existing table, registers this new column as a layer in topology.layer and returns the new layer_id.

<b>CreateTopology</b>	Creates a new topology schema and registers this new schema in the topology.topology table.
<b>DropTopology</b>	Use with caution: Drops a topology schema and deletes its reference from topology.topology table and references to tables in that schema from the geometry_columns table.
<b>CopyTopology<sup>1</sup></b>	Makes a copy of a topology structure (nodes, edges, faces, layers and TopoGeometries).
<b>DropTopoGeometryColumn</b>	Drops the topogeometry column from the table named table_name in schema schema_name and unregisters the columns from topology.layer table.
<b>ST_InitTopoGeo<sup>mm</sup></b>	Creates a new topology schema and registers this new schema in the topology.topology table and details summary of process.
<b>TopologySummary<sup>1</sup></b>	Takes a topology name and provides summary totals of types of objects in topology

<b>ValidateTopology<sup>2</sup></b>	Returns a set of validate_topology_returntype objects detailing issues with topology
<b>ST_CreateTopoGeo<sup>1 mm</sup></b>	Adds a collection of geometries to a given empty topology and returns a message detailing success.
<b>ST_AddIsoNode<sup>mm</sup></b>	Adds an isolated node to a face in a topology and returns the nodeid of the new node. If face is null, the node is still created.
<b>ST_AddIsoEdge<sup>mm</sup></b>	Adds an isolated edge defined by geometry a linestring to a topology connecting two existing isolated nodes a node and another node and returns the edge id of the new edge.
<b>ST_AddEdgeNewFaces<sup>1 mm</sup></b>	Add a new edge and, if in doing so it splits a face, delete the original face and replace it with two new faces.
<b>ST_AddEdgeModFace<sup>1 mm</sup></b>	Add a new edge and, if in doing so it splits a face, modify the original face and add a new face.

<b>ST_RemEdgeNewFace<sup>1 mm</sup></b>	Removes an edge and, if the removed edge separated two faces, delete the original faces and replace them with a new face.
<b>ST_RemEdgeModFace<sup>1 mm</sup></b>	Removes an edge and, if the removed edge separated two faces, delete one of the them and modify the other to take the space of both.
<b>ST_ChangeEdgeGeom<sup>mm</sup></b>	Changes the linestring that define the specified edge. Will raise an error if the start and end points are not the same as the original or the new linestring crosses an existing edge (not at end points) or new linestring is not simple.
<b>ST_ModEdgeSplit<sup>mm</sup></b>	Split an edge by creating a new node along an existing edge, modifying the original edge and adding a new edge.
<b>ST_ModEdgeHeal<sup>1 mm</sup></b>	Heal two edges by deleting the node connecting them, modifying the first edge and deleting the second edge. Returns the id of the deleted node.
<b>ST_NewEdgeHeal<sup>1 mm</sup></b>	Heal two edges by deleting the node connecting them, deleting both edges, and replacing them with an edge whose direction is the same as the first edge provided.

<b>ST_MoveIsoNode<sup>mm</sup></b>	Moves an isolated node in a topology from one point to another. If new apoint geometry exists as a node an error is thrown. REturns description of move.
<b>ST_NewEdgesSplit<sup>mm</sup></b>	Split an edge by creating a new node along an existing edge, deleting the original edge and replacing it with two new edges. Returns the id of the new node created that joins the new edges.
<b>ST_RemoveIsoNode<sup>mm</sup></b>	Removes an isolated node and returns description of action. If the node is not isolated (is start or end of an edge), then an exception is thrown.
<b>GetEdgeByPoint<sup>1 g3.3</sup></b>	Find the edge-id of an edge that intersects a given point
<b>GetFaceByPoint<sup>1 g3.3</sup></b>	Find the face-id of a face that intersects a given point
<b>GetNodeByPoint<sup>1 g3.3</sup></b>	Find the id of a node at a point location

<b>GetTopologyID</b>	Returns the id of a topology in the topology.topology table given the name of the topology.
<b>GetTopologyName</b>	Returns the name of a topology (schema) given the id of the topology.
<b>ST_GetFaceEdges<sup>1 mm</sup></b>	Returns a set of ordered edges that bound a face includes the sequence order.
<b>ST_GetFaceGeometry<sup>mm</sup></b>	Returns the polygon in the given topology with the specified face id.
<b>TopoGeo_AddPoint<sup>1</sup></b>	Adds a point to an existing topology using a tolerance and possibly splitting an existing edge.
<b>TopoGeo_AddLineString<sup>1</sup></b>	Adds a linestring to an existing topology using a tolerance and possibly splitting existing edges.



<b>TopoGeo_AddPolygon<sup>1</sup></b>	Adds a polygon to an existing topology using a tolerance and possibly splitting existing edges/faces.
<b>Polygonize<sup>1</sup></b>	Find and register all faces defined by topology edges
<b>AddNode<sup>1</sup></b>	Adds a point node to the node table in the specified topology schema and returns the nodeid of new node. If point already exists as node, the existing nodeid is returned.
<b>AddEdge<sup>1 g3.3</sup></b>	Adds a linestring edge to the edge table and associated start and end points to the point nodes table of the specified topology schema using the specified linestring geometry and returns the edgeid of the new (or existing) edge.
<b>AddFace<sup>1</sup></b>	Registers a face primitive to a topology and get it's identifier.
<b>CreateTopoGeom</b>	Creates a new topo geometry object from topo element array - tg_type: 1:[multi]point, 2:[multi]line, 3:[multi]poly, 4:collection

<b>toTopoGeom<sup>1</sup></b>	Creates a new topo geometry from a simple geometry
<b>TopoElementArray_Agg<sup>1</sup></b>	Returns a topoelementarray for a set of element_id, type arrays (topoelements)
<b>GetTopoGeomElementArray</b>	Returns a topoelementarray (an array of topoelements) containing the topological elements and type of the given TopoGeometry (primitive elements)
<b>GetTopoGeomElements</b>	Returns a set of topoelement objects containing the topological element_id,element_type of the given TopoGeometry (primitive elements)
<b>AsGML<sup>1</sup></b>	Returns the GML representation of a topogeometry.
<b>Drop_Indexes_Generate_Script<sup>1</sup></b>	Generates a script that drops all non-primary key and non-unique indexes on tiger schema and user specified schema. Defaults schema to tiger_data if no schema is specified.

<b>Drop_State_Tables_Generate_Script<sup>1</sup></b>	Generates a script that drops all tables in the specified schema that are prefixed with the state abbreviation. Defaults schema to tiger_data if no schema is specified.
<b>Geocode<sup>2</sup></b>	Takes in an address as a string (or other normalized address) and outputs a set of possible locations which include a point geometry in NAD 83 long lat, a normalized address for each, and the rating. The lower the rating the more likely the match. Results are sorted by lowest rating first. Can optionally pass in maximum results, defaults to 10, and restrict_region (defaults to NULL)
<b>Geocode_Intersection<sup>1</sup></b>	Takes in 2 streets that intersect and a state, city, zip, and outputs a set of possible locations on the first cross street that is at the intersection, also includes a point geometry in NAD 83 long lat, a normalized address for each location, and the rating. The lower the rating the more likely the match. Results are sorted by lowest rating first. Can optionally pass in maximum results, defaults to 10
<b>Install_Missing_Indexes<sup>1</sup></b>	Finds all tables with key columns used in geocoder joins and filter conditions that are missing used indexes on those columns and will add them.
<b>Loader_Generate_Script<sup>1</sup></b>	Generates a shell script for the specified platform for the specified states that will download Tiger data, stage and load into tiger_data schema. Each state script is returned as a separate record. Latest version supports Tiger 2010 structural changes.
<b>Missing_Indexes_Generate_Script<sup>1</sup></b>	Finds all tables with key columns used in geocoder joins that are missing indexes on those columns and will output the SQL DDL to define the index for those tables.

<b>Normalize_Address</b>	Given a textual street address, returns a composite norm_addy type that has road suffix, prefix and type standardized, street, streetname etc. broken into separate fields. This function will work with just the lookup data packaged with the tiger_geocoder (no need for tiger census data).
<b>Pprint_Addy</b>	Given a norm_addy composite type object, returns a pretty print representation of it. Usually used in conjunction with normalize_address.
<b>Reverse_Geocode<sup>1</sup></b>	Takes a geometry point in a known spatial ref sys and returns a record containing an array of theoretically possible addresses and an array of cross streets. If include_strnum_range = true, includes the street range in the cross streets.
<b>Topology_Load_Tiger<sup>1</sup></b>	Loads a defined region of tiger data into a PostGIS Topology and transforming the tiger data to spatial reference of the topology and snapping to the precision tolerance of the topology.