Post GIS Day 2012 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 2.1+*
- **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: ^{1} - Is super and beats anything else except another ^{1} or joker card. In the event of multiple ^{1}, the one that happens alphabetically first trumps the others. Symbols always trump letters.
  ^{2} - Second favorite, alphabetical rules apply (is beaten by a joker, ^{1})
  ^{mm} - 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 pronounciation. 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 possiblities are only limited by your imagination.

WHAT IS POST GIS?	**POSTGIS** 
WHAT DOES POST GIS?	**POSTGIS** 
box2d	A box composed of x min, ymin, xmax, ymax. Often used to return the 2d enclosing box of a geometry.
box3d	A box composed of x min, ymin, zmin, xmax, ymax, zmax. Often used to return the 3d extent of a geometry or collection of geometries.
geometry	Planar spatial data type.
geometry_dump	A spatial datatype with two fields - geom (holding a geometry object) and path[] (a 1-d array holding the position of the geometry within the dumped object.)

geography	Ellipsoidal spatial data type.
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 constraint based behavior
DropGeometryColumn[3d]	Removes a geometry column from a spatial table.
DropGeometryTable	Drops a table and all its references in geometry_columns.
PostGIS_Full_Version	Reports full postgis version and build configuration infos.
PostGIS_GEOS_Version	Returns the version number of the GEOS library.

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

PostGIS_Scripts_Released	Returns the version number of the postgis.sql script released with the installed postgis lib.
PostGIS_Version	Returns PostGIS version number and compile-time options.
Populate_Geometry_Columns	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
UpdateGeometrySRID[3d]	Updates the SRID of all features in a geometry column, geometry_columns metadata and srid. If it was enforced with constraints, the constraints will be updated with new srid constraint. If the old was enforced by type definition, the type definition will be changed.
ST_BdPolyFromText	Construct a Polygon given an arbitrary collection of closed linestrings as a MultiLineString Well-Known text representation.
ST_BdMPolyFromText	Construct a MultiPolygon given an arbitrary collection of closed linestrings as a MultiLineString text representation Well-Known text representation.

ST_GeogFromText[G]	Return a specified geography value from Well-Known Text representation or extended (WKT).
ST_GeographyFromText[G]	Return a specified geography value from Well-Known Text representation or extended (WKT).
ST_GeogFromWKB[G]	Creates a geography instance from a Well-Known Binary geometry representation (WKB) or extended Well Known Binary (EWKB).
ST_GeomCollFromText[mm]	Makes a collection Geometry from collection WKT with the given SRID. If SRID is not give, it defaults to -1.
ST_GeomFromEWKB[3d]	Return a specified ST_Geometry value from Extended Well-Known Binary representation (EWKB).
ST_GeomFromEWKT[3d]	Return a specified ST_Geometry value from Extended Well-Known Text representation (EWKT).

ST_GeometryFromText[mm]	Return a specified ST_Geometry value from Well-Known Text representation (WKT). This is an alias name for ST_GeomFromText
ST_GeomFromGML[3d]	Takes as input GML representation of geometry and outputs a PostGIS geometry object
ST_GeomFromGeoJSON[3d]	Takes as input a geojson representation of a geometry and outputs a PostGIS geometry object
ST_GeomFromKML[3d]	Takes as input KML representation of geometry and outputs a PostGIS geometry object
ST_GMLToSQL[mm]	Return a specified ST_Geometry value from GML representation. This is an alias name for ST_GeomFromGML
ST_GeomFromText[mm]	Return a specified ST_Geometry value from Well-Known Text representation (WKT).

ST_GeomFromWKB[mm]	Creates a geometry instance from a Well-Known Binary geometry representation (WKB) and optional SRID.
ST_LineFromMultiPoint[3d]	Creates a LineString from a MultiPoint geometry.
ST_LineFromText[mm]	Makes a Geometry from WKT representation with the given SRID. If SRID is not given, it defaults to -1.
ST_LineFromWKB[mm]	Makes a LINESTRING from WKB with the given SRID
ST_LinestringFromWKB[mm]	Makes a geometry from WKB with the given SRID.
ST_MakeBox2D	Creates a BOX2D defined by the given point geometries.

ST_3DMakeBox^{3d}	Creates a BOX3D defined by the given 3d point geometries.
ST_MakeLine^{3d}	Creates a Linestring from point or line geometries.
ST_MakeEnvelope	Creates a rectangular Polygon formed from the given minimums and maximums. Input values must be in SRS specified by the SRID.
ST_MakePolygon^{3d}	Creates a Polygon formed by the given shell. Input geometries must be closed LINESTRINGS.
ST_MakePoint^{3d}	Creates a 2D,3DZ or 4D point geometry.
ST_MakePointM	Creates a point geometry with an x y and m coordinate.

ST_MLineFromText[mm]	Return a specified ST_MultiLineString value from WKT representation.
ST_MPointFromText[mm]	Makes a Geometry from WKT with the given SRID. If SRID is not give, it defaults to -1.
ST_MPolyFromText[mm]	Makes a MultiPolygon Geometry from WKT with the given SRID. If SRID is not give, it defaults to -1.
ST_Point[mm]	Returns an ST_Point with the given coordinate values. OGC alias for ST_MakePoint.
ST_PointFromText[mm]	Makes a point Geometry from WKT with the given SRID. If SRID is not given, it defaults to unknown.
ST_PointFromWKB[mm 3d]	Makes a geometry from WKB with the given SRID

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

ST_CoordDim[mm 3d]	Return the coordinate dimension of the ST_Geometry value.
ST_Dimension[mm]	The inherent dimension of this Geometry object, which must be less than or equal to the coordinate dimension.
ST_EndPoint[mm 3d]	Returns the last point of a LINESTRING geometry as a POINT.
ST_Envelope[mm]	Returns a geometry representing the double precision (float8) bounding box of the supplied geometry.
ST_ExteriorRing[mm 3d]	Returns a line string representing the exterior ring of the POLYGON geometry. Return NULL if the geometry is not a polygon. Will not work with MULTIPOLYGON
ST_GeometryN[mm 3d]	Return the 1-based Nth geometry if the geometry is a GEOMETRYCOLLECTION, (MULTI)POINT, (MULTI)LINESTRING, MULTICURVE or (MULTI)POLYGON, POLYHEDRALSURFACE Otherwise, return NULL.

ST_GeometryType[mm 3d]	Return the geometry type of the ST_Geometry value.
ST_InteriorRingN[mm 3d]	Return the Nth interior linestring ring of the polygon geometry. Return NULL if the geometry is not a polygon or the given N is out of range.
ST_IsClosed[mm 3d]	Returns TRUE if the LINESTRING's start and end points are coincident. For Polyhedral surface is closed (volumetric).
ST_IsCollection[3d]	Returns TRUE if the argument is a collection (MULTI*, GEOMETRYCOLLECTION, ...)
ST_IsEmpty[mm]	Returns true if this Geometry is an empty geometrycollection, polygon, point etc.
ST_IsRing[mm]	Returns TRUE if this LINESTRING is both closed and simple.

ST_IsSimple[mm 3d]	Returns (TRUE) if this Geometry has no anomalous geometric points, such as self intersection or self tangency.
ST_IsValid[mm]	Returns true if the ST_Geometry is well formed.
ST_IsValidReason	Returns text stating if a geometry is valid or not and if not valid, a reason why.
ST_IsValidDetail	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.
ST_M[mm 3d]	Return the M coordinate of the point, or NULL if not available. Input must be a point.
ST_NDims[3d]	Returns coordinate dimension of the geometry as a small int. Values are: 2,3 or 4.

ST_NPoints[3d]	Return the number of points (vertexes) in a geometry.
ST_NRings[3d]	If the geometry is a polygon or multi-polygon returns the number of rings.
ST_NumGeometries[mm 3d]	If geometry is a GEOMETRYCOLLECTION (or MULTI*) return the number of geometries, for single geometries will return 1, otherwise return NULL.
ST_NumInteriorRings[mm]	Return the number of interior rings of the first polygon in the geometry. This will work with both POLYGON and MULTIPOLYGON types but only looks at the first polygon. Return NULL if there is no polygon in the geometry.
ST_NumInteriorRing[mm]	Return the number of interior rings of the first polygon in the geometry. Synonym to ST_NumInteriorRings.
ST_NumPatches[mm 3d]	Return the number of faces on a Polyhedral Surface. Will return null for non-polyhedral geometries.

ST_NumPoints[mm]	Return the number of points in an ST_LineString or ST_CircularString value.
ST_PatchN[mm 3d]	Return the 1-based Nth geometry (face) if the geometry is a POLYHEDRALSURFACE, POLYHEDRALSURFACEM. Otherwise, return NULL.
ST_PointN[mm 3d]	Return the Nth point in the first linestring or circular linestring in the geometry. Return NULL if there is no linestring in the geometry.
ST_SRID[mm]	Returns the spatial reference identifier for the ST_Geometry as defined in spatial_ref_sys table.
ST_StartPoint[mm 3d]	Returns the first point of a LINESTRING geometry as a POINT.
ST_Summary[G]	Returns a text summary of the contents of the geometry.

ST_X[mm 3d]	Return the X coordinate of the point, or NULL if not available. Input must be a point.
ST_XMax[3d]	Returns X maxima of a bounding box 2d or 3d or a geometry.
ST_XMin[3d]	Returns X minima of a bounding box 2d or 3d or a geometry.
ST_Y[mm 3d]	Return the Y coordinate of the point, or NULL if not available. Input must be a point.
ST_YMax[3d]	Returns Y maxima of a bounding box 2d or 3d or a geometry.
ST_YMin[3d]	Returns Y minima of a bounding box 2d or 3d or a geometry.

ST_Z[mm 3d]	Return the Z coordinate of the point, or NULL if not available. Input must be a point.
ST_ZMax[3d]	Returns Z minima of a bounding box 2d or 3d or a geometry.
ST_Zmflag[3d]	Returns ZM (dimension semantic) flag of the geometries as a small int. Values are: 0=2d, 1=3dm, 2=3dz, 3=4d.
ST_ZMin[3d]	Returns Z minima of a bounding box 2d or 3d or a geometry.
ST_AddPoint[3d]	Adds a point to a LineString before point position (0-based index).
ST_Affine[3d]	Applies a 3d affine transformation to the geometry to do things like translate, rotate, scale in one step.

ST_Force_2D[3d]	Forces the geometries into a "2-dimensional mode" so that all output representations will only have the X and Y coordinates.
ST_Force_3D[3d]	Forces the geometries into XYZ mode. This is an alias for ST_Force_3DZ.
ST_Force_3DZ[3d]	Forces the geometries into XYZ mode. This is a synonym for ST_Force_3D.
ST_Force_3DM	Forces the geometries into XYM mode.
ST_Force_4D[3d]	Forces the geometries into XYZM mode.
ST_Force_Collection[3d]	Converts the geometry into a GEOMETRYCOLLECTION.

ST_ForceRHR^{3d}	Forces the orientation of the vertices in a polygon to follow the Right-Hand-Rule.
ST_LineMerge	Returns a (set of) LineString(s) formed by sewing together a MULTILINESTRING.
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_Multi	Returns the geometry as a MULTI* geometry. If the geometry is already a MULTI*, it is returned unchanged.
ST_RemovePoint^{3d}	Removes point from a linestring. Offset is 0-based.

ST_Reverse	Returns the geometry with vertex order reversed.
ST_Rotate^{3d}	Rotate a geometry rotRadians counter-clockwise about an origin.
ST_RotateX^{3d}	Rotate a geometry rotRadians about the X axis.
ST_RotateY^{3d}	Rotate a geometry rotRadians about the Y axis.
ST_RotateZ^{3d}	Rotate a geometry rotRadians about the Z axis.
ST_Scale^{3d}	Scales the geometry to a new size by multiplying the ordinates with the parameters. Ie: ST_Scale(geom, Xfactor, Yfactor, Zfactor).

ST_Segmentize[2 G]	Return a modified geometry/geography having no segment longer than the given distance. Distance computation is performed in 2d only. For geometry, length units are in units of spatial reference. For geography, units are in meters.
ST_SetPoint[3d]	Replace point N of linestring with given point. Index is 0-based.
ST_SetSRID	Sets the SRID on a geometry to a particular integer value.
ST_SnapToGrid[3d]	Snap all points of the input geometry to a regular grid.
ST_Snap	Snap segments and vertices of input geometry to vertices of a reference geometry.
ST_Transform[mm]	Returns a new geometry with its coordinates transformed to the SRID referenced by the integer parameter.

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

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

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

&>	Returns TRUE if A' bounding box overlaps or is to the right of B's.
<<	Returns TRUE if A's bounding box is strictly to the left of B's.
<<|	Returns TRUE if A's bounding box is strictly below B's.
=^{G}	Returns TRUE if A's bounding box is the same as B's. Uses double precision bounding box.
>>	Returns TRUE if A's bounding box is strictly to the right of B's.
@	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.
<->	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.
<#>	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.

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

ST_3DMaxDistance[3d]	For geometry type Returns the 3-dimensional cartesian maximum distance (based on spatial ref) between two geometries in projected units.
ST_3DShortestLine[3d]	Returns the 3-dimensional shortest line between two geometries
ST_Area[mm G]	Returns the area of the surface if it is a polygon or multi-polygon. For "geometry" type area is in SRID units. For "geography" area is in square meters.
ST_Azimuth[G]	Returns the angle in radians from the horizontal of the vector defined by pointA and pointB. Angle is computed clockwise from down-to-up: on the clock: 12=0; 3=PI/2; 6=PI; 9=3PI/2.
ST_Centroid[mm]	Returns the geometric center of a geometry.
ST_ClosestPoint	Returns the 2-dimensional point on g1 that is closest to g2. This is the first point of the shortest line.

ST_Contains[mm]	Returns true if and only if no points of B lie in the exterior of A, and at least one point of the interior of B lies in the interior of A.
ST_ContainsProperly	Returns true if B intersects the interior of A but not the boundary (or exterior). A does not contain properly itself, but does contain itself.
ST_Covers[G]	Returns 1 (TRUE) if no point in Geometry B is outside Geometry A
ST_CoveredBy[G]	Returns 1 (TRUE) if no point in Geometry/Geography A is outside Geometry/Geography B
ST_Crosses[mm]	Returns TRUE if the supplied geometries have some, but not all, interior points in common.
ST_LineCrossingDirection	Given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behavior. 0 is no crossing.

ST_Disjoint[mm]	Returns TRUE if the Geometries do not "spatially intersect" - if they do not share any space together.
ST_Distance[2 mm G]	For geometry type Returns the 2-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units. For geography type defaults to return spheroidal minimum distance between two geographies in meters.
ST_HausdorffDistance	Returns the Hausdorff distance between two geometries. Basically a measure of how similar or dissimilar 2 geometries are. Units are in the units of the spatial reference system of the geometries.
ST_MaxDistance	Returns the 2-dimensional largest distance between two geometries in projected units.
ST_Distance_Sphere	Returns minimum distance in meters between two lon/lat geometries. Uses a spherical earth and radius of 6370986 meters. Faster than ST_Distance_Spheroid , but less accurate. PostGIS versions prior to 1.5 only implemented for points.
ST_Distance_Spheroid	Returns the minimum distance between two lon/lat geometries given a particular spheroid. PostGIS versions prior to 1.5 only support points.

ST_DFullyWithin	Returns true if all of the geometries are within the specified distance of one another
ST_DWithin[2 G]	Returns true if the geometries are within the specified distance of one another. For geometry units are in those of spatial reference and For geography units are in meters and measurement is defaulted to use_spheroid=true (measure around spheroid), for faster check, use_spheroid=false to measure along sphere.
ST_Equals[mm]	Returns true if the given geometries represent the same geometry. Directionality is ignored.
ST_HasArc[3d]	Returns true if a geometry or geometry collection contains a circular string
ST_Intersects[mm G]	Returns TRUE if the Geometries/Geography "spatially intersect in 2D" - (share any portion of space) and FALSE if they don't (they are Disjoint). For geography -- tolerance is 0.00001 meters (so any points that close are considered to intersect)
ST_Length[mm G]	Returns the 2d length of the geometry if it is a linestring or multilinestring. geometry are in units of spatial reference and geography are in meters (default spheroid)

ST_Length2D	Returns the 2-dimensional length of the geometry if it is a linestring or multi-linestring. This is an alias for ST_Length
ST_3DLength[3d]	Returns the 3-dimensional or 2-dimensional length of the geometry if it is a linestring or multi-linestring.
ST_Length_Spheroid[3d]	Calculates the 2D or 3D length of a linestring/multilinestring on an ellipsoid. This is useful if the coordinates of the geometry are in longitude/latitude and a length is desired without reprojection.
ST_Length2D_Spheroid	Calculates the 2D length of a linestring/multilinestring on an ellipsoid. This is useful if the coordinates of the geometry are in longitude/latitude and a length is desired without reprojection.
ST_3DLength_Spheroid[3d]	Calculates the length of a geometry on an ellipsoid, taking the elevation into account. This is just an alias for ST_Length_Spheroid.
ST_LongestLine	Returns the 2-dimensional longest line points of two geometries. The function will only return the first longest line if more than one, that the function finds. The line returned will always start in g1 and end in g2. The length of the line this function returns will always be the same as st_maxdistance returns for g1 and g2.

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

ST_Project[G]	Returns a POINT projected from a start point using a bearing and distance.
ST_Relate[mm]	Returns true if this Geometry is spatially related to anotherGeometry, by testing for intersections between the Interior, Boundary and Exterior of the two geometries as specified by the values in the intersectionMatrixPattern. If no intersectionMatrixPattern is passed in, then returns the maximum intersectionMatrixPattern that relates the 2 geometries.
ST_RelateMatch	Returns true if intersectionMattrixPattern1 implies intersectionMatrixPattern2
ST_ShortestLine	Returns the 2-dimensional shortest line between two geometries
ST_Touches[mm]	Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.
ST_Within[mm]	Returns true if the geometry A is completely inside geometry B

ST_Buffer[mm G]	(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. buffer_style options: quad_segs=#,endcap=round|flat|square,join=round|mitre|bevel,mitre_lin
ST_BuildArea	Creates an areal geometry formed by the constituent linework of given geometry
ST_Collect[3d]	Return a specified ST_Geometry value from a collection of other geometries.
ST_ConcaveHull	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.
ST_ConvexHull[mm 3d]	The convex hull of a geometry represents the minimum convex geometry that encloses all geometries within the set.
ST_CurveToLine[mm 3d]	Converts a CIRCULARSTRING/CURVEDPOLYGON to a LINESTRING/POLYGON

ST_DelaunayTriangles[1 g3.4 3d]	Return a Delaunay triangulation around the given input points.
ST_Difference[mm 3d]	Returns a geometry that represents that part of geometry A that does not intersect with geometry B.
ST_Dump[3d]	Returns a set of geometry_dump (geom,path) rows, that make up a geometry g1.
ST_DumpPoints[2 3d]	Returns a set of geometry_dump (geom,path) rows of all points that make up a geometry.
ST_DumpRings[3d]	Returns a set of geometry_dump rows, representing the exterior and interior rings of a polygon.
ST_FlipCoordinates[3d]	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.

ST_Intersection[mm G]	(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.
ST_LineToCurve[3d]	Converts a LINESTRING/POLYGON to a CIRCULARSTRING, CURVED POLYGON
ST_MakeValid[2 3d]	Attempts to make an invalid geometry valid w/out loosing vertices.
ST_MemUnion[3d]	Same as ST_Union, only memory-friendly (uses less memory and more processor time).
ST_MinimumBoundingCircle	Returns the smallest circle polygon that can fully contain a geometry. Default uses 48 segments per quarter circle.
ST_Polygonize	Aggregate. Creates a GeometryCollection containing possible polygons formed from the constituent linework of a set of geometries.

ST_Node[3d]	Node a set of linestrings.
ST_OffsetCurve	Return an offset line at a given distance and side from an input line. Useful for computing parallel lines about a center line
ST_RemoveRepeatedPoints[3d]	Returns a version of the given geometry with duplicated points removed.
ST_SharedPaths	Returns a collection containing paths shared by the two input linestrings/multilinestrings.
ST_Shift_Longitude[3d]	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
ST_Simplify	Returns a "simplified" version of the given geometry using the Douglas-Peucker algorithm.

ST_SimplifyPreserveTopology	Returns a "simplified" version of the given geometry using the Douglas-Peucker algorithm. Will avoid creating derived geometries (polygons in particular) that are invalid.
ST_Split	Returns a collection of geometries resulting by splitting a geometry.
ST_SymDifference[mm 3d]	Returns a geometry that represents the portions of A and B that do not intersect. It is called a symmetric difference because ST_SymDifference(A,B) = ST_SymDifference(B,A).
ST_Union[mm]	Returns a geometry that represents the point set union of the Geometries.
ST_UnaryUnion[3d]	Like ST_Union, but working at the geometry component level.
ST_Line_Interpolate_Point[3d]	Returns a point interpolated along a line. Second argument is a float8 between 0 and 1 representing fraction of total length of linestring the point has to be located.

ST_Line_Locate_Point	Returns a float between 0 and 1 representing the location of the closest point on LineString to the given Point, as a fraction of total 2d line length.
ST_Line_Substring^{3d}	Return a linestring being a substring of the input one starting and ending at the given fractions of total 2d length. Second and third arguments are float8 values between 0 and 1.
ST_LocateAlong	Return a derived geometry collection value with elements that match the specified measure. Polygonal elements are not supported.
ST_LocateBetween	Return a derived geometry collection value with elements that match the specified range of measures inclusively. Polygonal elements are not supported.
ST_LocateBetweenElevations^{3d}	Return a derived geometry (collection) value with elements that intersect the specified range of elevations inclusively. Only 3D, 4D LINESTRINGS and MULTILINESTRINGS are supported.
ST_InterpolatePoint^{3d}	Return the value of the measure dimension of a geometry at the point closed to the provided point.

ST_AddMeasure^{3d}	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.
AddAuth	Add an authorization token to be used in current transaction.
CheckAuth	Creates trigger on a table to prevent/allow updates and deletes of rows based on authorization token.
DisableLongTransactions	Disable long transaction support. This function removes the long transaction support metadata tables, and drops all triggers attached to lock-checked tables.
EnableLongTransactions	Enable long transaction support. This function creates the required metadata tables, needs to be called once before using the other functions in this section. Calling it twice is harmless.
LockRow	Set lock/authorization for specific row in table

UnlockRows	Remove all locks held by specified authorization id. Returns the number of locks released.
ST_Accum^{3d}	Aggregate. Constructs an array of geometries.
Box2D	Returns a BOX2D representing the maximum extents of the geometry.
Box3D^{3d}	Returns a BOX3D representing the maximum extents of the geometry.
ST_EstimatedExtent	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.
ST_Expand	Returns bounding box expanded in all directions from the bounding box of the input geometry. Uses double-precision

ST_Extent	an aggregate function that returns the bounding box that bounds rows of geometries.
ST_3DExtent^{3d}	an aggregate function that returns the box3D bounding box that bounds rows of geometries.
Find_SRID	The syntax is find_srid(db/schema, table, column) and the function returns the integer SRID of the specified column by searching through the GEOMETRY_COLUMNS table.
ST_Mem_Size^{3d}	Returns the amount of space (in bytes) the geometry takes.
ST_Point_Inside_Circle	Is the point geometry insert circle defined by center_x, center_y, radius
PostGIS_AddBBox	Add bounding box to the geometry.

PostGIS_DropBBox	Drop the bounding box cache from the geometry.
PostGIS_HasBBox	Returns TRUE if the bbox of this geometry is cached, FALSE otherwise.
geomval	A spatial datatype with two fields - geom (holding a geometry object) and val (holding a double precision pixel value from a raster band).
addbandarg	A composite type used as input into the ST_AddBand function defining the attributes and initial value of the new band.
rastbandarg	A composite type for use when needing to express a raster and a band index of that raster.
raster	raster spatial data type.

reclassarg	A composite type used as input into the ST_Reclass function defining the behavior of reclassification.
unionarg	A composite type used as input into the ST_Union function defining the bands to be processed and behavior of the UNION operation.
AddRasterConstraints	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 of the constraint setting was accomplished and if issues a notice.
DropRasterConstraints	Drops PostGIS raster constraints that refer to a raster table column. Useful if you need to reload data or update your raster column data.
PostGIS_Raster_Lib_Build_Date	Reports full raster library build date.
PostGIS_Raster_Lib_Version	Reports full raster version and build configuration infos.

ST_GDALDrivers	Returns a list of raster formats supported by your lib gdal. These are the formats you can output your raster using ST_AsGDALRaster.
UpdateRasterSRID[1]	Change the SRID of all rasters in the user-specified column and table.
ST_AddBand[2]	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.
ST_AsRaster	Converts a PostGIS geometry to a PostGIS raster.
ST_Band	Returns one or more bands of an existing raster as a new raster. Useful for building new rasters from existing rasters.
ST_MakeEmptyRaster	Returns an empty raster (having no bands) of given dimensions (width & height), upperleft X and Y, pixel size and rotation (scalex, scaley, skewx & skewy) and reference system (srid). If a raster is passed in, returns a new raster with the same size, alignment and SRID. If srid is left out, the spatial ref is set to unknown (0).

ST_Tile[1]	Returns a set of rasters resulting from the split of the input raster based upon the desired dimensions of the output rasters.
ST_GeoReference	Returns the georeference meta data in GDAL or ESRI format as commonly seen in a world file. Default is GDAL.
ST_Height	Returns the height of the raster in pixels.
ST_MetaData	Returns basic meta data about a raster object such as pixel size, rotation (skew), upper, lower left, etc.
ST_NumBands	Returns the number of bands in the raster object.
ST_PixelHeight	Returns the pixel height in geometric units of the spatial reference system.

ST_PixelWidth	Returns the pixel width in geometric units of the spatial reference system.
ST_ScaleX	Returns the X component of the pixel width in units of coordinate reference system.
ST_ScaleY	Returns the Y component of the pixel height in units of coordinate reference system.
ST_Raster2WorldCoord[1]	Returns the raster's upper left corner as geometric X and Y (longitude and latitude) given a column and row. Column and row starts at 1.
ST_Raster2WorldCoordX	Returns the geometric X coordinate upper left of a raster, column and row. Numbering of columns and rows starts at 1.
ST_Raster2WorldCoordY	Returns the geometric Y coordinate upper left corner of a raster, column and row. Numbering of columns and rows starts at 1.

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

ST_Width	Returns the width of the raster in pixels.
ST_World2RasterCoord[1]	Returns the upper left corner as column and row given geometric X and Y (longitude and latitude) or a point geometry expressed in the spatial reference coordinate system of the raster.
ST_World2RasterCoordX	Returns the column in the raster of the point geometry (pt) or a X and Y world coordinate (xw, yw) represented in world spatial reference system of raster.
ST_World2RasterCoordY	Returns the row in the raster of the point geometry (pt) or a X and Y world coordinate (xw, yw) represented in world spatial reference system of raster.
ST_IsEmpty	Returns true if the raster is empty (width = 0 and height = 0). Otherwise, returns false.
ST_BandMetaData	Returns basic meta data for a specific raster band. band num 1 is assumed if none-specified.

ST_BandNoDataValue	Returns the value in a given band that represents no data. If no band num 1 is assumed.
ST_BandIsNoData	Returns true if the band is filled with only nodata values.
ST_BandPath	Returns system file path to a band stored in file system. If no bandnum specified, 1 is assumed.
ST_BandPixelType	Returns the type of pixel for given band. If no bandnum specified, 1 is assumed.
ST_HasNoBand	Returns true if there is no band with given band number. If no band number is specified, then band number 1 is assumed.
ST_PixelAsPolygon	Returns the polygon geometry that bounds the pixel for a particular row and column.

ST_PixelAsPolygons[2]	Returns the polygon geometry that bounds every pixel of a raster band along with the value, the X and the Y raster coordinates of each pixel.
ST_PixelAsPoint[1]	Returns a point geometry of the pixel's upper-left corner.
ST_PixelAsPoints[1]	Returns a point geometry for each pixel of a raster band along with the value, the X and the Y raster coordinates of each pixel. The coordinates of the point geometry are of the pixel's upper-left corner.
ST_PixelAsCentroid[1]	Returns the centroid (point geometry) of the area represented by a pixel.
ST_PixelAsCentroids[1]	Returns the centroid (point geometry) for each pixel of a raster band along with the value, the X and the Y raster coordinates of each pixel. The point geometry is the centroid of the area represented by a pixel.
ST_Value	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.

ST_NearestValue[1]	Returns the nearest non-NODATA value of a given band's pixel specified by a columnx and rowy or a geometric point expressed in the same spatial reference coordinate system as the raster.
ST_Neighborhood[1]	Returns a 2-D double precision array of the non-NODATA values around a given band's pixel specified by either a columnX and rowY or a geometric point expressed in the same spatial reference coordinate system as the raster.
ST_SetValue[2]	Returns modified raster resulting from setting the value of a given band in a given columnx, rowy pixel or the pixels that intersect a particular geometry. Band numbers start at 1 and assumed to be 1 if not specified.
ST_SetValues[1]	Returns modified raster resulting from setting the values of a given band.
ST_DumpValues[1]	Get the values of the specified band as a 2-dimension array.
ST_PixelOfValue[1]	Get the columnx, rowy coordinates of the pixel whose value equals the search value.

ST_SetGeoReference	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.
ST_SetRotation	Set the rotation of the raster in radian.
ST_SetScale	Sets the X and Y size of pixels in units of coordinate reference system. Number units/pixel width/height.
ST_SetSkew	Sets the georeference X and Y skew (or rotation parameter). If only one is passed in, sets X and Y to the same value.
ST_SetSRID	Sets the SRID of a raster to a particular integer srid defined in the spatial_ref_sys table.
ST_SetUpperLeft	Sets the value of the upper left corner of the pixel to projected X and Y coordinates.

ST_Resample	Resample a raster using a specified resampling algorithm, new dimensions, an arbitrary grid corner and a set of raster georeferencing attributes defined or borrowed from another raster. New pixel values are computed using the NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbor.
ST_Rescale	Resample a raster by adjusting only its scale (or pixel size). New pixel values are computed using the NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbor.
ST_Reskew	Resample a raster by adjusting only its skew (or rotation parameters). New pixel values are computed using the NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbor.
ST_SnapToGrid	Resample a raster by snapping it to a grid. New pixel values are computed using the NearestNeighbor (english or american spelling), Bilinear, Cubic, CubicSpline or Lanczos resampling algorithm. Default is NearestNeighbor.
ST_Transform	Reprojects a raster in a known spatial reference system to another known spatial reference system using specified resampling algorithm. Options are NearestNeighbor, Bilinear, Cubic, CubicSpline, Lanczos defaulting to NearestNeighbor.
ST_SetBandNoDataValue	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.

ST_SetBandIsNoData	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.
ST_Count	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 exclude_nodata_value is set to true, will only count pixels that are not equal to the nodata value.
ST_Histogram	Returns a set of record summarizing a raster or raster coverage data distribution separate bin ranges. Number of bins are autocomputed if not specified.
ST_Quantile	Compute quantiles for a raster or raster table coverage in the context of the sample or population. Thus, a value could be examined to be at the raster's 25%, 50%, 75% percentile.
ST_SummaryStats	Returns record 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.
ST_ValueCount	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.

ST_AsBinary	Return the Well-Known Binary (WKB) representation of the raster without SRID meta data.
ST_AsGDALRaster	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.
ST_AsJPEG	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.
ST_AsPNG	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.
ST_AsTIFF	Return the raster selected bands as a single TIFF image (byte array). If no band is specified, then will try to use all bands.
Box3D	Returns the box 3d representation of the enclosing box of the raster.

ST_Clip[2]	Returns the raster clipped by the input geometry. If band number not is specified, all bands are processed. If crop is not specified or TRUE, the output raster is cropped.
ST_ConvexHull	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 result as ST_Envelope so only useful for irregularly shaped or skewed rasters.
ST_DumpAsPolygons	Returns a set of geomval (geom,val) rows, from a given raster band. If no band number is specified, band num defaults to 1.
ST_Envelope	Returns the polygon representation of the extent of the raster.
ST_HillShade[2]	Returns the hypothetical illumination of an elevation raster band using provided azimuth, altitude, brightness and scale inputs.
ST_Aspect[2]	Returns the aspect (in degrees by default) of an elevation raster band. Useful for analyzing terrain.

ST_Slope[2]	Returns the slope (in degrees by default) of an elevation raster band. Useful for analyzing terrain.
ST_Intersection	Returns a raster or a set of geometry-pixelvalue pairs representing the shared portion of two rasters or the geometrical intersection of a vectorization of the raster and a geometry.
ST_MapAlgebra[1]	Callback function version - Returns a one-band raster given one or more input rasters, band indexes and one user-specified callback function.
ST_MapAlgebra[1]	Expression version - Returns a one-band raster given one or two input rasters, band indexes and one or more user-specified SQL expressions.
ST_MapAlgebraExpr	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 pixeltype provided. Band 1 is assumed if no band is specified.
ST_MapAlgebraExpr	2 raster band version: Creates a new one band raster formed by applying a valid PostgreSQL algebraic operation on the two 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 and have its extent defined by the "extenttype" parameter. Values for "extenttype" can be: INTERSECTION, UNION, FIRST, SECOND.

ST_MapAlgebraFct	1 band version - Creates a new one band raster formed by applying a valid PostgreSQL function on the input raster band and of pixeltype prodived. Band 1 is assumed if no band is specified.
ST_MapAlgebraFct	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 prodived. Band 1 is assumed if no band is specified. Extent type defaults to INTERSECTION if not specified.
ST_MapAlgebraFctNgb	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.
ST_Polygon^{2}	Returns a multipolygon 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.
ST_Reclass	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 8BUI and so forth for simpler rendering as viewable formats.
ST_Union^{2}	Returns the union of a set of raster tiles into a single raster composed of 1 or more bands.

ST_Min4ma[2]	Raster processing function that calculates the minimum pixel value in a neighborhood.
ST_Max4ma[2]	Raster processing function that calculates the maximum pixel value in a neighborhood.
ST_Sum4ma[2]	Raster processing function that calculates the sum of all pixel values in a neighborhood.
ST_Mean4ma[2]	Raster processing function that calculates the mean pixel value in a neighborhood.
ST_Range4ma[2]	Raster processing function that calculates the range of pixel values in a neighborhood.
ST_Distinct4ma[2]	Raster processing function that calculates the number of unique pixel values in a neighborhood.

ST_StdDev4ma[2]	Raster processing function that calculates the standard deviation of pixel values in a neighborhood.
ST_InvDistWeight4ma[1]	Raster processing function that interpolates a pixel's value from the pixel's neighborhood.
ST_MinDist4ma[1]	Raster processing function that returns the minimum distance (in number of pixels) between the pixel of interest and a neighboring pixel with value.
&&	Returns TRUE if A's bounding box intersects B's bounding box.
&<	Returns TRUE if A's bounding box is to the left of B's.
&>	Returns TRUE if A's bounding box is to the right of B's.

ST_Contains[1]	Return true if no points of raster rastB lie in the exterior of raster rastA and at least one point of the interior of rastB lies in the interior of rastA.
ST_ContainsProperly[1]	Return true if rastB intersects the interior of rastA but not the boundary or exterior of rastA.
ST_Covers[1]	Return true if no points of raster rastB lie outside raster rastA.
ST_CoveredBy[1]	Return true if no points of raster rastA lie outside raster rastB.
ST_Disjoint[1]	Return true if raster rastA does not spatially intersect rastB.
ST_Intersects	Return true if raster rastA spatially intersects raster rastB.

ST_Overlaps[1]	Return true if raster rastA and rastB intersect but one does not completely contain the other.
ST_Touches[1]	Return true if raster rastA and rastB have at least one point in common but their interiors do not intersect.
ST_SameAlignment[2]	Returns true if rasters have same skew, scale, spatial ref and false if they don't with notice detailing issue.
ST_Within[1]	Return true if no points of raster rastA lie in the exterior of raster rastB and at least one point of the interior of rastA lies in the interior of rastB.
ST_DWithin[1]	Return true if rasters rastA and rastB are within the specified distance of each other.
ST_DFullyWithin[1]	Return true if rasters rastA and rastB are fully within the specified distance of each other.

getfaceedges_returntype	A composite type that consists of a sequence number and edge number. This is the return type for ST_GetFaceEdges
TopoGeometry	A composite type representing a topologically defined geometry
validatetopology_returntype	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
TopoElement	An array of 2 integers generally used to identify a TopoGeometry component.
TopoElementArray	An array of TopoElement objects
AddTopoGeometryColumn	Adds a topogeometry column to an existing table, registers this new column as a layer in topology.layer and returns the new layer_id.

DropTopology	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.
DropTopoGeometryColumn	Drops the topogeometry column from the table named table_name in schema schema_name and unregisters the columns from topology.layer table.
TopologySummary	Takes a topology name and provides summary totals of types of objects in topology
ValidateTopology	Returns a set of validatetopology_returntype objects detailing issues with topology
CreateTopology	Creates a new topology schema and registers this new schema in the topology.topology table.
CopyTopology	Makes a copy of a topology structure (nodes, edges, faces, layers and TopoGeometries).

ST_InitTopoGeo[mm]	Creates a new topology schema and registers this new schema in the topology.topology table and details summary of process.
ST_CreateTopoGeo[mm]	Adds a collection of geometries to a given empty topology and returns a message detailing success.
TopoGeo_AddPoint	Adds a point to an existing topology using a tolerance and possibly splitting an existing edge.
TopoGeo_AddLineString	Adds a linestring to an existing topology using a tolerance and possibly splitting existing edges/faces.
TopoGeo_AddPolygon	Adds a polygon to an existing topology using a tolerance and possibly splitting existing edges/faces.
ST_AddIsoNode[mm]	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.

ST_AddIsoEdge[mm]	Adds an isolated edge defined by geometry alinestring to a topology connecting two existing isolated nodes anode and anothernode and returns the edge id of the new edge.
ST_AddEdgeNewFaces[mm]	Add a new edge and, if in doing so it splits a face, delete the original face and replace it with two new faces.
ST_AddEdgeModFace[mm]	Add a new edge and, if in doing so it splits a face, modify the original face and add a new face.
ST_RemEdgeNewFace[mm]	Removes an edge and, if the removed edge separated two faces, delete the original faces and replace them with a new face.
ST_RemEdgeModFace[mm]	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.
ST_ChangeEdgeGeom[mm]	Changes the shape of an edge without affecting the topology structure.

ST_ModEdgeSplit[mm]	Split an edge by creating a new node along an existing edge, modifying the original edge and adding a new edge.
ST_ModEdgeHeal[mm]	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.
ST_NewEdgeHeal[mm]	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.
ST_MoveIsoNode[mm]	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.
ST_NewEdgesSplit[mm]	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.
ST_RemoveIsoNode[mm]	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.

GetEdgeByPoint	Find the edge-id of an edge that intersects a given point
GetFaceByPoint	Find the face-id of a face that intersects a given point
GetNodeByPoint	Find the id of a node at a point location
GetTopologyID	Returns the id of a topology in the topology.topology table given the name of the topology.
GetTopologySRID	Returns the SRID of a topology in the topology.topology table given the name of the topology.
GetTopologyName	Returns the name of a topology (schema) given the id of the topology.

ST_GetFaceEdges[mm]	Returns a set of ordered edges that bound aface includes the sequence order.
ST_GetFaceGeometry[mm]	Returns the polygon in the given topology with the specified face id.
GetRingEdges	Returns an ordered set of edges forming a ring with the given edge .
GetNodeEdges	Returns an ordered set of edges incident to the given node.
Polygonize	Find and register all faces defined by topology edges
AddNode	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.

AddEdge	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.
AddFace	Registers a face primitive to a topology and get it's identifier.
CreateTopoGeom	Creates a new topo geometry object from topo element array - tg_type: 1:[multi]point, 2:[multi]line, 3:[multi]poly, 4:collection
toTopoGeom	Creates a new topo geometry from a simple geometry
TopoElementArray_Agg	Returns a topoelementarray for a set of element_id, type arrays (topoelements)
GetTopoGeomElementArray	Returns a topoelementarray (an array of topoelements) containing the topological elements and type of the given TopoGeometry (primitive elements)

GetTopoGeomElements	Returns a set of topoelement objects containing the topological element_id,element_type of the given TopoGeometry (primitive elements)
AsGML	Returns the GML representation of a topogeometry.
Drop_Indexes_Generate_Script	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.
Drop_Nation_Tables_Generate_Script[1]	Generates a script that drops all tables in the specified schema that start with county_all, state_all or stae code followed by county or state.
Drop_State_Tables_Generate_Script	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.
Geocode	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)

Geocode_Intersection	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
Get_Geocode_Setting[1]	Returns value of specific setting stored in tiger.geocode_settings table.
Get_Tract	Returns census tract or field from tract table of where the geometry is located. Default to returning short name of tract.
Install_Missing_Indexes	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.
Loader_Generate_Census_Script	Generates a shell script for the specified platform for the specified states that will download Tiger census state tract, bg, and tabblocks data tables, stage and load into tiger_data schema. Each state script is returned as a separate record.
Loader_Generate_Script	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 and also loads census tract, block groups, and blocks tables.

Loader_Generate_Nation_Script[1]	Generates a shell script for the specified platform that loads in the county and state lookup tables.
Missing_Indexes_Generate_Script	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.
Normalize_Address	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).
Pprint_Addy	Given a norm_addy composite type object, returns a pretty print representation of it. Usually used in conjunction with normalize_address.
Reverse_Geocode	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.
Topology_Load_Tiger	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.

Set_Geocode_Setting[1]	Sets a setting that affects behavior of geocoder functions.