

PostGIS 3.3 Topology Cheatsheet

New in this release ¹ Enhanced in this release ² Requires GEOS 3.4 or higher^{3,4}

Topology Types

getfaceedges_returntype A composite type that consists of a sequence number and an edge number.

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.

Topology Domains

TopoElement An array of 2 integers generally used to identify a TopoGeometry component.

TopoElementArray An array of TopoElement objects.

Topology and TopoGeometry Management

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.

1. topology_name, schema_name, table_name, column_name, feature_type
2. topology_name, schema_name, table_name, column_name, feature_type, child_layer

DropTopology (topology_schema_name) 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 (schema_name, table_name, column_name) Drops the topogeometry column from the table named table_name in schema schema_name and unregisters the columns from topology.layer table.

Populate_Topology_Layer () Adds missing entries to topology.layer table by reading metadata from topo tables.

TopologySummary (topology_schema_name) Takes a topology name and provides summary totals of types of objects in topology.

ValidateTopology (toponame, bbox) Returns a set of validatetopology_returntype objects detailing issues with topology.

ValidateTopologyRelation (toponame) Returns info about invalid topology relation records

FindTopology Returns a topology record by different means.

1. topogeom
2. layerTable, layerColumn
3. layerSchema, layerTable, layerColumn
4. topoName
5. id

FindLayer Returns a topology.layer record by different means.

1. tg
2. layer_table, feature_column
3. schema_name, table_name, feature_column
4. topology_id, layer_id

Topology Statistics Management

Topology Constructors

CreateTopology Creates a new topology schema and registers this new schema in the topology.topology table.

1. topology_schema_name
2. topology_schema_name, srid
3. topology_schema_name, srid, prec
4. topology_schema_name, srid, prec, hasz

CopyTopology (existing_topology_name, new_name) Makes a copy of a topology structure (nodes, edges, faces, layers and TopoGeometries).

ST_InitTopoGeo^{mm} (topology_schema_name) Creates a new topology schema and registers this new schema in the topology.topology table and details summary of process.

ST_CreateTopoGeo^{mm} (atopology, acollection) Adds a collection of geometries to a given empty topology and returns a message detailing success.

TopoGeo_AddPoint (atopology, apoint, tolerance) Adds a point to an existing topology using a tolerance and possibly splitting an existing edge.

TopoGeo_AddLineString (atopology, aline, tolerance) Adds a linestring to an existing topology using a tolerance and possibly splitting existing edges/faces. Returns edge identifiers.

TopoGeo_AddPolygon (atopology, apoly, tolerance) Adds a polygon to an existing topology using a tolerance and possibly splitting existing edges/faces. Returns face identifiers.

Topology Editors

ST_AddIsoNode^{mm} (atopology, aface, apoint) 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} (atopology, anode, anothernode, alinestring) 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} (atopology, anode, anothernode, acurve) 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} (atopology, anode, anothernode, acurve) 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} (atopology, anedge) 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} (atopology, anedge) 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} (atopology, anedge, acurve) Changes the shape of an edge without affecting the topology structure.

ST_ModEdgeSplit^{mm} (atopology, anedge, apoint) Split an edge by creating a new node along an existing edge, modifying the original edge and adding a new edge.

ST_ModEdgeHeal^{mm} (atopology, anedge, anotheredge) Heals 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} (atopology, anedge, anotheredge) Heals 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} (atopology, anode, apoint) 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} (atopology, anedge, apoint) 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} (atopology, anode) 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.

ST_RemoveIsoEdge^{mm} (atopology, anedge) Removes an isolated edge and returns description of action. If the edge is not isolated, then an exception is thrown.

Topology Accessors

GetEdgeByPoint (atopology, apoint, tol1) Finds the edge-id of an edge that intersects a given point.

GetFaceByPoint (atopology, apoint, tol1) Finds face intersecting a given point.

GetFaceContainingPoint (atopology, apoint) Finds the face containing a point.

GetNodeByPoint (atopology, apoint, tol1) Finds the node-id of a node at a point location.

GetTopologyID (toponame) Returns the id of a topology in the topology.topology table given the name of the topology.

GetTopologySRID (toponame) Returns the SRID of a topology in the topology.topology table given the name of the topology.

GetTopologyName (topology_id) Returns the name of a topology (schema) given the id of the topology.

ST_GetFaceEdges^{mm} (atopology, aface) Returns a set of ordered edges that bound aface.

ST_GetFaceGeometry^{mm} (atopology, aface) Returns the polygon in the given topology with the specified face id.

GetRingEdges (atopology, aring, max_edges=null) Returns the ordered set of signed edge identifiers met by walking on an a given edge side.

GetNodeEdges (atopology, anode) Returns an ordered set of edges incident to the given node.

Topology Processing

Polygonize (toponame) Finds and registers all faces defined by topology edges.

AddNode (toponame, apoint, allowEdgeSplitting=false, computeContainingFace=false) 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 (toponame, aline) 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 (toponame, apolygon, force_new=false) Registers a face primitive to a topology and gets its identifier.

ST_Simplify (tg, tolerance) Returns a "simplified" geometry version of the given TopoGeometry using the Douglas-Peucker algorithm.

RemoveUnusedPrimitives¹ (topology_name, bbox) Removes topology primitives which not needed to define existing TopoGeometry objects.

TopoGeometry Constructors

CreateTopoGeom Creates a new topo geometry object from topo element array - tg_type: 1:[multi]point, 2:[multi]line, 3:[multi]poly, 4:collection

1. toponame, tg_type, layer_id, tg_objs
2. toponame, tg_type, layer_id

toTopoGeom Converts a simple Geometry into a topo geometry.

1. geom, toponame, layer_id, tolerance
2. geom, topogeom, tolerance

TopoElementArray_Agg (tefield) Returns a topelementarray for a set of element_id, type arrays (topelements).

TopoGeometry Editors

clearTopoGeom (topogeom) Clears the content of a topo geometry.

TopoGeom_addElement (tg, el) Adds an element to the definition of a TopoGeometry.

TopoGeom_remElement (tg, el) Removes an element from the definition of a TopoGeometry.

TopoGeom_addTopoGeom (tgt, src) Adds element of a TopoGeometry to the definition of another TopoGeometry.

toTopoGeom Adds a geometry shape to an existing topo geometry.

TopoGeometry Accessors

GetTopoGeomElementArray Returns a topelementarray (an array of topelements) containing the topological elements and type of the given TopoGeometry (primitive elements).

1. toponame, layer_id, tg_id
2. tg

GetTopoGeomElements Returns a set of topelement objects containing the topological element_id, element_type of the given TopoGeometry (primitive elements).

1. toponame, layer_id, tg_id
2. tg

ST_SRID^{mm} (tg) Returns the spatial reference identifier for a topogeometry.

TopoGeometry Outputs

AsGML Returns the GML representation of a topogeometry.

1. tg
2. tg, nsprefix_in
3. tg, visitedTable
4. tg, visitedTable, nsprefix
5. tg, nsprefix_in, precision, options
6. tg, nsprefix_in, precision, options, visitedTable
7. tg, nsprefix_in, precision, options, visitedTable, idprefix
8. tg, nsprefix_in, precision, options, visitedTable, idprefix, gmlversion

AsTopoJSON (tg, edgeMapTable) Returns the TopoJSON representation of a topogeometry.

Topology Spatial Relationships

Equals^{3D} (tg1, tg2) Returns true if two topogeometries are composed of the same topology primitives.

Intersects^{3D} (tg1, tg2) Returns true if any pair of primitives from the two topogeometries intersect.

Importing and exporting Topologies