Lab 7
Topology

Learning Objectives:

- ✓ The construction of topologies, and
- ✓ Understanding and construction of the resultant schemas, tables, and
- ✓ Apply respective constraints.

The PostGIS topology types and functions are used to manage topological objects. Examples of topological objects are: faces, edges and nodes.

Pre-Requisitve:

- ✓ Check that your PostGIS installation contains the topology plugin. Topology is part of PostGIS since version PostGIS20

The construction of topologies and the resultant schemas, tables, and constraints.

For this section we will assume pgAdmin is used. Connect to the Postgis20 database and explicitly set the search path, so we do not have to explicitly include schema names.

```
SET search_path TO public,topology;
```

Without the above line you will have to explicitly put `TOPOLOGY.` In front of every function or table that you are using.

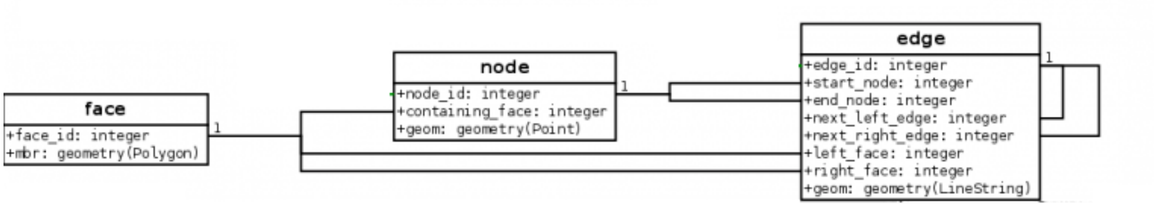
Execute the following SQL
-- If tables or topology exist they can be deleted as follows.

```
-- SELECT topology.DropTopology('toposchema1');
-- DROP TABLE public.topo1;

SELECT CreateTopology('toposchema1', 0, 1.0);
CREATE TABLE public.topo1 (id SERIAL PRIMARY KEY);
SELECT  AddTopoGeometryColumn('toposchema1',  'public',  'topo1',  'topogeom',
'LINESTRING');

INSERT  INTO  topo1  (topogeom)  VALUES  (toTopoGeom('LINESTRING(0  0  ,  100
0)'::geometry, 'toposchema1', 1, 1));

INSERT  INTO  topo1  (topogeom)  VALUES  (toTopoGeom('LINESTRING(100  0  ,  110
50)'::geometry, 'toposchema1', 1, 1));
INSERT  INTO  topo1  (topogeom)  VALUES  (toTopoGeom('LINESTRING(110  50  ,  100
100)'::geometry, 'toposchema1', 1, 1));
INSERT  INTO  topo1  (topogeom)  VALUES  (toTopoGeom('LINESTRING(100  100  ,  0
100)'::geometry, 'toposchema1', 1, 1));
```

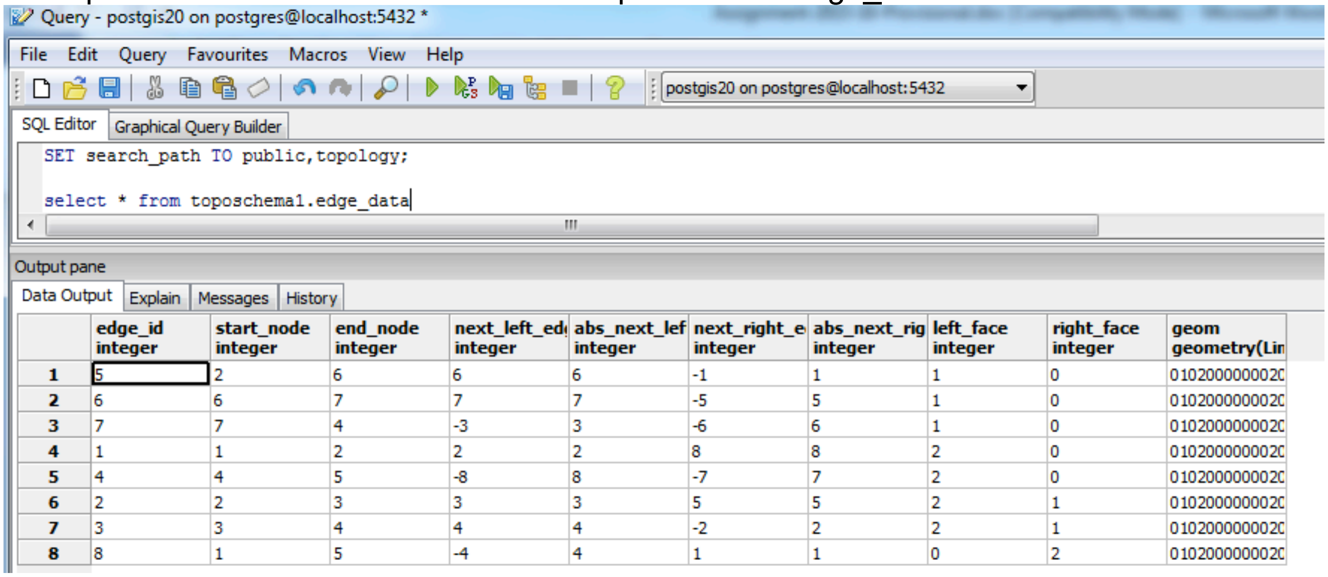
Lab 7
Topology

```
INSERT INTO topo1 (topogeom) VALUES (toTopoGeom('LINESTRING(100 0 , 200
0)'::geometry, 'toposchema1', 1, 1));
INSERT INTO topo1 (topogeom) VALUES (toTopoGeom('LINESTRING(200 0 , 200
100)'::geometry, 'toposchema1', 1, 1));
INSERT INTO topo1 (topogeom) VALUES (toTopoGeom('LINESTRING(200 100 , 100
100)'::geometry, 'toposchema1', 1, 1));
INSERT INTO topo1 (topogeom) VALUES (toTopoGeom('LINESTRING(0 0 , 0
100)'::geometry, 'toposchema1', 1, 1));
```

The main table created by `CreateTopology` are:



Example one of the created tables called toposchema1.edge_data.

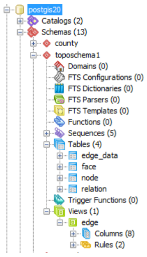

Query - postgis20 on postgres@localhost:5432 *
File Edit Query Favourites Macros View Help
postgis20 on postgres@localhost:5432
SQL Editor
Graphical Query Builder
Output pane
Data Output
Explain
Messages
History

```
SET search_path TO public,topology;

select * from toposchema1.edge_data
```

	edge_id integer	start_node integer	end_node integer	next_left_ed integer	abs_next_lef integer	next_right_e integer	abs_next_rig integer	left_face integer	right_face integer	geom geometry(Lin
1	5	2	6	6	6	-1	1	1	0	0102000000020
2	6	6	7	7	7	-5	5	1	0	0102000000020
3	7	7	4	-3	3	-6	6	1	0	0102000000020
4	1	1	2	2	2	8	8	2	0	0102000000020
5	4	4	5	-8	8	-7	7	2	0	0102000000020
6	2	2	3	3	3	5	5	2	1	0102000000020
7	3	3	4	4	4	-2	2	2	1	0102000000020
8	8	1	5	-4	4	1	1	0	2	0102000000020

Note that 4 new tables and a view have been created in the toposchema1 Schema.
We will focus on `edge_data` table.

Lab 7
Topology

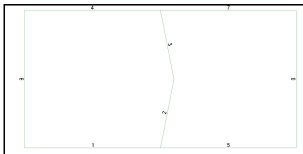


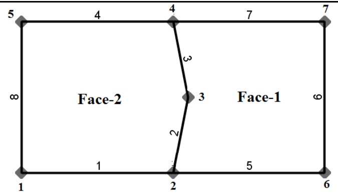
Look at all topology on the system

```
SELECT l.layer_id, l.schema_name, l.table_name, l.feature_column,
 l.feature_type, l.level, l.child_id
from topology.layer l, topology.topology  t where l.topology_id = t.id
order by l.layer_id;
```

View the newly created tables in GQIS, select the `toposchema1.edge_data` table for viewing.

Below left we use edge_id column to label the edges.





We will now run some queries on `toposchema1.edge_data`.
The `topo1` table contains references to the topological geometry.

```
SELECT * FROM topo1;
```

We can access all the geometry of the topology from the table topo1 as follows:

```
SELECTst_asText(topology.Geometry(TopoGeom))FROM topo1;
```

You can view this is QGIS too.

List all the topologies:

```
SELECT * FROM topology.topology;
```

Get the geometry for the start node of edge 6.

```
SELECT  ST_astext(ST_StartPoint(e.geom))FROM  toposchema1.edge_data  e  WHERE
start_node=6;
```

What face contains the start node of edge 6.

```
SELECT  topology.GetFaceByPoint('toposchema1',  ST_StartPoint(e.geom),  1.0)FROM
topo1,toposchema1.edge_data e WHERE start_node=6;
```

Lab 7
Topology

Note error if node is in more than one face.

```
SELECT * FROM topology.ValidateTopology(’toposchema1’);
SELECT * FROM toposchema1.relation;
SELECT ST_equals(f1.mbr,f2.mbr)FROM toposchema1.face f1, toposchema1.face f2;
SELECT ST_NumGeometries(f1.mbr), ST_NumGeometries(f2.mbr)FROM toposchema1.face
f1, toposchema1.face f2;

SELECT ST_NumGeometries(f1.geom), ST_NumGeometries(f2.geom)FROM
toposchema1.edge f1, toposchema1.edge f2;

SELECT ST_NumGeometries(f1.geom), ST_NumGeometries(f2.geom)FROM
toposchema1.node f1, toposchema1.node f2;

SELECT ST_GeometryType(geom)FROM toposchema1.edge_data;
SELECT ST_GeometryType(mbr)FROM toposchema1.face;
SELECT ST_GeometryType(geom)FROM toposchema1.node;
SELECT ST_GeometryType(geom)FROM toposchema1.edge_data;
SELECT ST_GeometryType(mbr)FROM toposchema1.face;
SELECT ST_GeometryType(geom)FROM toposchema1.node;

-- Accessing geometry is a little indirect.
SELECT topology.GetTopoGeomElements(topogeom) FROM topo1

SELECT COUNT(face_id) As num_faces,
max(ST_Area(ST_GetFaceGeometry('toposchema1',face_id))) As area
FROM (SELECT (GetTopoGeomElements(topogeom))[2] As face_id FROM topo1)  AS x;

-- The topology can be exported as GML
SELECTtopology.AsGML(topogeom)FROM topo1;
```