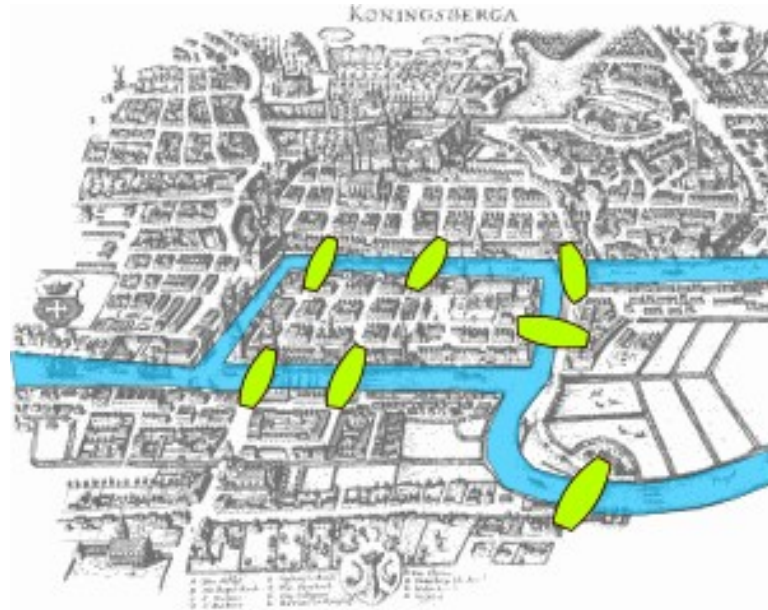


PostGIS Topology



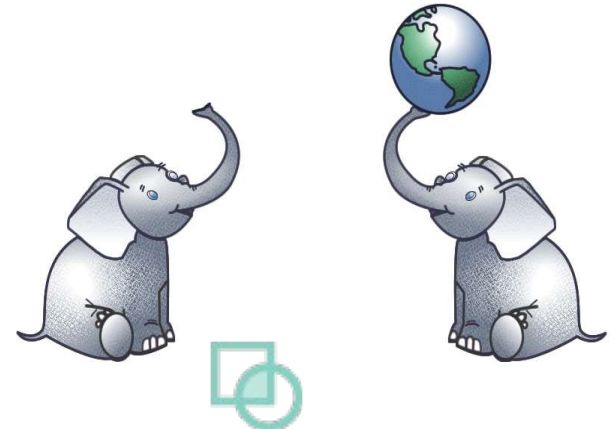
Sandro Santilli <strk@kbt.io>

- <http://strk.kbt.io> -

Wellington 2017

PostGIS topology timeline

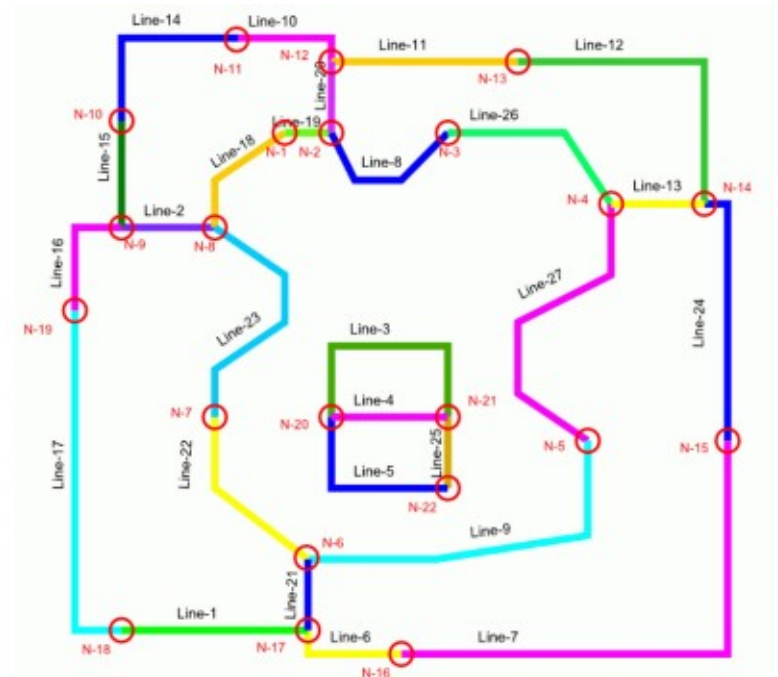
- Presented at FOSS4G 2006
PostGIS: future developments
- Drafted in PostGIS 1.1.0
Based on ISO SQL/MM
- Integrated in 2010
Testsuite, build scripts, packaging
- Further Improved in 2011
Full SQL/MM implementation,
more utility functions
- Prime-time in PostGIS 2.0.0 (2012)
QGIS support



Regione Toscana

Why topology ?

- Normalized spatial data
- Standard interface
- Topological integrity
- Reduced storage size
- Explicit spatial relationships



Why topology ?

Topological integrity

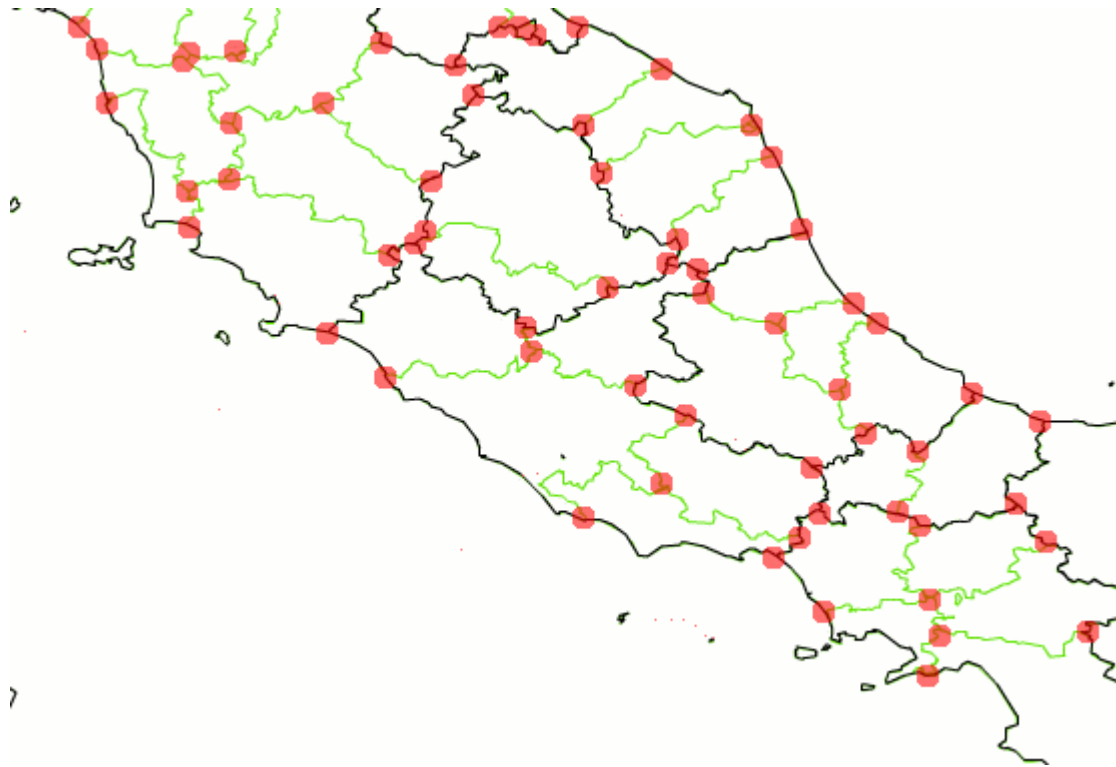
- Every intersection is a node



Why topology ?

Topological integrity

- Every intersection is a node



Why topology ?

Topological integrity

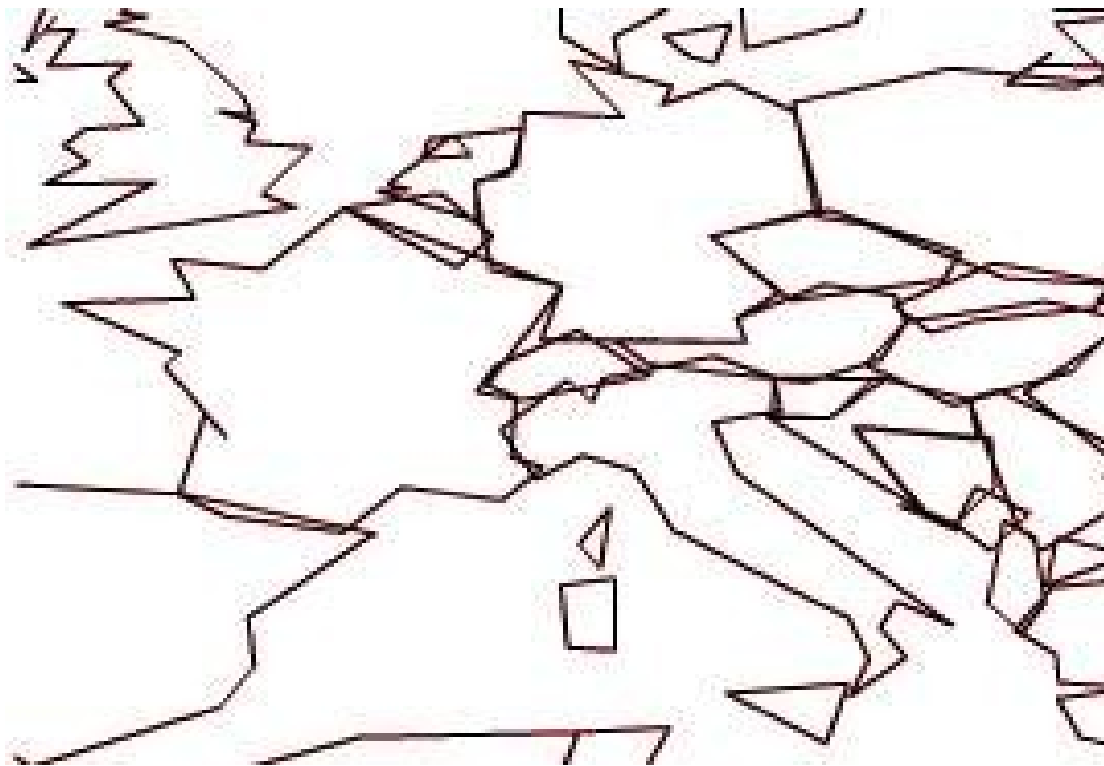
- Edges are **shared** ...



Why topology ?

Topological integrity

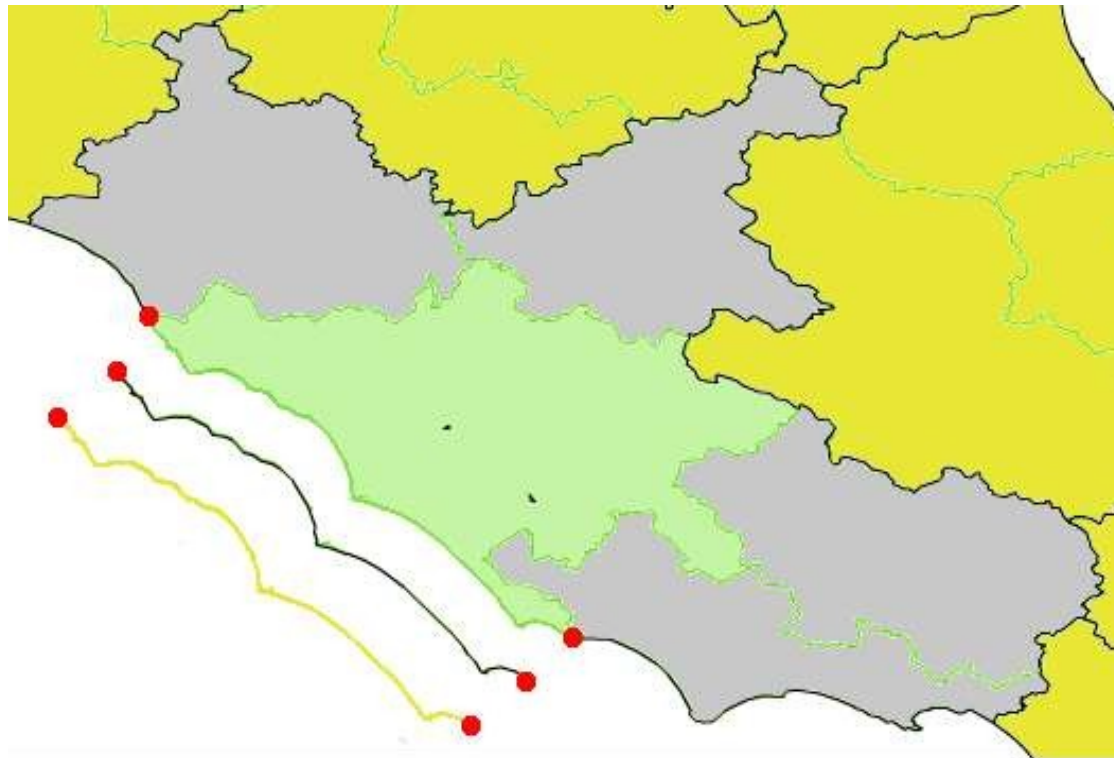
- ... rather than **separate** entities



Why topology ?

Reduced storage size

- Every edge is stored only **once**



Why topology ?

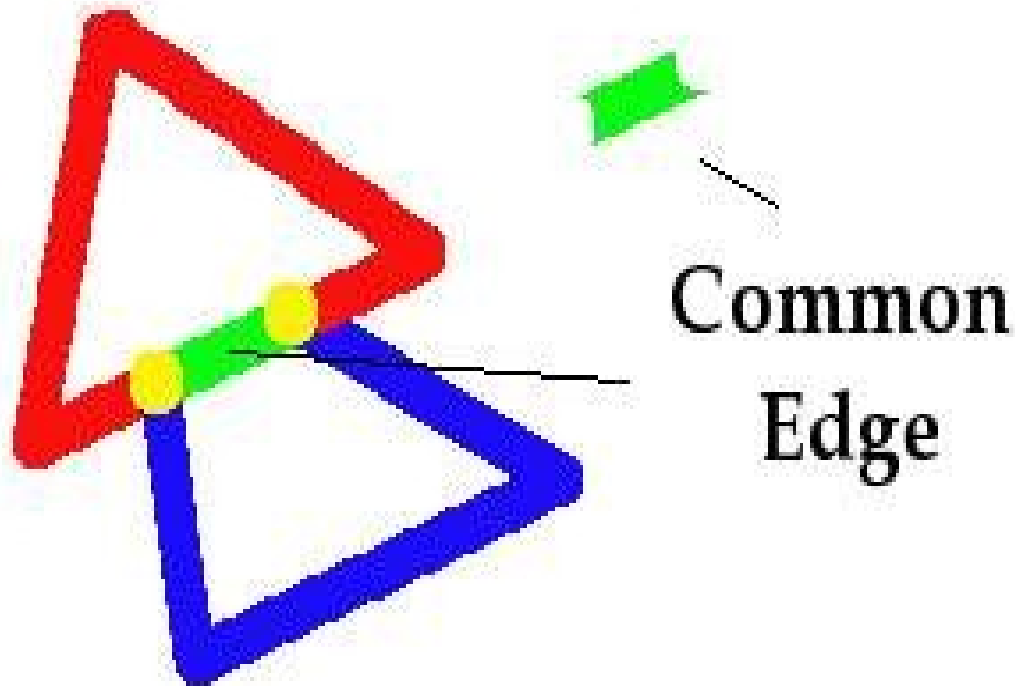
Reduced storage size

- Features in hierarchical layers can be defined by **composition**. For example:
 - A state is a collection of regions
 - A region is a collection of provinces
 - A province is a collection of municipalities
 -

Why topology ?

Explicit spatial relationships

- Do they touch ? YES !



Why topology ?

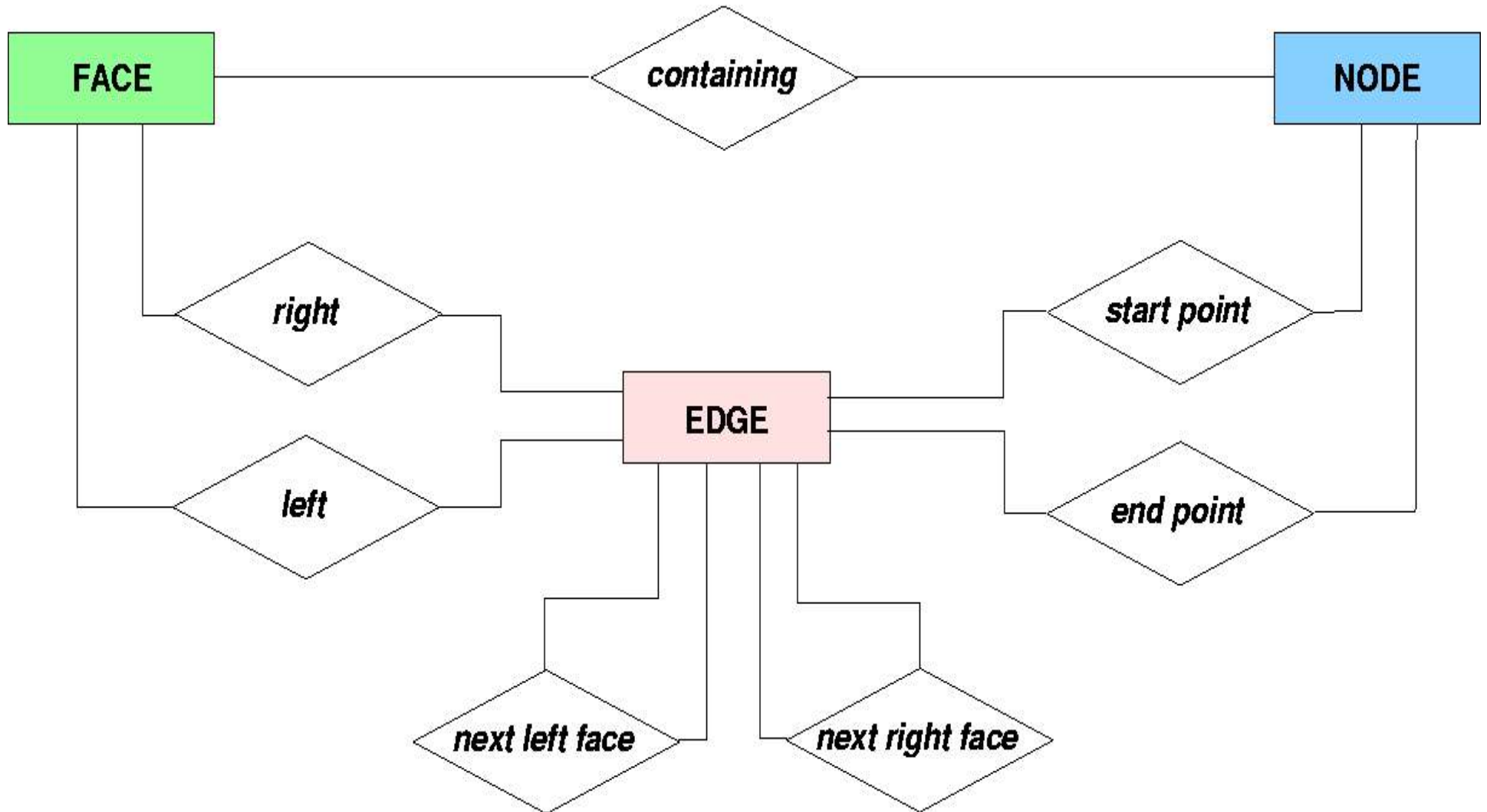
Explicit spatial relationships

- For each edge you know the right and left face
- For each isolated node you know the face it's in
- Every intersection is a node
- Nodes are shared

Conceptual Model

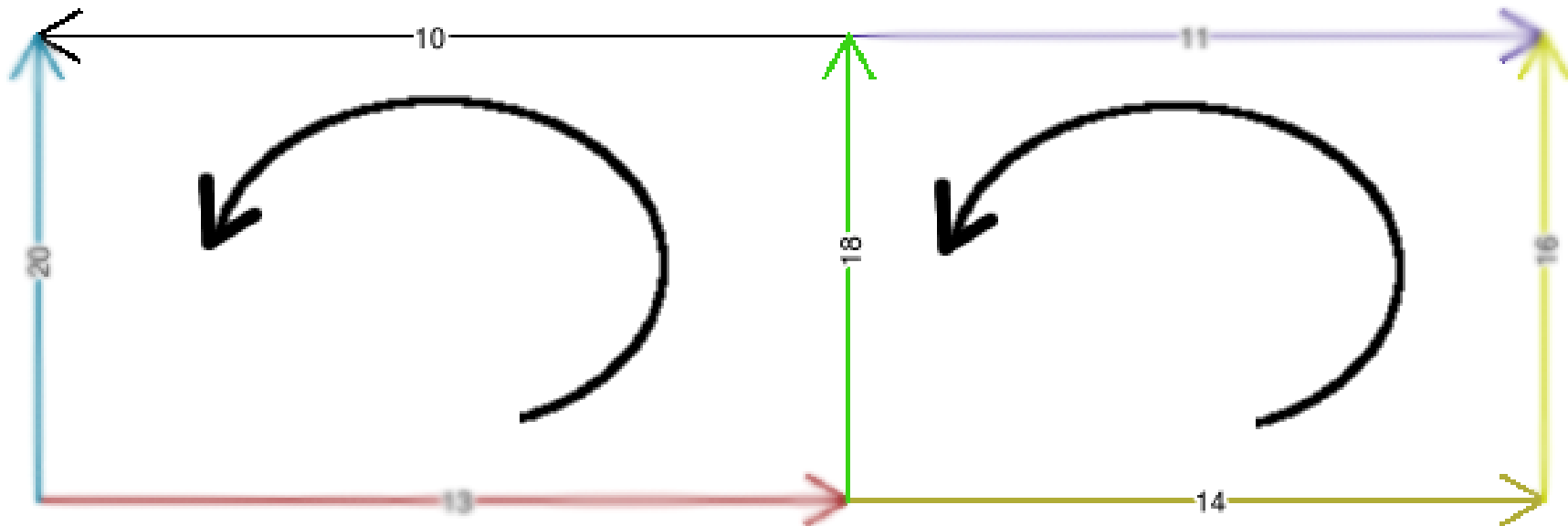
Conceptual model

Faces, Edges and Nodes

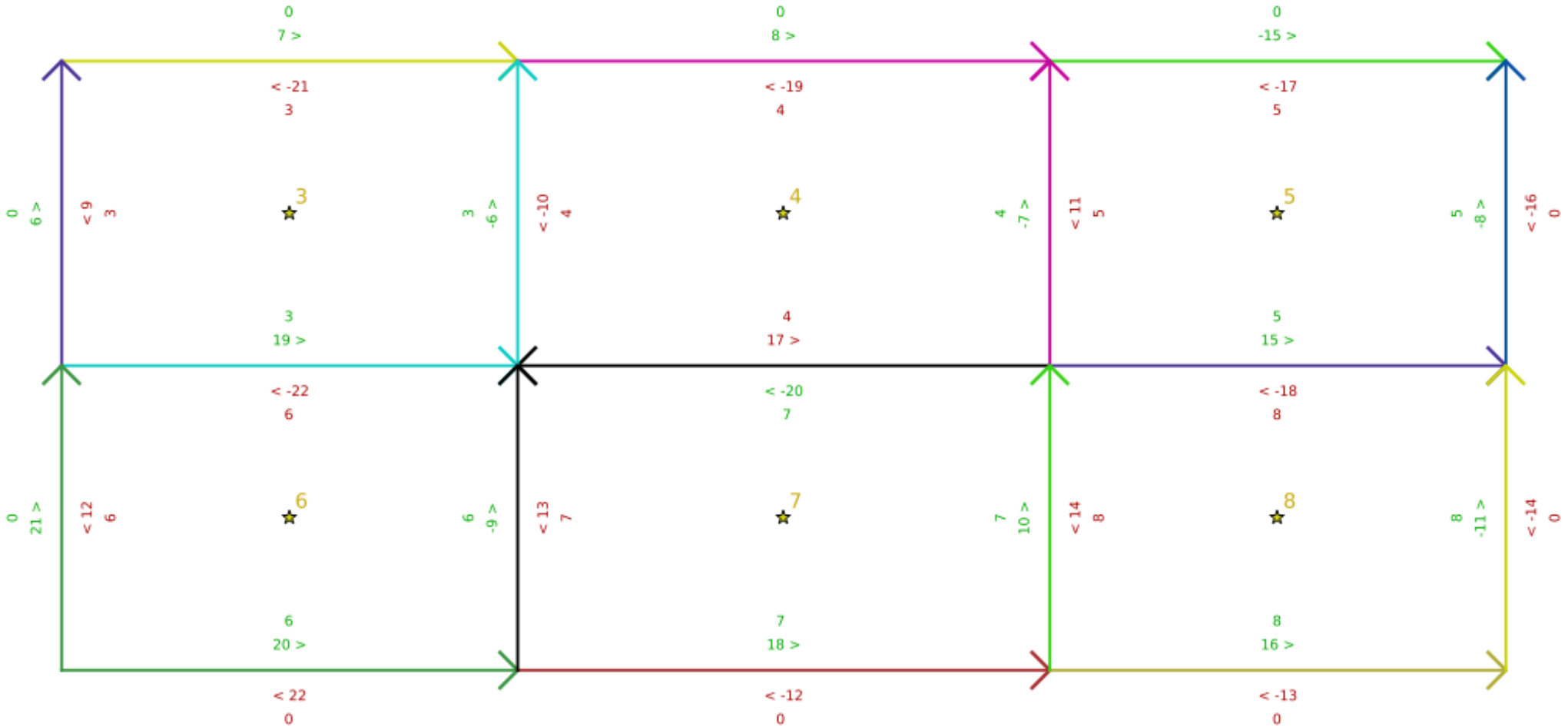


Conceptual model

Next right and left face edges

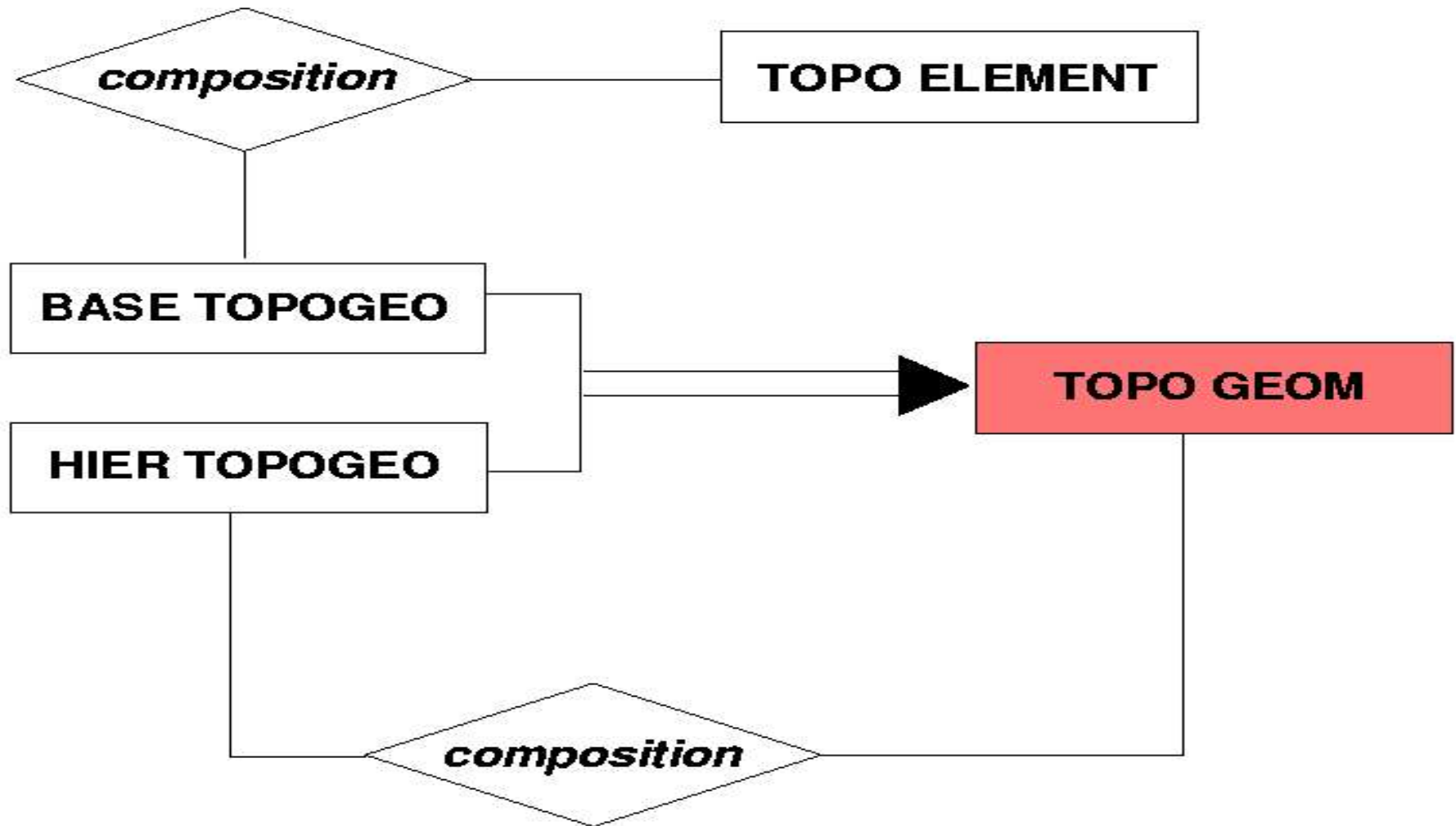


Conceptual model



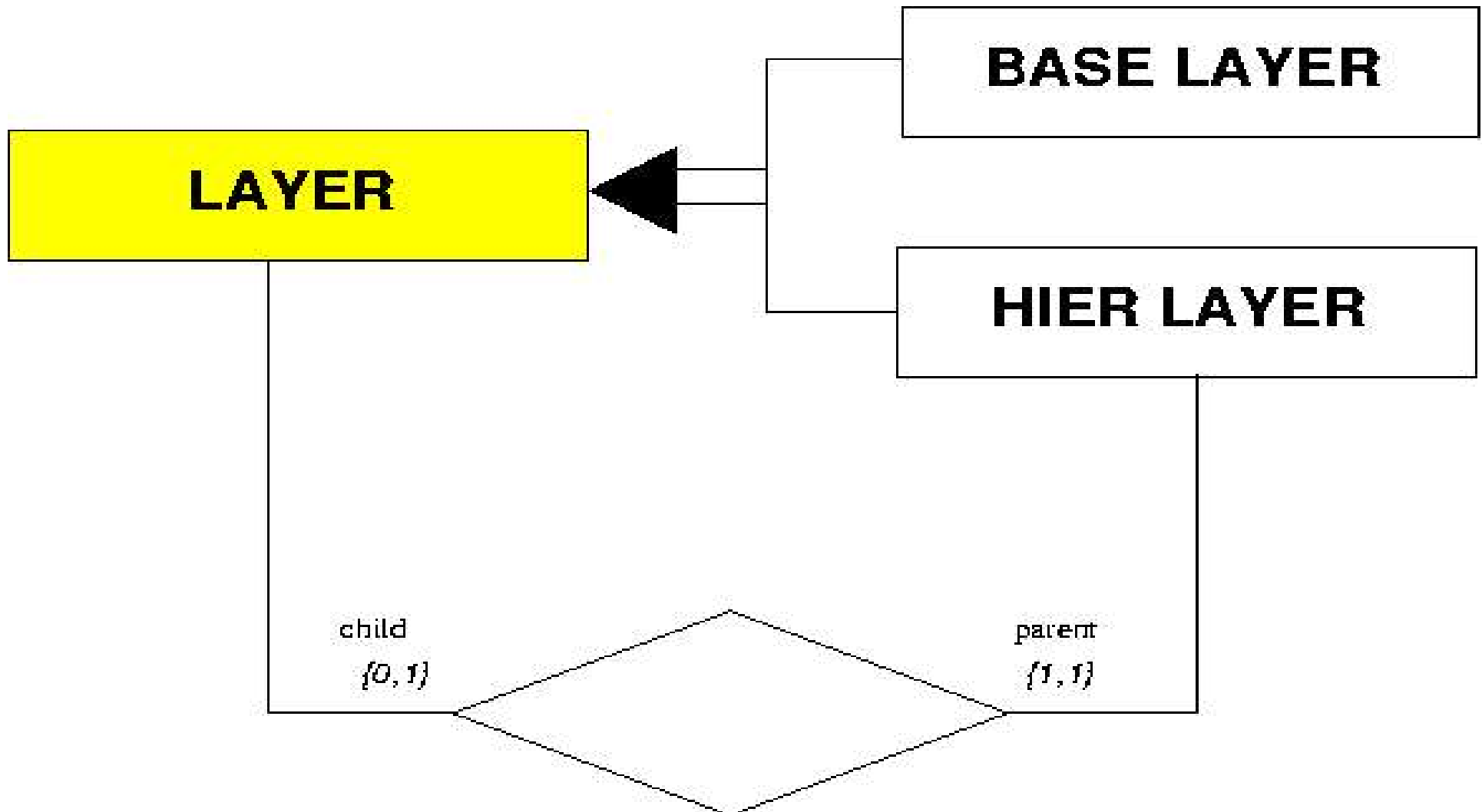
Conceptual model

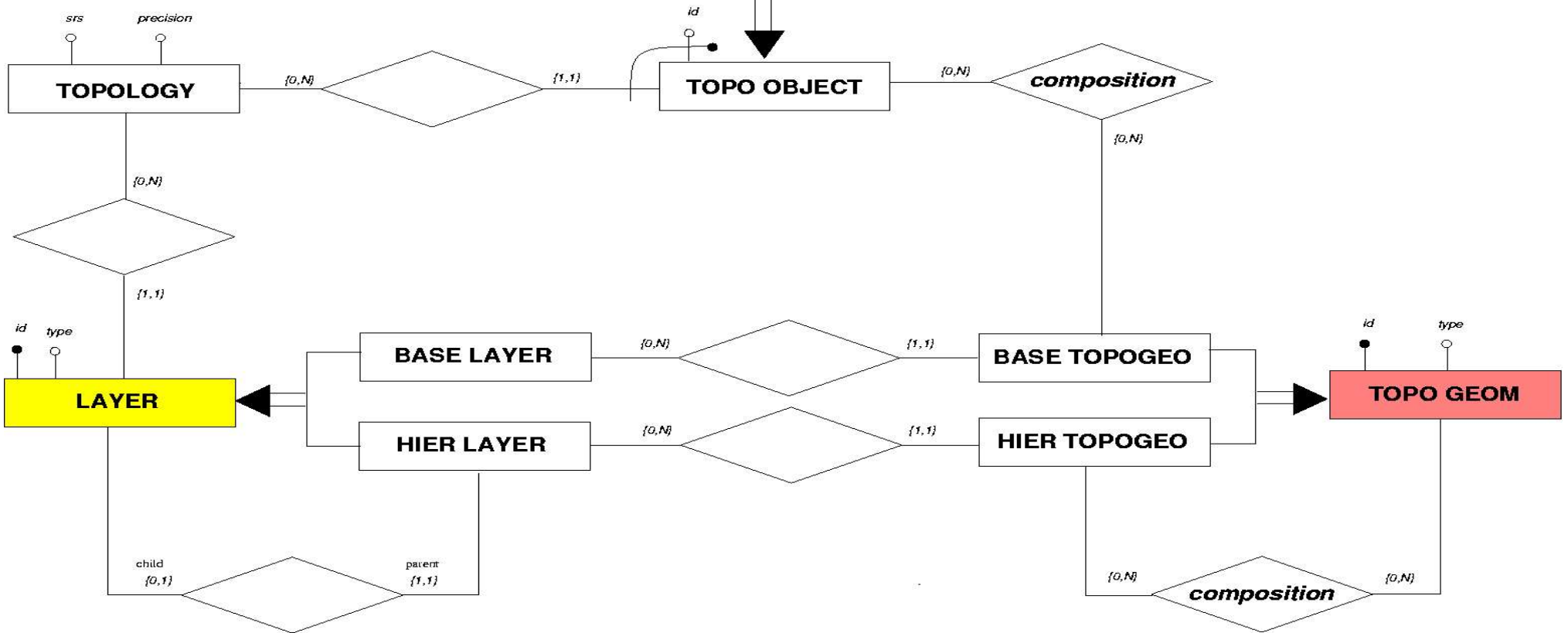
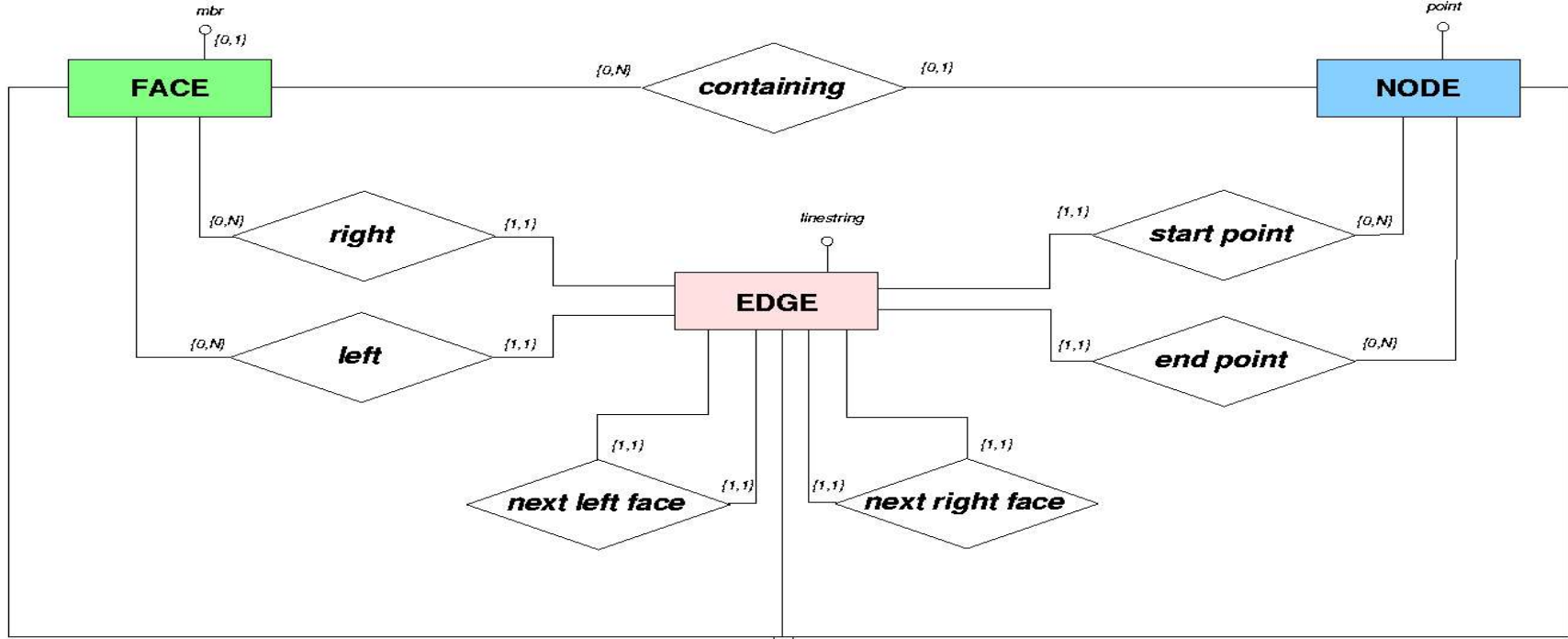
Topo-Geometries



Conceptual model

Layers





Physical Model

Physical model

- All routines, types and other management objects are stored in the "topology" schema
- Each topology is stored in its own schema
- Metadata tables with informations about available topologies and topological layers
- TopoGeometry datatype

Topology schema

- <name>.edge
- <name>.face
- <name>.node
- <name>.relation (TopoGeometry composition)

Physical model

Metadata tables

- topology.topology
- topology.layer

Functions

- Create, destroy, copy, summarize topologies
- Edit and validate topologies (ISO SQL/MM and more)
- Define **Layers** and **TopoGeometries** (simple and hierarchical)
- TopoGeometry <-> Geometry converters
- Topological GML and TopoJSON outputs

Topologies management

- CreateTopology
- ValidateTopology
- TopologySummary
- CopyTopology
- DropTopology

SQL/MM topology editing

- **Add an isolated node:** ST_AddIsoNode
- **Move an isolated node:** ST_MovelsoNode
- **Remove an isolated node:** ST_RemovelsoNode
- **Remove an isolated edge:** ST_RemovelsoEdge
- **Add an isolated edge:** AddIsoEdge
- **Add an edge:** ST_AddEdgeNewFaces, ST_AddEdgeModFace
- **Remove an edge:** ST_RemEdgeNewFace, ST_RemEdgeModFace
- **Change an edge shape:** ST_ChangeEdgeGeom
- **Merge two edges:** ST_ModEdgeHeal, ST_NewEdgesHeal
- **Split an edge:** ST_ModEdgeSplit, ST_NewEdgesSplit

Layers and TopoGeometries

- AddTopoGeometryColumn
- DropTopoGeometryColumn
- CreateTopoGeom (by element)
- toTopoGeom (by geometry)

Importing geometries

- TopoGeo_AddPoint
- TopoGeo_AddLineString
- TopoGeo_AddPolygon

Topology inspection

- GetNodeByPoint
- GetEdgeByPoint
- GetFaceByPoint
- GetRingEdges – walk around a ring of edges
- GetNodeEdges – spin on a node
- GetFaceGeometry

TopoGeometry outputs

- AsGML – with xref support
- AsTopoJSON

Examples

Topology creation and review

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

```
=> SELECT CreateTopology('conf');
```

```
=> SELECT TopologySummary('conf');
```

```
Topology conf (1), SRID -1, precision 0
```

```
0 nodes, 0 edges, 1 faces, 0 topogeoms in 0 layers
```

ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
    'conf', 0,  
    'POINT(10 20) '  
);
```


ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
    'conf', 0,  
    'POINT(10 20) '  
);
```



ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 20) '  
);
```



```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(100 20) '  
);
```

ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 20) '  
);
```

①

②

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(100 20) '  
);
```

ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 20) '  
);
```

①

②

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(100 20) '  
);
```

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 -90) '  
);
```

ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 20) '  
);
```

①

②

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(100 20) '  
);
```

```
=> SELECT ST_AddIsoNode (  
  'conf', 0,  
  'POINT(10 -90) '  
);
```

③

ISO SQL/MM Topology Population

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 1, 2,  
  'LINESTRING(10 20,  
  100 20) '  
);
```

1

2

3

ISO SQL/MM Topology Population

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 1, 2,  
  'LINESTRING(10 20,  
  100 20) '  
);
```



ISO SQL/MM Topology Population

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 1, 2,  
  'LINESTRING(10 20,  
  100 20) '  
);
```



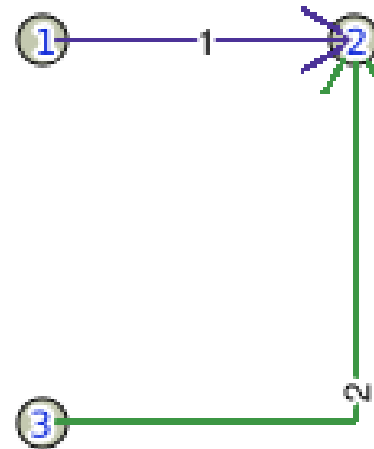
```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 3, 2,  
  'LINESTRING(10 -90,  
  100 -90, 100 20) '  
);
```



ISO SQL/MM Topology Population

```
=> SELECT
ST_AddEdgeModFace (
  'conf', 1, 2,
  'LINESTRING(10 20,
  100 20) '
);

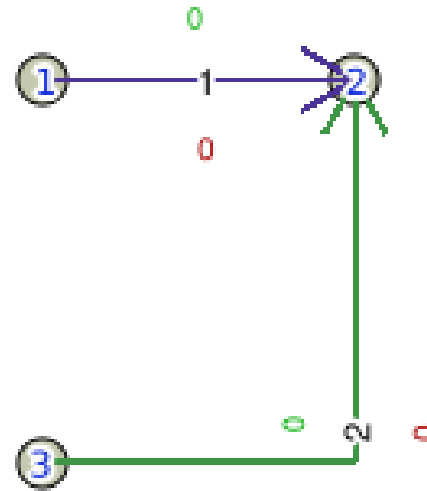
=> SELECT
ST_AddEdgeModFace (
  'conf', 3, 2,
  'LINESTRING(10 -90,
  100 -90, 100 20) '
);
```



ISO SQL/MM Topology Population

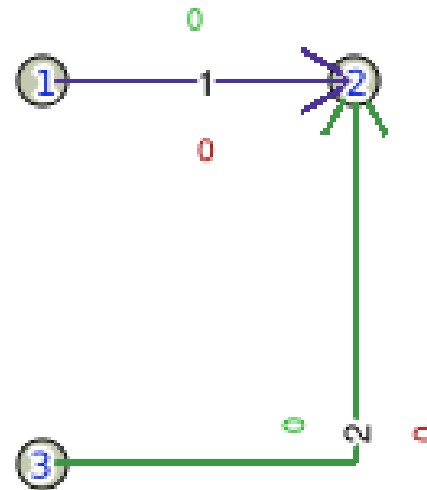
```
=> SELECT
ST_AddEdgeModFace (
  'conf', 1, 2,
  'LINESTRING(10 20,
  100 20) '
);

=> SELECT
ST_AddEdgeModFace (
  'conf', 3, 2,
  'LINESTRING(10 -90,
  100 -90, 100 20) '
);
```



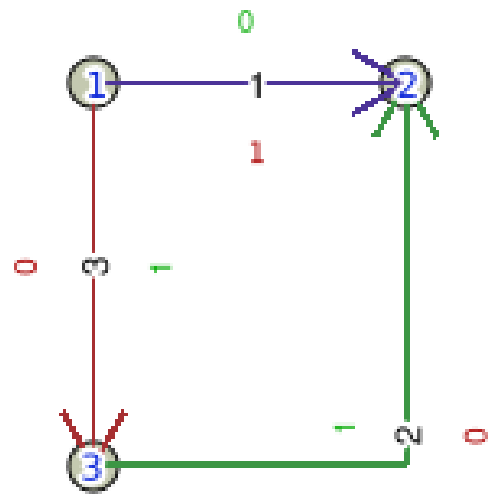
ISO SQL/MM Topology Population

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 1, 3,  
  'LINESTRING(10 20,  
  10 -90) '  
);
```



ISO SQL/MM Topology Population

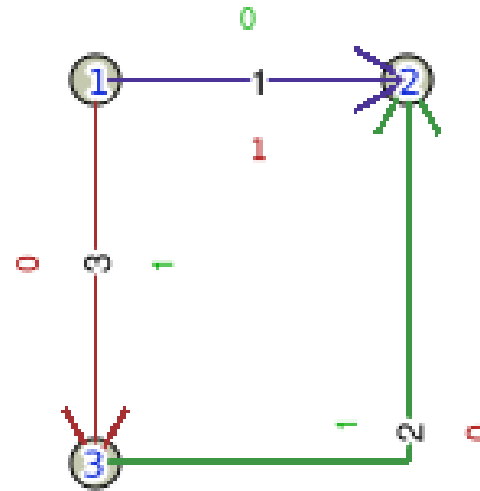
```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 1, 3,  
  'LINESTRING(10 20,  
  10 -90) '  
);
```



ISO SQL/MM Topology Population

```
=> SELECT
ST_AddEdgeModFace (
  'conf', 1, 3,
  'LINESTRING(10 20,
  10 -90) '
);

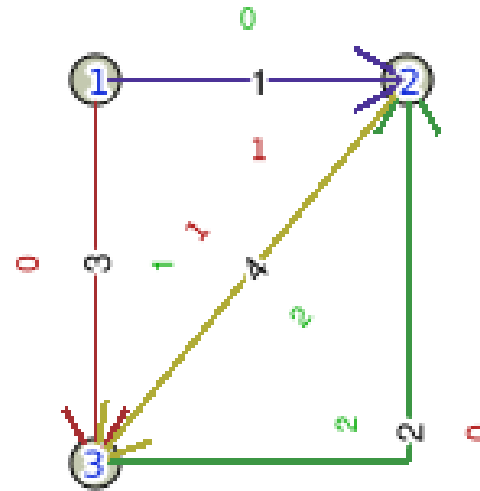
=> SELECT
ST_AddEdgeModFace (
  'conf', 2, 3,
  'LINESTRING(100 20,
  10 -90) '
);
```



ISO SQL/MM Topology Population

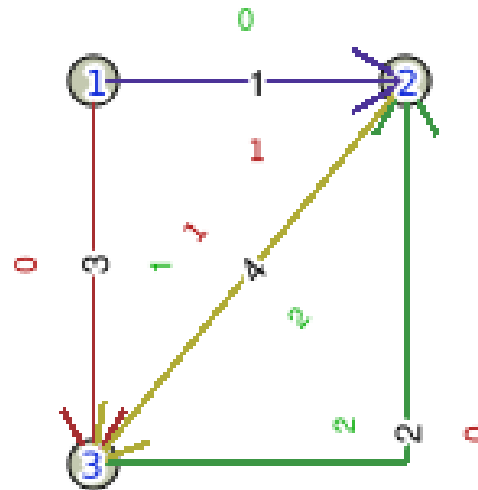
```
=> SELECT
ST_AddEdgeModFace (
  'conf', 1, 3,
  'LINESTRING(10 20,
  10 -90) '
);

=> SELECT
ST_AddEdgeModFace (
  'conf', 2, 3,
  'LINESTRING(100 20,
  10 -90) '
);
```



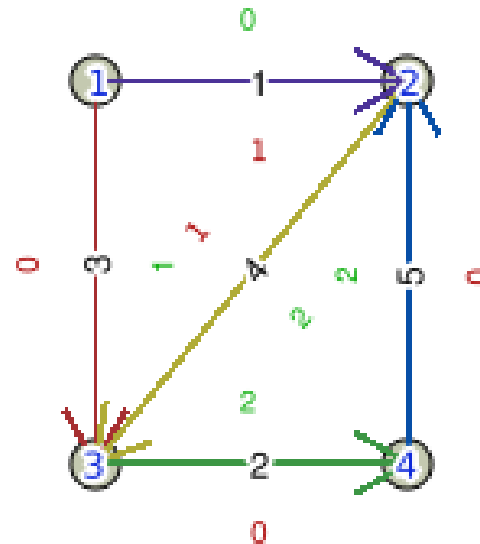
ISO SQL/MM Topology Population

```
=> SELECT  
ST_ModEdgeSplit(  
  'conf', 2,  
  'POINT(100 -90)'  
);
```



ISO SQL/MM Topology Population

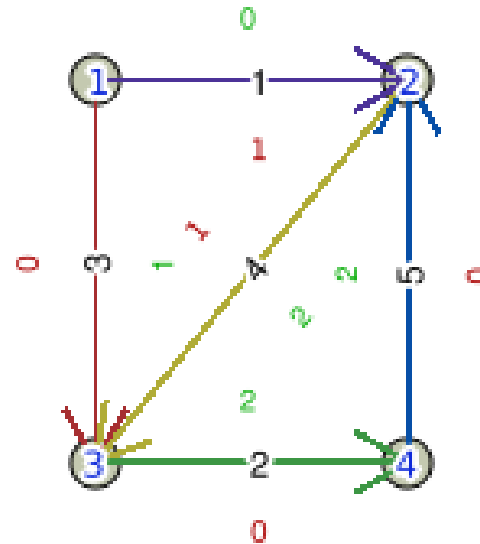
```
=> SELECT  
ST_ModEdgeSplit(  
  'conf', 2,  
  'POINT(100 -90)'  
);
```



ISO SQL/MM Topology Population

```
=> SELECT  
ST_ModEdgeSplit (  
  'conf', 2,  
  'POINT(100 -90) '  
);
```

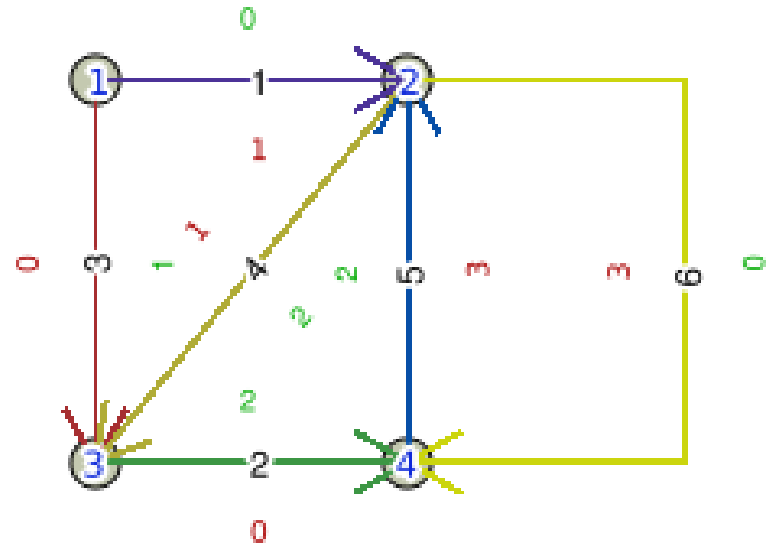
```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 2, 4,  
  'LINESTRING(100 20,  
180 20, 180 -90,  
100 -90) ');
```



ISO SQL/MM Topology Population

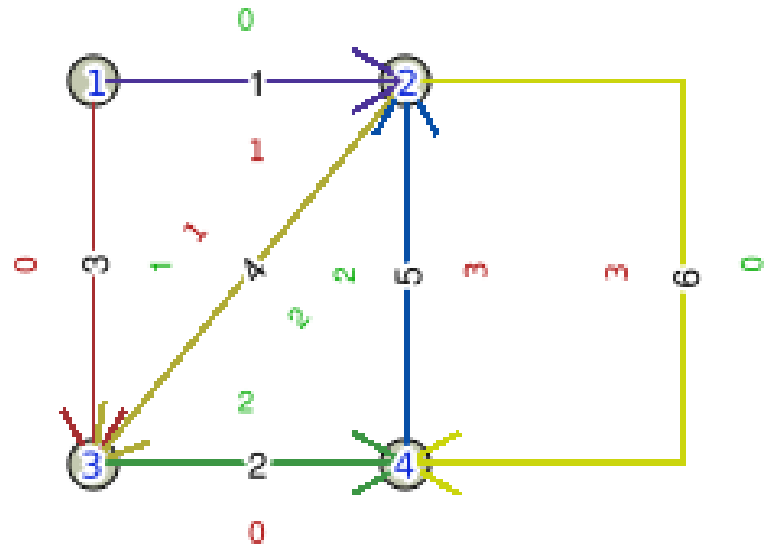
```
=> SELECT  
ST_ModEdgeSplit (  
  'conf', 2,  
  'POINT(100 -90) '  
);
```

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 2, 4,  
  'LINESTRING(100 20,  
180 20, 180 -90,  
100 -90) ');
```



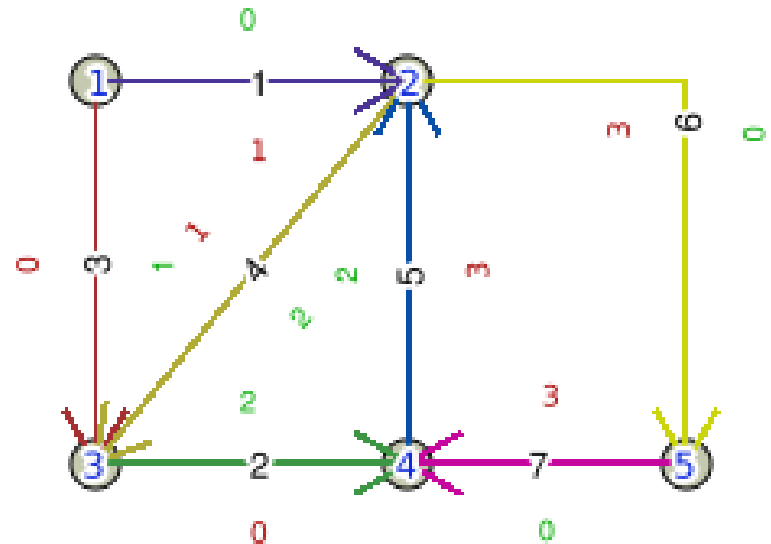
ISO SQL/MM Topology Population

```
=> SELECT  
ST_ModEdgeSplit(  
  'conf', 6,  
  'POINT(180 -90)'  
);
```



ISO SQL/MM Topology Population

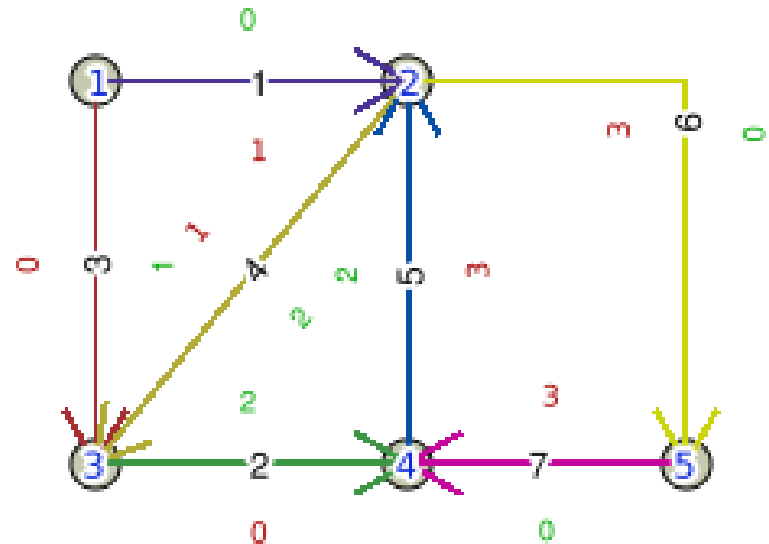
```
=> SELECT  
ST_ModEdgeSplit(  
  'conf', 6,  
  'POINT(180 -90)'  
);
```



ISO SQL/MM Topology Population

```
=> SELECT  
ST_ModEdgeSplit (  
  'conf', 6,  
  'POINT(180 -90) '  
);
```

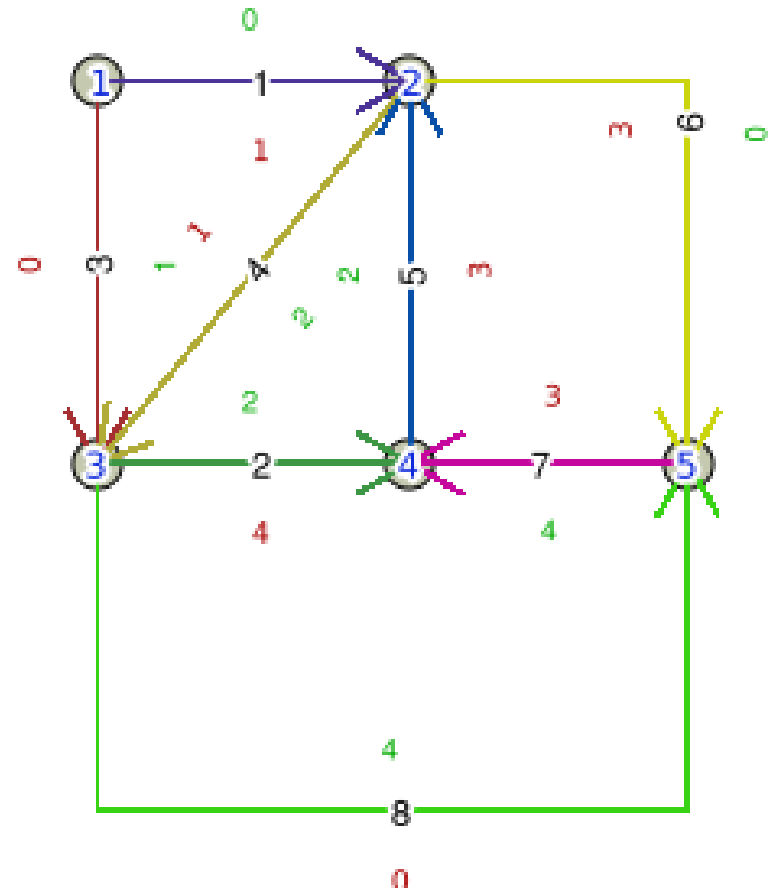
```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 3, 5,  
  'LINESTRING(10 -90,  
10 -190, 180 -190,  
180 -90) ');
```



ISO SQL/MM Topology Population

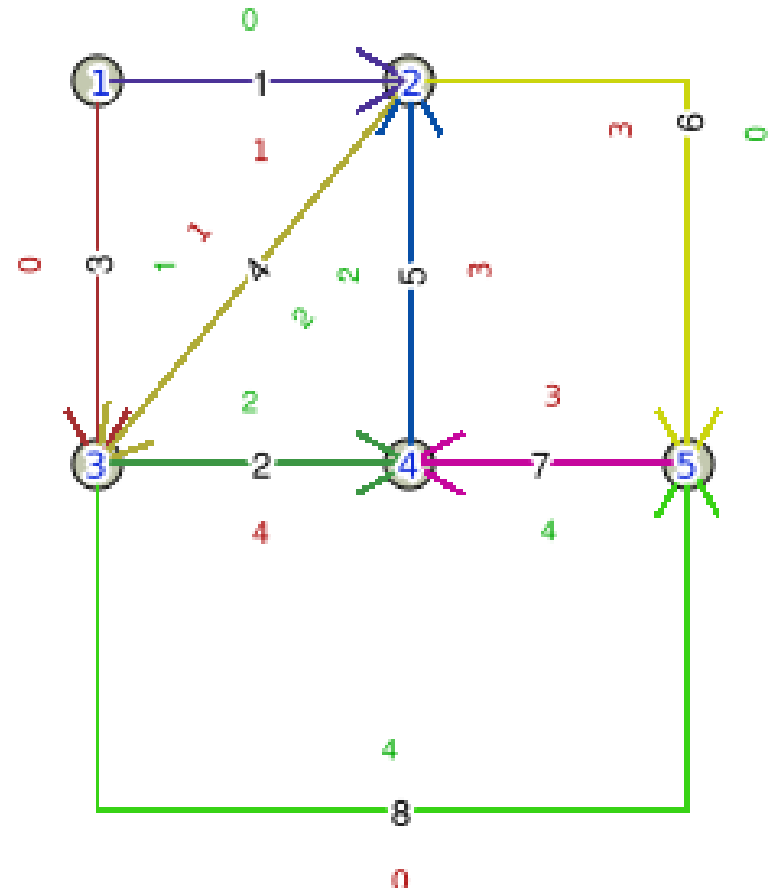
```
=> SELECT  
ST_ModEdgeSplit (  
  'conf', 6,  
  'POINT(180 -90) '  
) ;
```

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 3, 5,  
  'LINESTRING(10 -90,  
  10 -190, 180 -190,  
  180 -90) ' ) ;
```



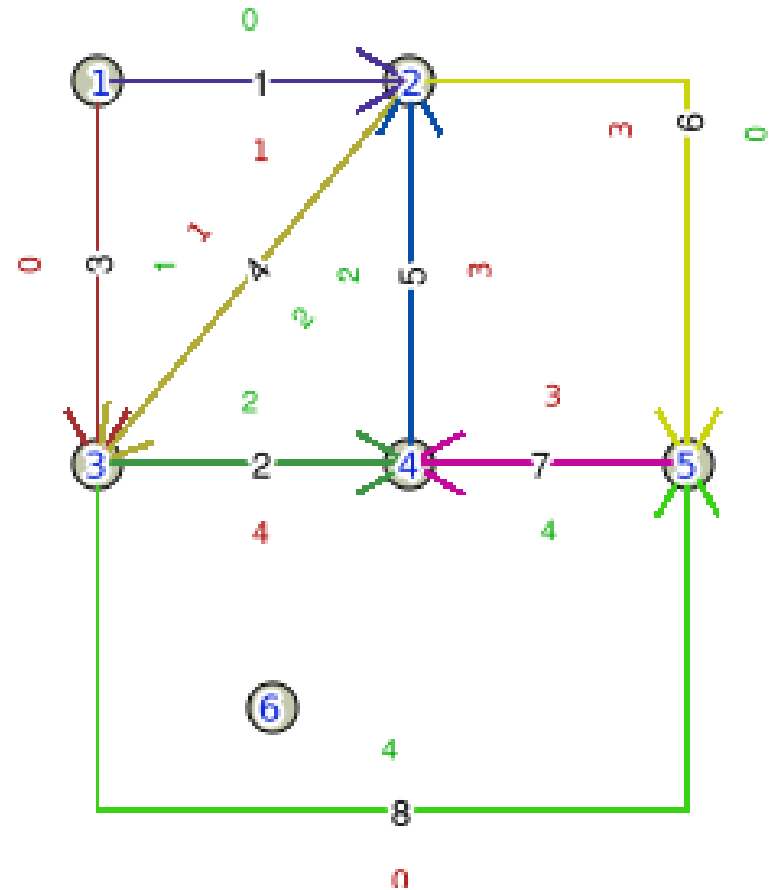
ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 4,  
  'POINT(60 -160) '  
);
```



ISO SQL/MM Topology Population

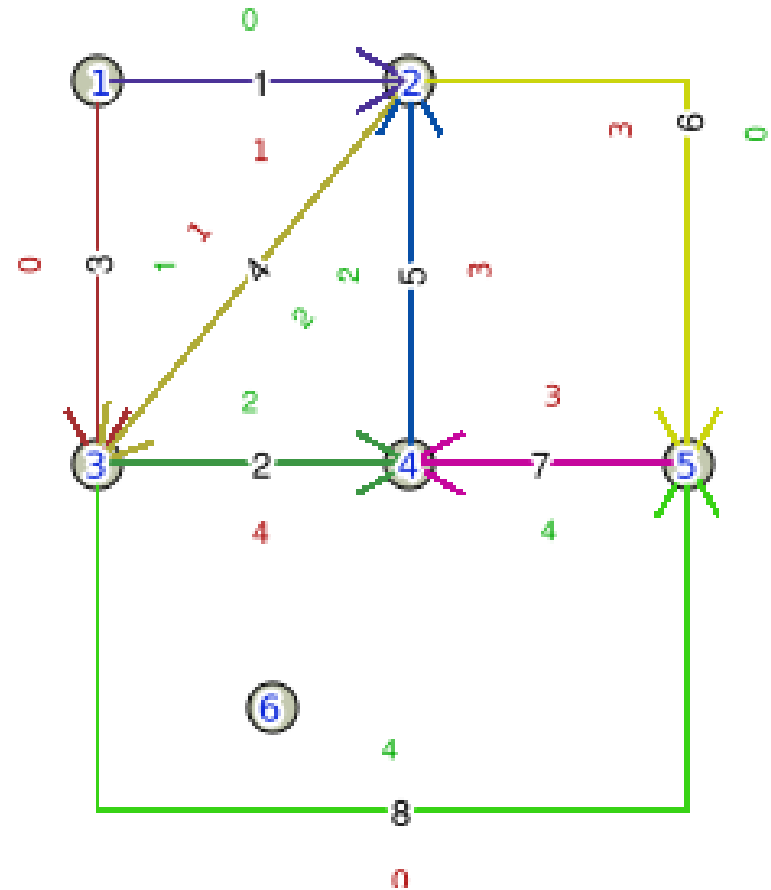
```
=> SELECT ST_AddIsoNode (  
  'conf', 4,  
  'POINT(60 -160) '  
);
```



ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 4,  
  'POINT(60 -160) '  
);
```

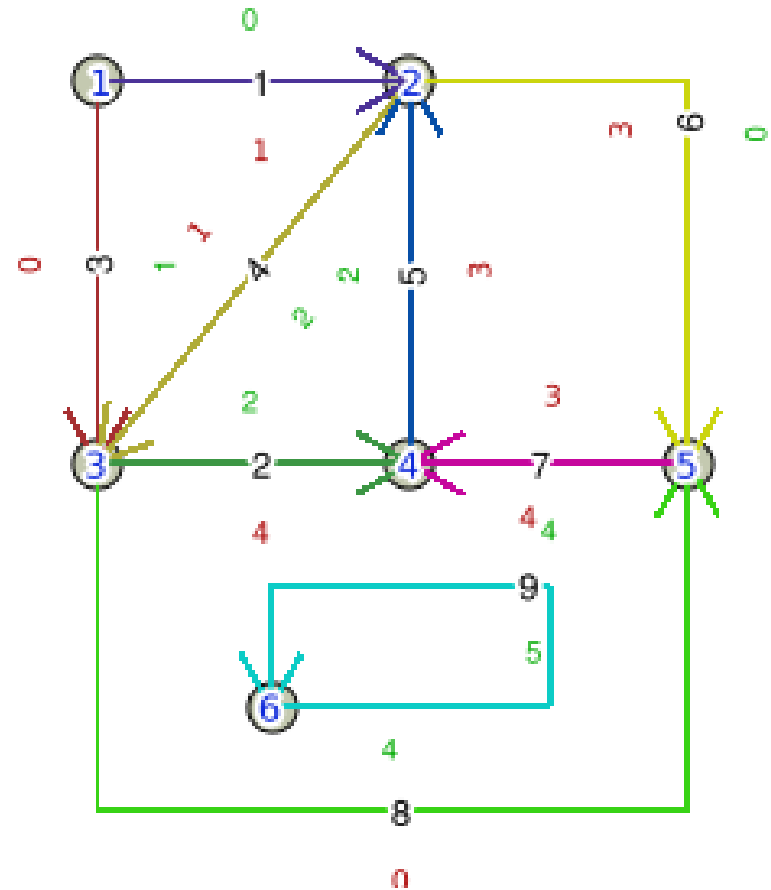
```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 6, 6,  
  'LINESTRING(60 -160,  
140 -160, 140 -125,  
60 -125, 60 -160) ');
```



ISO SQL/MM Topology Population

```
=> SELECT ST_AddIsoNode (  
  'conf', 4,  
  'POINT(60 -160) '  
);
```

```
=> SELECT  
ST_AddEdgeModFace (  
  'conf', 6, 6,  
  'LINESTRING(60 -160,  
140 -160, 140 -125,  
60 -125, 60 -160) ');
```

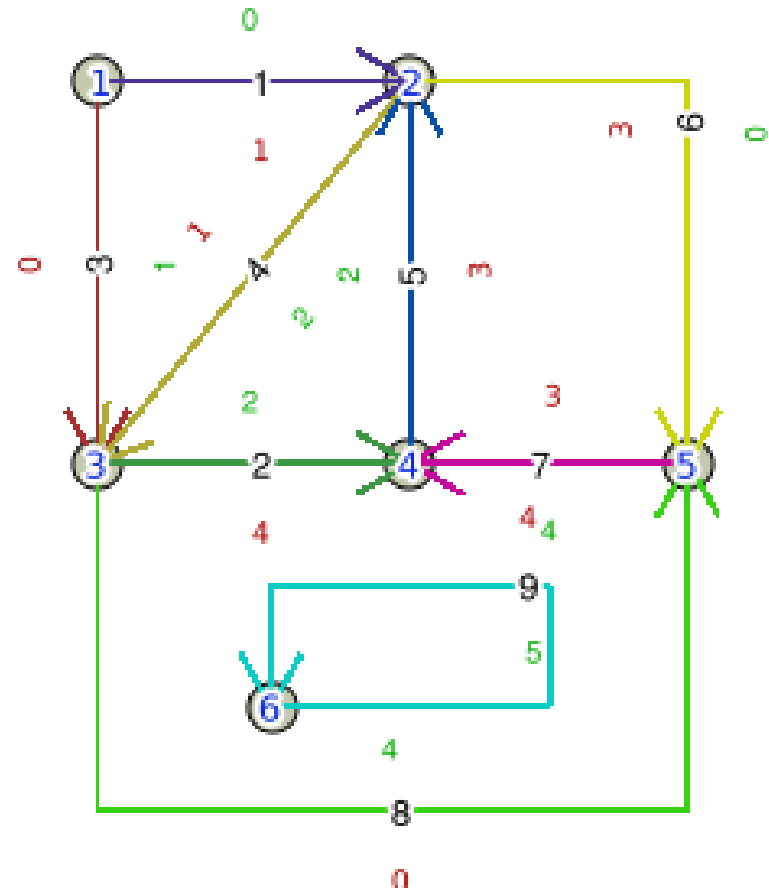


ISO SQL/MM Topology Population

```
=> SELECT  
TopologySummary('conf');
```

```
Topology conf (1),  
SRID -1, precision 0
```

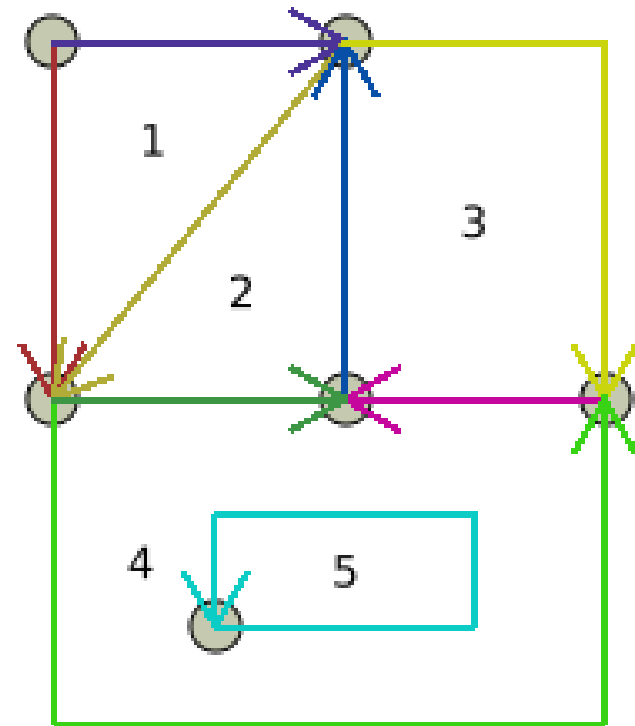
```
6 nodes, 9 edges, 6 faces,  
0 topogeoms in 0 layers
```



TopoGeometry: areal

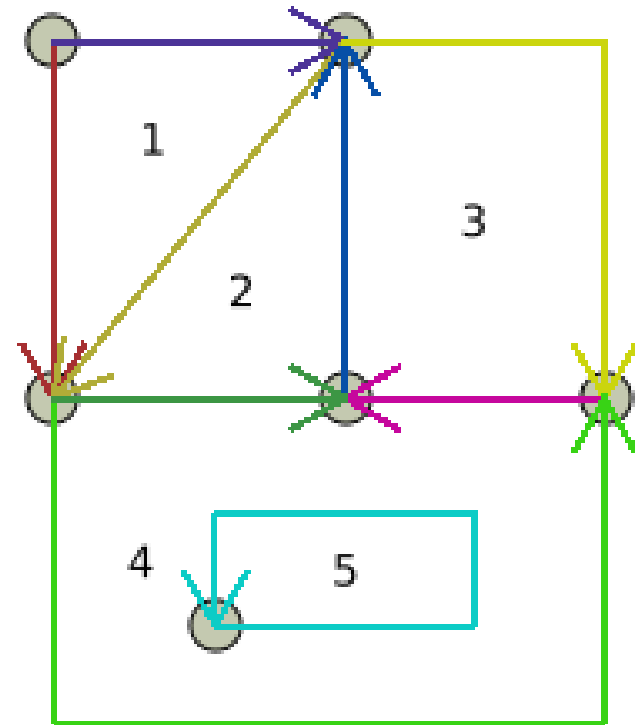
```
=> CREATE TABLE  
conf.fa(id SERIAL  
PRIMARY KEY);
```

```
=> SELECT  
AddTopoGeometryColumn(  
'conf', 'conf', 'fa',  
'g', 'POLYGON');
```



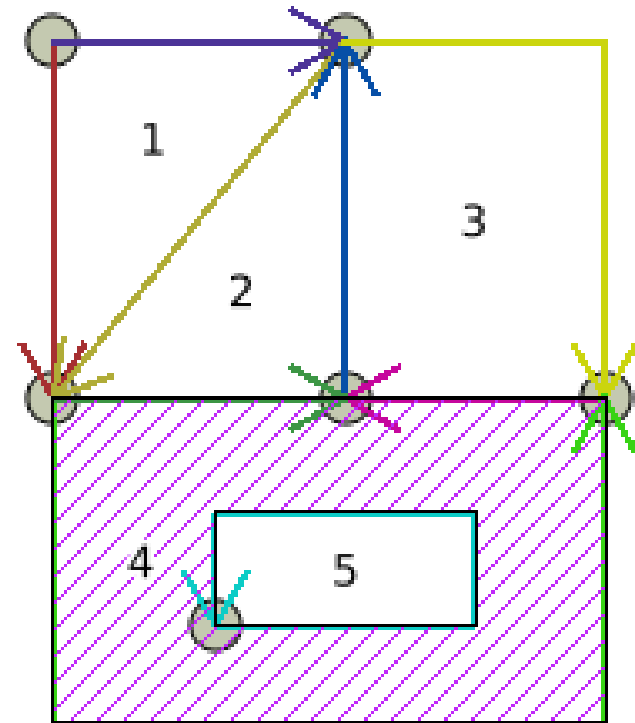
TopoGeometry: areal

```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      3, -- type (areal)
      1, -- layer id
      '{{4,3}}' -- face 4
    )
  );
```



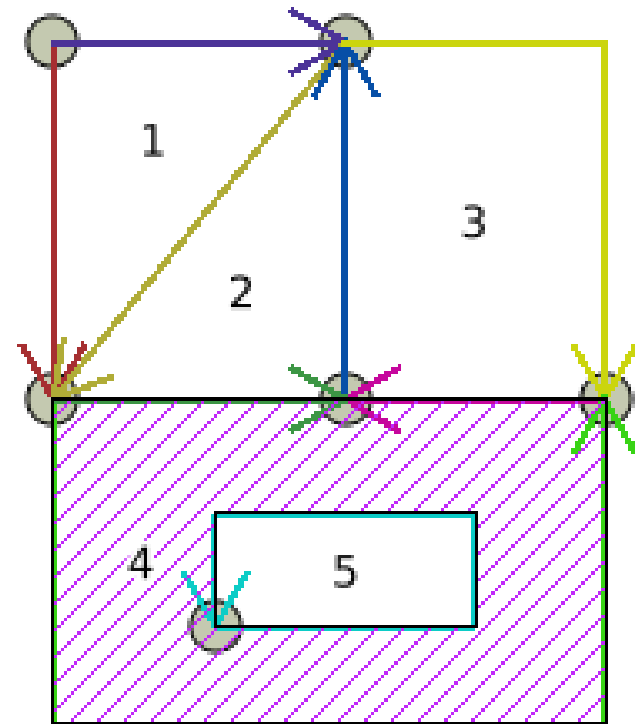
TopoGeometry: areal

```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      3, -- type (areal)
      1, -- layer id
      '{{4,3}}' -- face 4
    )
  );
```



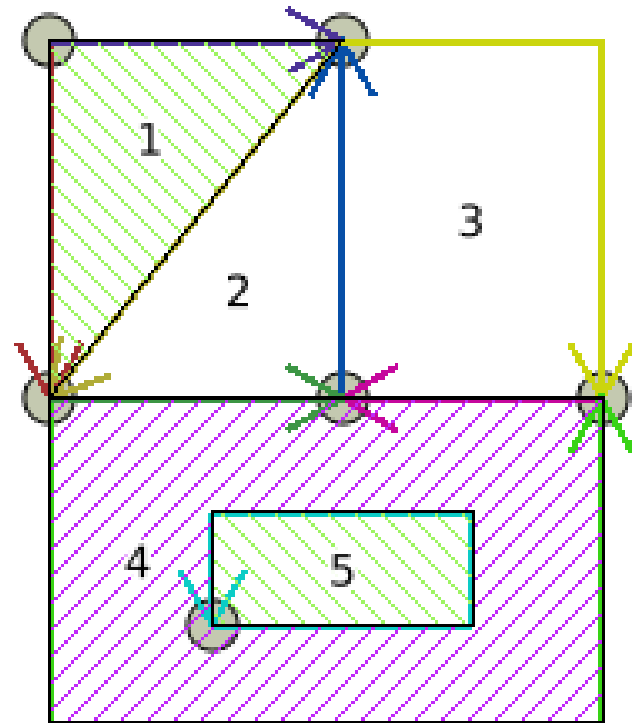
TopoGeometry: areal

```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', 3, 1,
      -- faces 1 and 5
      '{{1,3},{5,3}}'
    )
  );
```



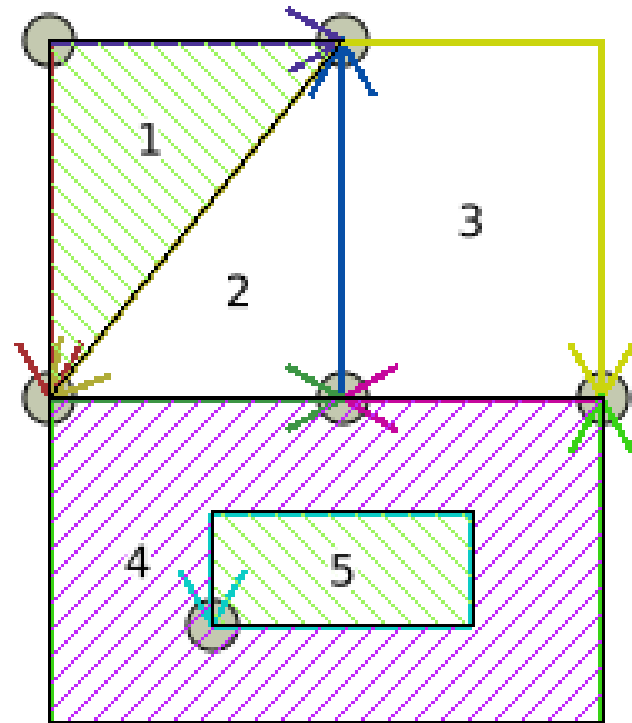
TopoGeometry: areal

```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', 3, 1,
      -- faces 1 and 5
      '{{1,3},{5,3}}'
    )
  );
```



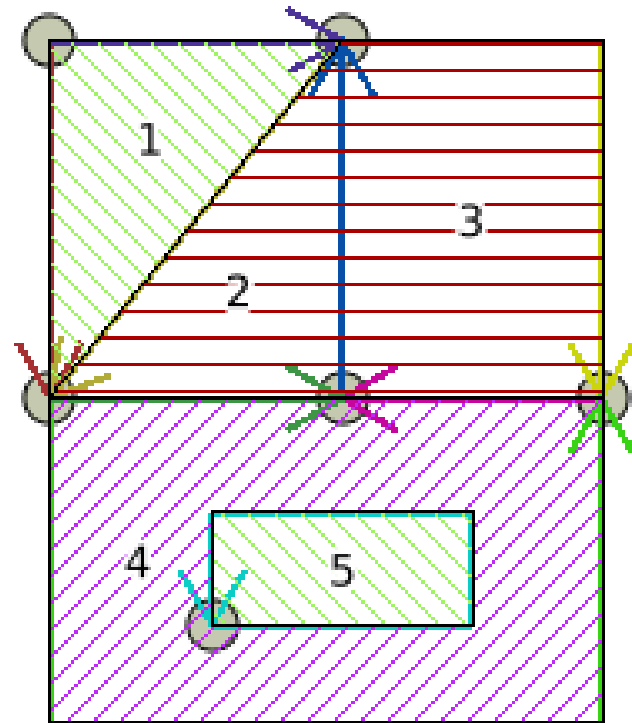
TopoGeometry: areal

```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', 3, 1,
      -- faces 2 and 3
      '{{2,3},{3,3}}'
    )
  );
```



TopoGeometry: areal

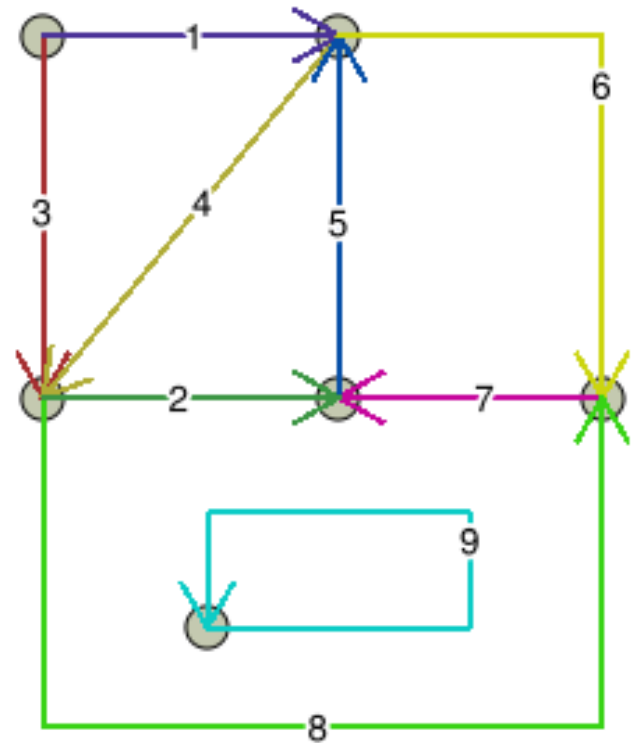
```
=> INSERT
  INTO conf.fa (g)
  VALUES (
    CreateTopoGeom(
      'conf', 3, 1,
      -- faces 2 and 3
      '{{2,3},{3,3}}'
    )
  );
```



TopoGeometry: lineal

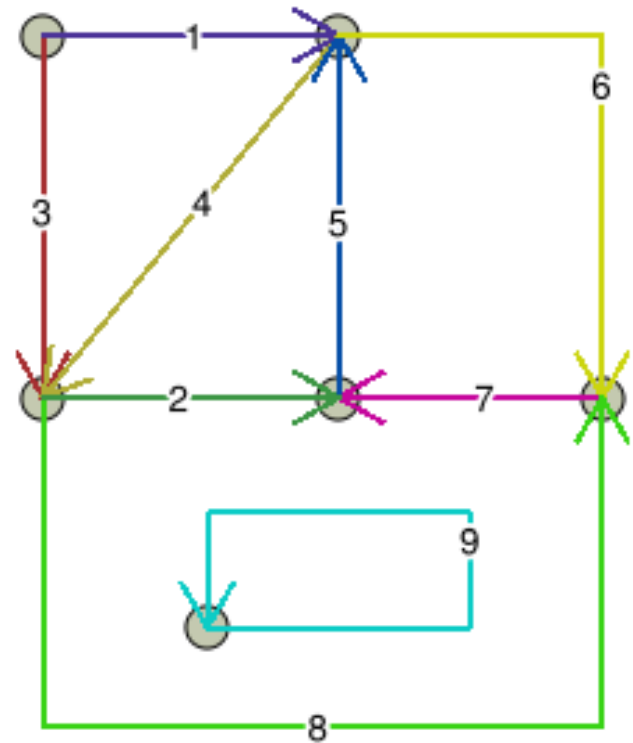
```
=> CREATE TABLE  
  conf.fl(id SERIAL  
  PRIMARY KEY);
```

```
=> SELECT  
AddTopoGeometryColumn(  
  'conf', 'conf', 'fl',  
  'g', 'LINE');
```



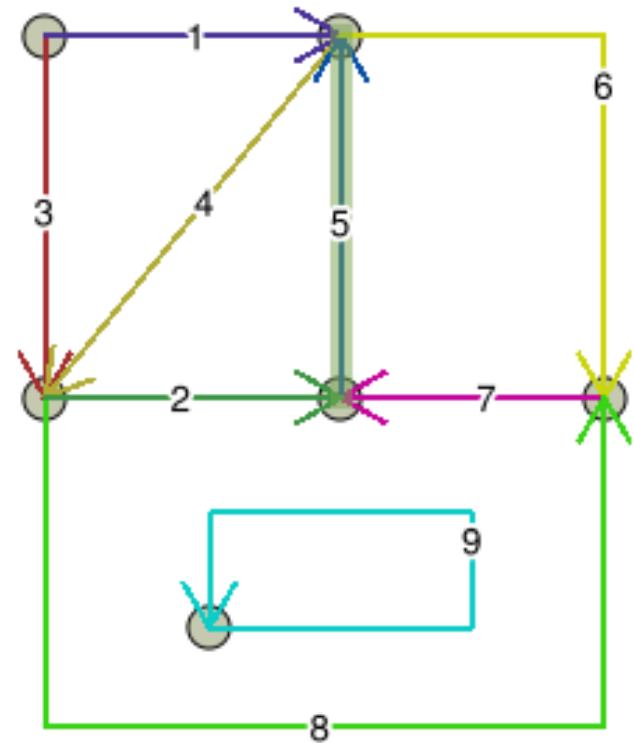
TopoGeometry: lineal

```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      2, -- type (lineal)
      2, -- layer id
      '{{5,2}}' -- edge 5
    )
  );
```



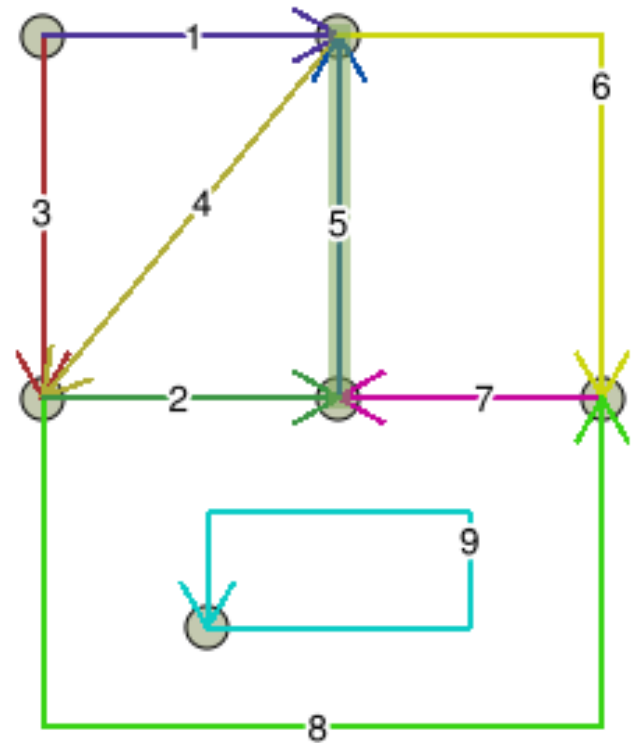
TopoGeometry: lineal

```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      2, -- type (lineal)
      2, -- layer id
      '{{5,2}}' -- edge 5
    )
  );
```



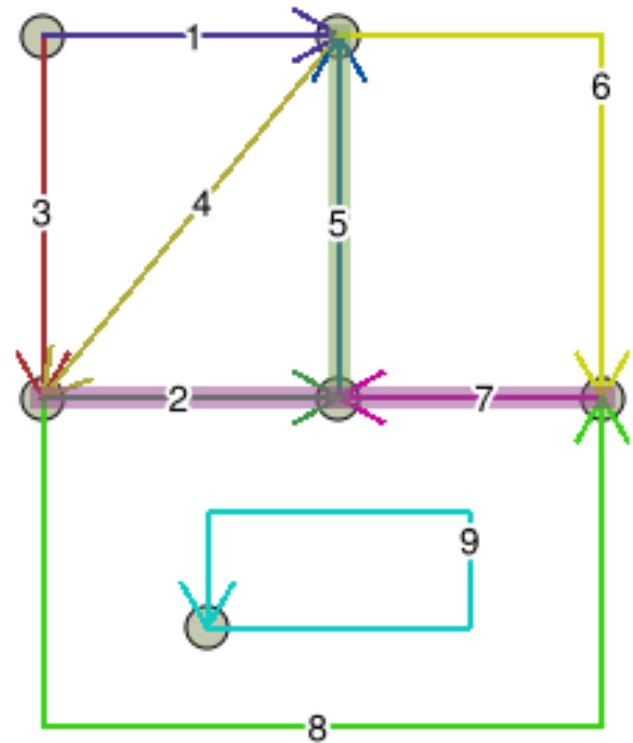
TopoGeometry: lineal

```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', 2, 2,
      -- edges 2 and 7
      '{{2,2},{7,2}}'
    )
  );
```



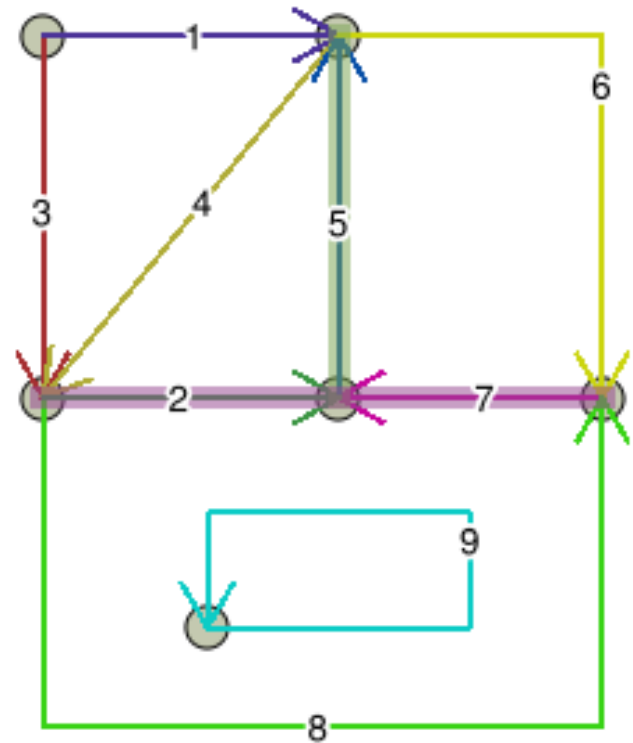
TopoGeometry: lineal

```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', 2, 2,
      -- edges 2 and 7
      '{{2,2},{7,2}}'
    )
  );
```



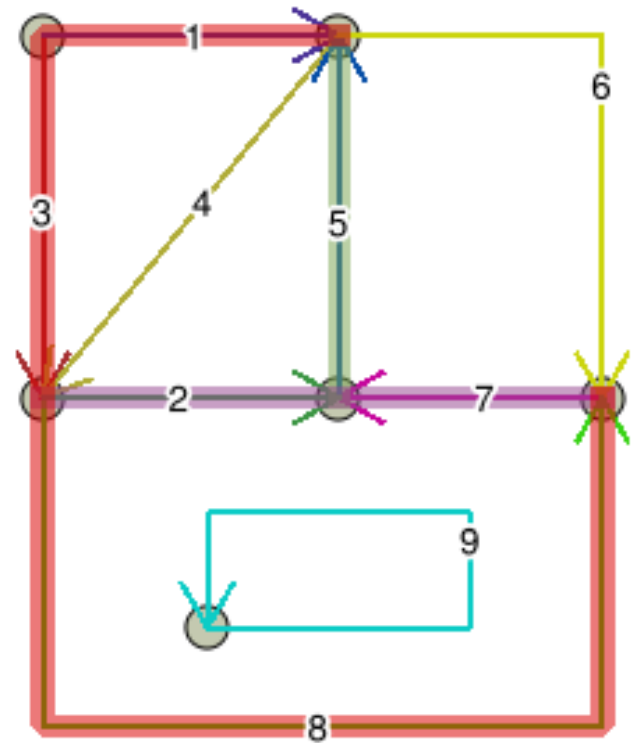
TopoGeometry: lineal

```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', 2, 2,
      -- edges 3, 1 and 8
      '{{3,2},{1,2},{8,2}}'
    )
  );
```



TopoGeometry: lineal

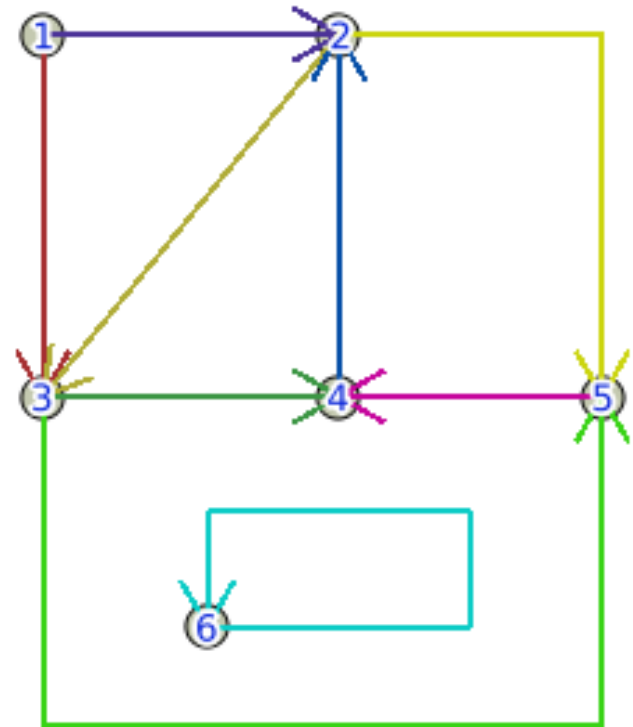
```
=> INSERT
  INTO conf.fl (g)
  VALUES (
    CreateTopoGeom(
      'conf', 2, 2,
      -- edges 3, 1 and 8
      '{{3,2},{1,2},{8,2}}'
    )
  );
```



TopoGeometry: puntal

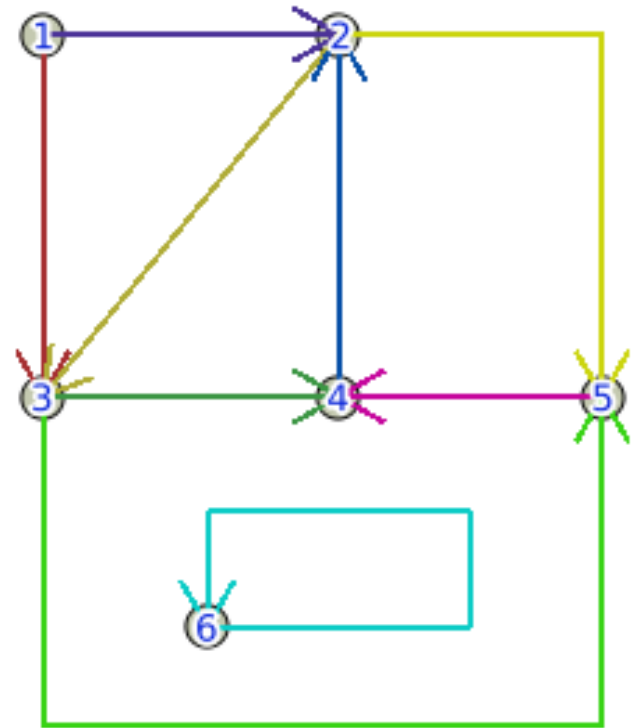
```
=> CREATE TABLE  
conf.fp(id SERIAL  
PRIMARY KEY);
```

```
=> SELECT  
AddTopoGeometryColumn(  
'conf', 'conf', 'fp',  
'g', 'POINT');
```



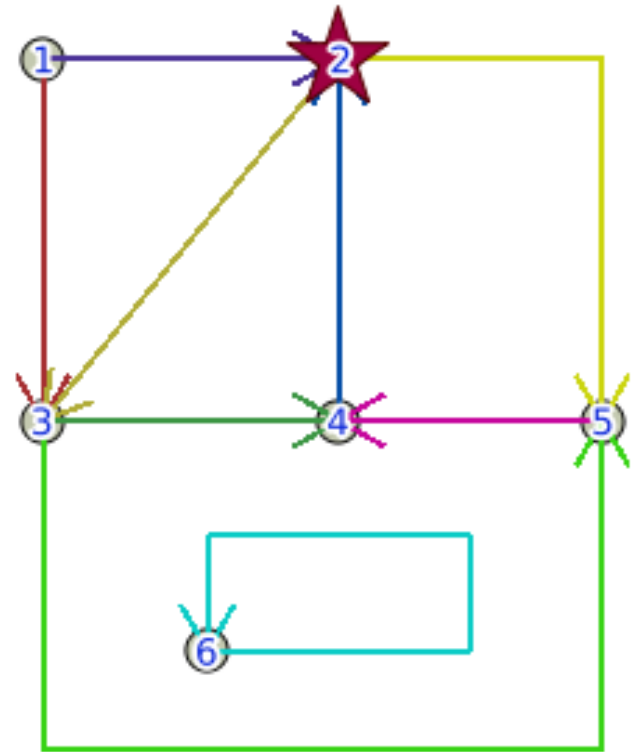
TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      1, -- type (puntal)
      3, -- layer id
      '{{2,1}}' -- node 2
    )
  );
```



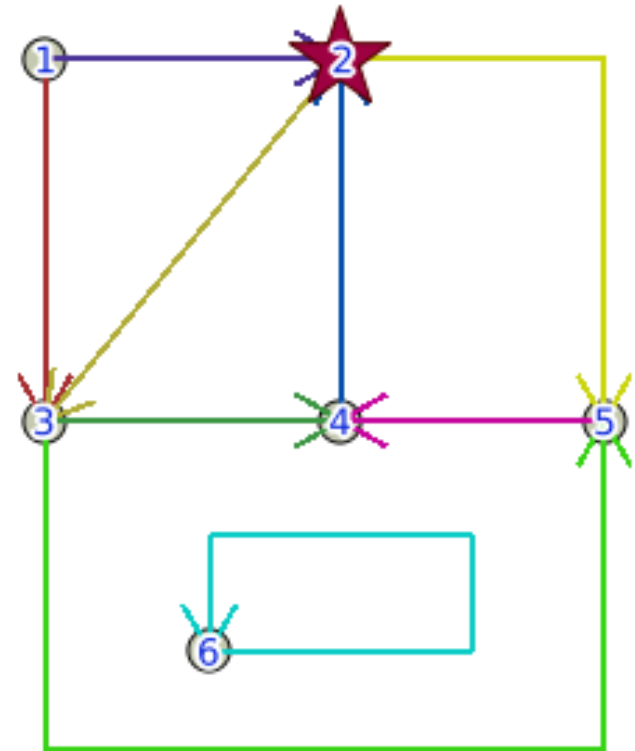
TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', -- Topo name
      1, -- type (puntal)
      3, -- layer id
      '{{2,1}}' -- node 2
    )
  );
```



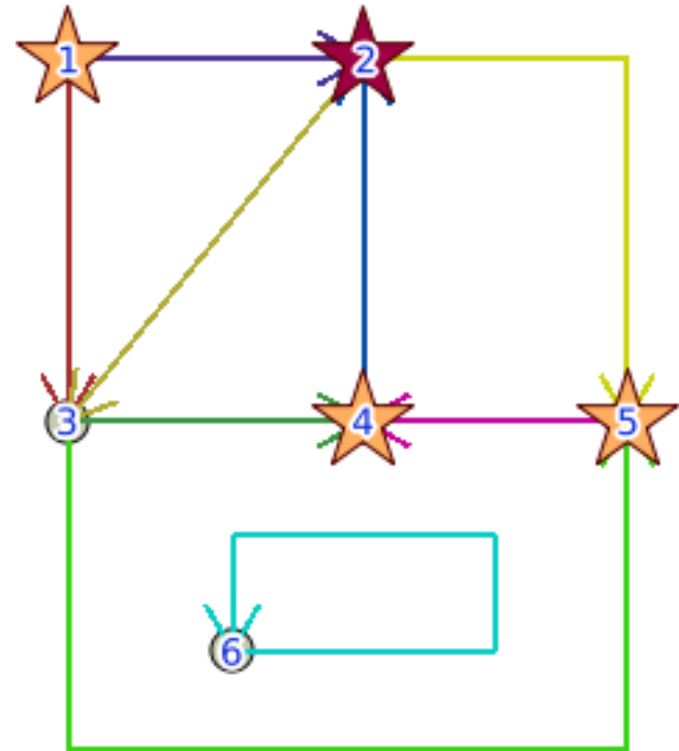
TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', 1, 3,
      -- nodes 1, 4 and 5
      '{{1,1},{4,1},{5,1}}'
    )
  );
```



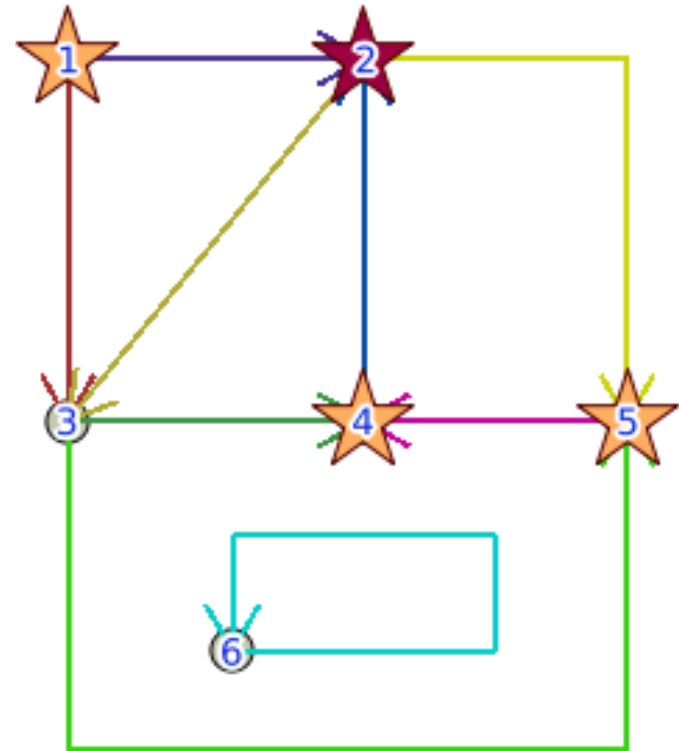
TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', 1, 3,
      -- nodes 1, 4 and 5
      '{{1,1},{4,1},{5,1}}'
    )
  );
```



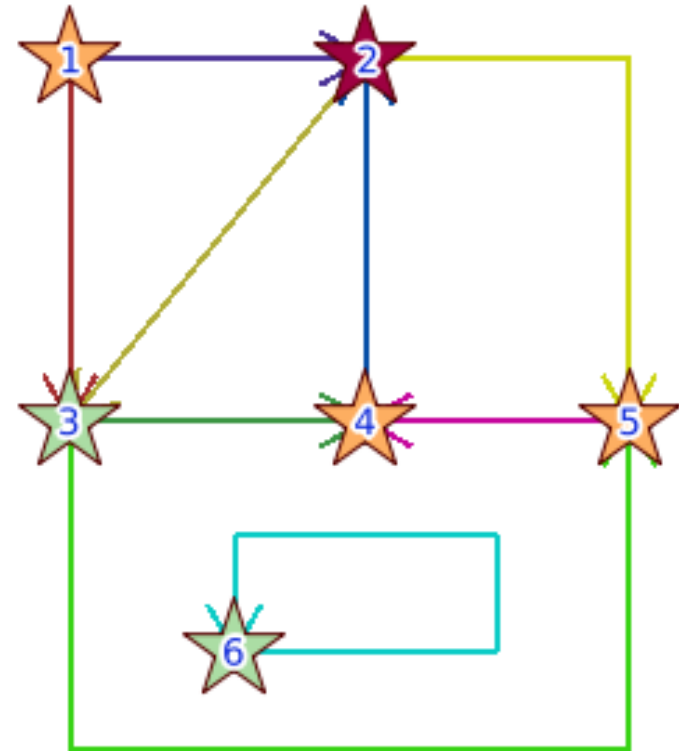
TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', 1, 3,
      -- nodes 3 and 6
      '{{3,1},{6,1}}'
    )
  );
```



TopoGeometry: puntal

```
=> INSERT
  INTO conf.fp (g)
  VALUES (
    CreateTopoGeom(
      'conf', 1, 3,
      -- nodes 3 and 6
      '{{3,1},{6,1}}'
    )
  );
```



TopoGeometry layers summary

```
=> SELECT  
TopologySummary('conf');
```

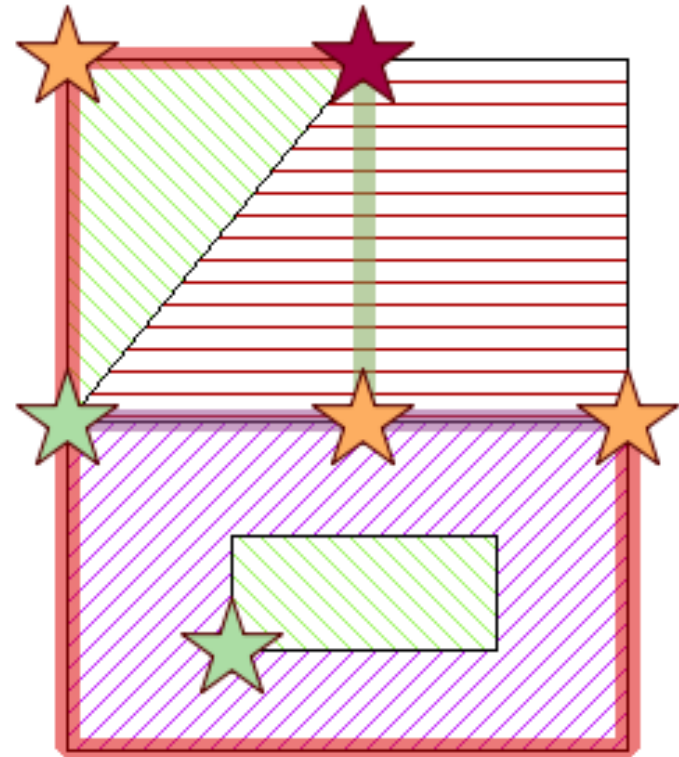
```
Topology conf (1),  
SRID -1, precision 0
```

```
6 nodes, 9 edges, 6 faces,  
9 topogeoms in 3 layers
```

```
Layer 1, type Polygonal (3),  
3 topogeoms Deploy: conf.fa.g
```

```
Layer 2, type Lineal (2),  
3 topogeoms Deploy: conf.fl.g
```

```
Layer 3, type Puntal (1),  
3 topogeoms Deploy: conf.fp.g
```



QGIS support

- Since QGIS version 1.8.0 (2012)
- TopoGeometry objects core edit support
- TopoViewer via DBManager
- Cleanup tools via PostGIS Topology editor plugin
https://github.com/strk/qgis_pgis_topoedit

GRASS support

- Since GRASS 7.0 (2013)
- Export to PostGIS Topology
- Import from PostGIS Topology

Question time

