Karlsruhe Institute of Technology

PostGIS
Data Integrity

Laboratory course

GEODETIC INSTITUTE, DEPARTMENT OF CIVIL ENGINEERING, GEO AND ENVIRONMENTAL SCIENCES, CHAIR IN GEOINFORMATICS

Content

- Basic information on data integrity according to OGC Simple Feature
- Rules of *simplicity* and *validity* according to the OGC
- Implementation of spatial data integrity in PostGIS
- Application examples
- Exercises

Geodetic Institute, Department Civil Engineering, Geo and Environmental Sciences, Chair in Geoinformatics

Data integrity according to OGC Simple Feature

- *Integrity/correctness* of spatial data plays important role in spatial data management and processing
- Depending on the application field and objectives of the data: different definition of when data are correct
- OGC uses a set of integrity rules that have proven as useful in broad application
  - Useful for *visual perception by users*
    - Geometries may not be drawn in such way that they appear visually ambiguous
  - Useful for *algorithmic processing*
    - If geometry shapes obey certain rules, then *simplified algorithms* with *better runtime behavior* in geometric processing can be used
    - Disadvantage: algorithms operate *faulty* if geometries are *not correct*

Geodetic Institute, Department Civil Engineering, Geo and Environmental Sciences, Chair in Geoinformatics

Data integrity according to OGC Simple Feature

OGC White Paper "Simple Features Specification" distinguishes between rules for **simplicity** and rules for **validity** as integrity criteria

- There is no fundamental difference between rules of simplicity and rules of validity, both define correctness of spatial data
- However, simplicity mostly refers to correctness of point and line geometries, validity mostly on correctness of polygons
- Only simple geometries may be used to create a valid polygon

Geodetic Institute, Department Civil Engineering, Geo and Environmental Sciences, Chair in Geoinformatics

Data integrity according to OGC Simple Feature

Rules of simplicity

- *Point geometry*

  "A point geometry is always simple."

  "A multi-point geometry is simple if there are no identical points (no points have identical coordinate values)."

- *Line geometry*

  "A linestring is simple if it does not intersect itself (except for endpoints, then it is linear ring)."

"A *multi-linestring* is simple if all its sub-linestrings are simple and there is *no intersection* between the linestrings, except at the end points of different linestrings."

Data integrity according to OGC Simple Feature

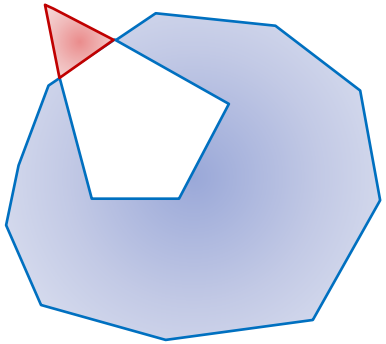
Rules of validity

- *Polygon geometry*

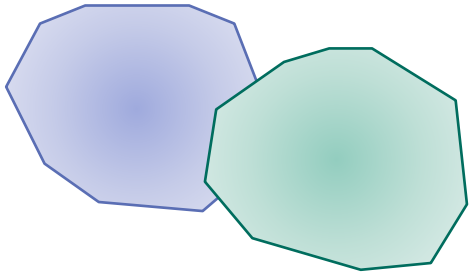
"A *polygon* is a simple object."

Examples of non-valid geometries

"A *polygon* is valid when its rings do not cross (intersect). However, the rings can intersect in one point. The inner rings must lie *entirely within* the outer ring."



"A *multi-polygon* is valid if all part polygons are valid and the interiors of the polygons do *not intersect each other*."



Data integrity according to OGC Simple Feature

- Objective of PostGIS: implementation of the requirements of spatial integrity of OGC specifications for Geo-DBMS
  - Implementation of OGC whitepaper "*Simple Features Specification for SQL*"
- Basically, it is possible to keep even *non-simple* and *invalid* geometries in PostGIS-DB
  - Advantages:
    - In some applications, such geometries are required
    - Algorithms for checking the integrity are expensive (in runtime); checks can be dispensed if known that the spatial data that has to be imported are already correct
  - Disadvantage:
    - Algorithms operate incorrectly if analysis functions (e.g. area calculation on polygons) are performed on incorrect spatial data
- PostGIS functions for testing simplicity and validity: `ST_isSimple()` and `ST_isValid()`

Geodetic Institute, Department Civil Engineering, Geo and Environmental Sciences, Chair in Geoinformatics

In practice: test on simplicity

```
SELECT ST_isSimple('LINESTRING(1 2, 3 4)');
```

```
 issimple
----------
 t
(1 row)
```

t = true

f = false

In practice: test on validity

```
SELECT ST_isValid('LINESTRING(0 0, 0 0)');
```

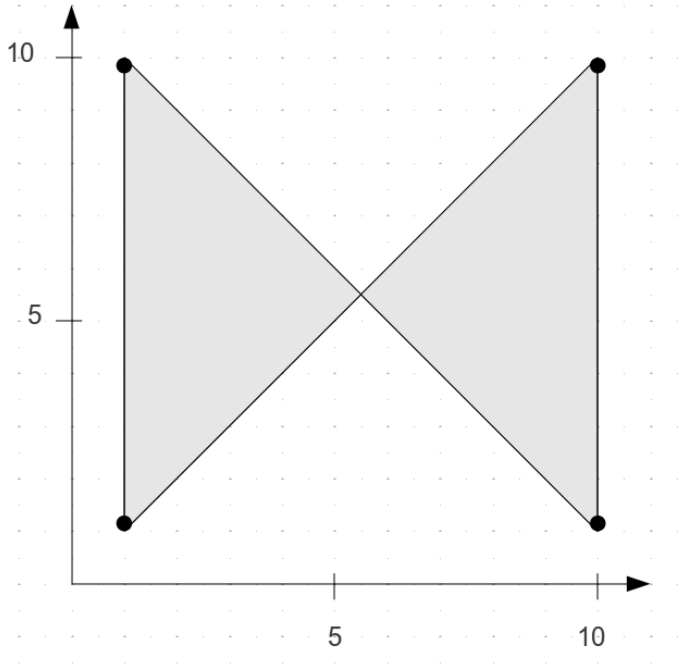
```
NOTICE:  Too few points in geometry
component at or near point 0 0
 isvalid
---------
 f
(1 row)
```

Exercise: Verify the geometries that have been inserted into the tables *streets* and *lakes* in the preceding tasks. Let all tuples (features) with *non-simple* geometries (from table *streets*) and *non-valid* geometries (from table *lakes*) be printed to the console.

Exercise: Delete all features with non-valid geometries from the tables.

Geodetic Institute, Department Civil Engineering, Geo and Environmental Sciences, Chair in Geoinformatics

Exercise: Create a table *polygons* that can include geometries of type *polygon* (with local coordinate system). Insert a new polygon with the coordinates *[0 0]*, *[10 10]*, *[10 0]* and *[0 10]* into the table. Let PostGIS check the validity of the geometry.



- Hint: Learn about permitted geometries in the whitepaper “OGC Simple Feature Specification for SQL”. Specifications on valid geometries are defined in the document (see illustrations on p. 23 ff.).

Exercise: The examples have shown that even non-valid geometries generally can be inserted into PostGIS tables. How can this be prevented? Alter the table *lakes* in a way that non-valid polygons may not be inserted any more.

- Hints:
  - You have to set a *constraint* on the table
  - Learn about *constraints* in the PostGIS documentation
  - Think about the issue, on which attribute of the relation the constraint has to be set