

Simple Features

in Geodatenbanken

Referent: Niko Kolaxidis
Seminar Geodatenbanken
Universität Heidelberg
19.05.2022



Einstieg in die Thematik

Was sind Simple Features?

Was sind Simple Features?

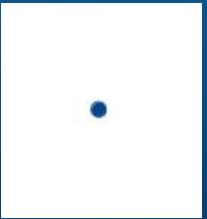
Feature =

Abstraktion von Objekten der realen Welt in digitale (Vektor-)daten

näher beschrieben in ISO 19107 des OpenGIS® Implementation Standard for Geographic information (OGC 2010 & 2011)

0 Dimensionen

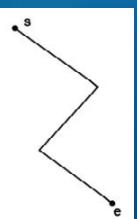
A



Geometry	
Point	
+	X(): Double
+	Y(): Double
+	Z(): Double
+	M(): Double

1 Dimension

B

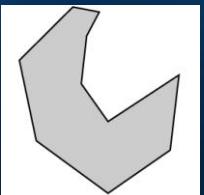


Geometry	
Curve	
+	length(): Double
+	startPoint(): Point
+	endPoint(): Point
+	isClosed(): Boolean
+	isRing(): Boolean

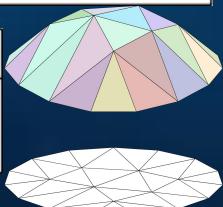
LineString	
+	numPoints(): Integer
+	pointN(Integer): Point

2 (bis 3) Dimensionen

C



Polygon	
+	exteriorRing(): LineString
+	numInteriorRing(): Integer
+	interiorRingN(N: Integer): LineString

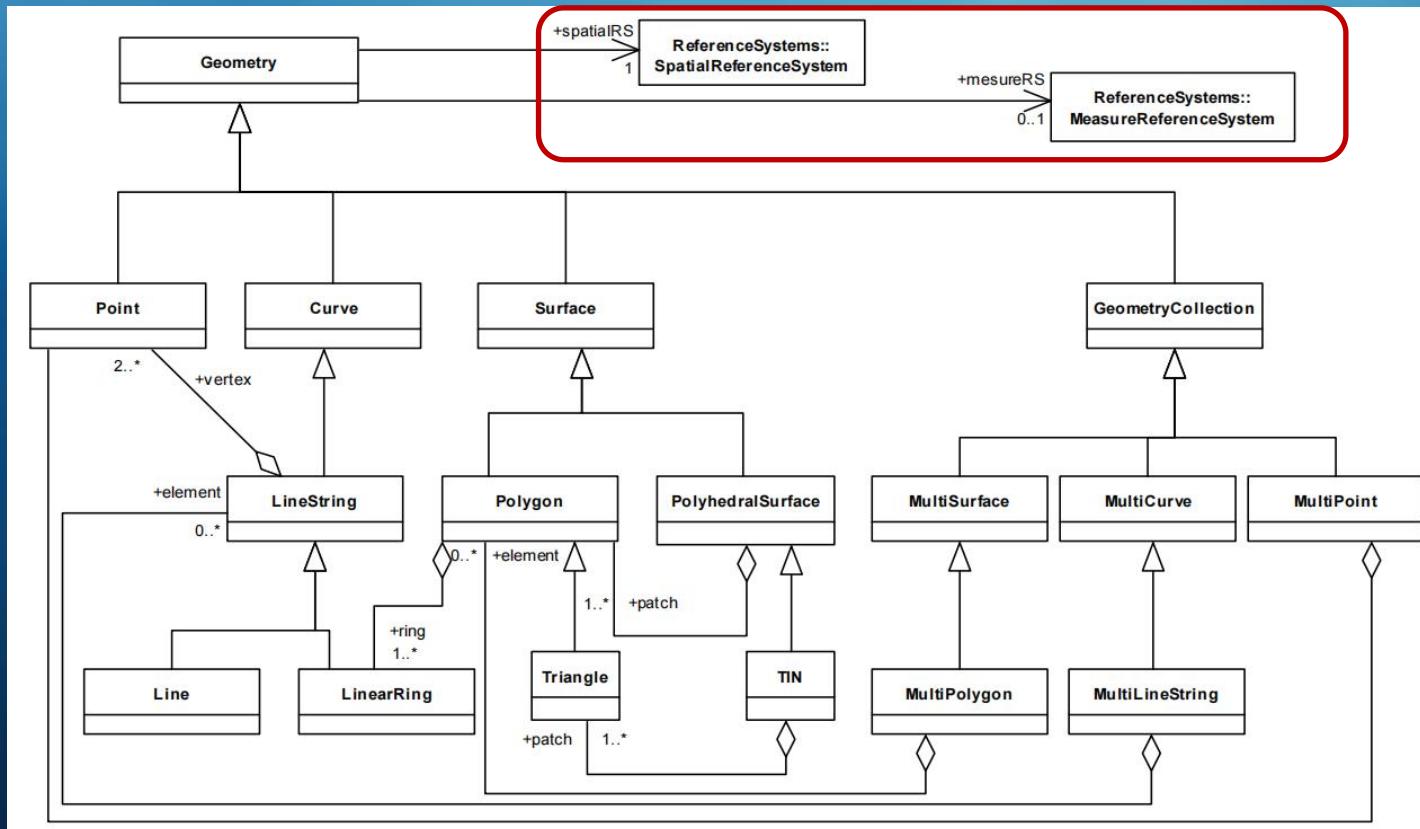


Geometry	
Surface	
+	area(): Area
+	centroid(): Point
+	pointOnSurface(): Point
+	boundary(): MultiCurve

PolyhedralSurface	
+	numPatches(): Integer
+	PatchN(N: Integer): Polygon
+	boundingPolygons(p: Polygon): MultiPolygon
+	isClosed(): Boolean

1

Geometry object model für Simple-Features





Integrierung in SQL

WKT vs. WKB

Erweiterungen

Repräsentation der Geometrie

Well-known Text (WKT)

Point(1,1)

LineString(2 2, 9 9)

Well-known Binary (WKB)

01010000000000000000F03F000000000000F03F

01 : Byte-Reihenfolge (Leserichtung)

01000000 : Geometriertyp (Punkt)

000000000000F03F : X

000000000000F03F : Y

WKT:

```
INSERT INTO geatable ( geom, name )
VALUES ( ST_GeomFromText('POINT(-126.4 45.32)', 312), 'A Place');
```

WKB:

```
INSERT INTO geatable ( geom, name )
VALUES ( ST_GeomFromWKB('\x01010000000000000000f03f000000000000f03f', 312), 'A Place');
```

Geometrische Objekte in PostgreSQL

Name	Description	Representation
point	Point on a plane	(x,y)
line	Infinite line	{A,B,C}
lseg	Finite line segment	((x1,y1),(x2,y2))
box	Rectangular box	((x1,y1),(x2,y2))
path	Closed path (similar to polygon)	((x1,y1),...)
path	Open path	[(x1,y1),...]
polygon	Polygon (similar to closed path)	((x1,y1),...)
circle	Circle	<(x,y),r> (center point and radius)

Geometrische Objekte in PostGIS

- POINT (1 2)
- POINT Z (1 2 3)
- POINT ZM (1 2 3 4)
- POINT M (1 2 4)
- LINESTRING (1 2, 3 4, 5 6)
- LINEARRING (0 0 0, 4 0 0, 4 4 0, 0 4 0, 0 0 0)
- POLYGON ((0 0 0,4 0 0,4 4 0,0,0 4 0,0 0 0),(1 1 0,2 1 0,2 2 0,1 2 0,1 1 0))
- TRIANGLE ((0 0, 0 9, 9 0, 0 0))
- TIN Z (((0 0 0, 0 0 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 0 0 0)))
- POLYHEDRALSURFACE Z (((0 0 0, 0 0 1, 0 1 1, 0 1 0, 0 0 0)), ((0 0 0, 0 1 0, 1 1 0, 1 0 0, 0 0 0)), ((0 0 0, 1 0 0, 1 0 1, 0 0 1, 0 0 0)), ((1 1 0, 1 1 1, 1 0 1, 1 0 0, 1 1 0)), ((0 1 0, 0 1 1, 1 1 1, 1 1 0, 0 1 0)), ((0 0 1, 1 0 1, 1 1 1, 0 1 1, 0 0 1)))
- MULTIPOINT ((0 0), (1 2))
- MULTILINESTRING ((0 0,1 1,1 2), (2 3,3 2,5 4))
- MULTIPOLYGON (((1 5, 5 5, 5 1, 1 1, 1 5)), ((6 5, 9 1, 6 1, 6 5)))
- GEOMETRYCOLLECTION (POINT(2 3), LINESTRING(2 3, 3 4))
- CIRCULARSTRING(0 0, 1 1, 1 0)
- CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0)
- COMPOUNDCURVE(CIRCULARSTRING(0 0, 1 1, 1 0),(1 0, 0 1))
- CURVEPOLYGON(CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0),(1 1, 3 3, 3 1, 1 1))
- CURVEPOLYGON(COMPOUNDCURVE(CIRCULARSTRING(0 0,2 0, 2 1, 2 3, 4 3),(4 3, 4 5, 1 4, 0 0)), CIRCULARSTRING(1.7 1, 1.4 0.4, 1.6 0.4, 1.6 0.5, 1.7 1))
- MULTICURVE((0 0, 5 5), CIRCULARSTRING(4 0, 4 4, 8 4))
- MULTISURFACE(CURVEPOLYGON(CIRCULARSTRING(0 0, 4 0, 4 4, 0 4, 0 0),(1 1, 3 3, 3 1, 1 1)), ((10 10, 14 12, 11 10, 10 10), (11 11, 11.5 11, 11 11.5, 11 11)))

Erweiterungen

- MySQL Spatial Extensions
- MonetDB/GIS extension für MonetDB
- PostGIS extension für PostgreSQL
- SpatiaLite extension für SQLite
- Oracle Spatial
- IBM DB2 Spatial Extender und IBM Informix Spatial DataBlade
- Microsoft SQL Server
- SAP Sybase IQ
- SAP HANA
- OGR für GDAL
- etc.

Geodatenbanken

- **Geometrische Datentypen**
 - z.B. Punkte, Linien, Polygone

- **Geometrische Operationen / Funktionen**
 - Schnittfläche / Intersection
 - Flächen- und Längenberechnung

- **Räumliche Datenstrukturen**
 - "räumlicher Index" (z.B. R-Baum)

- **Räumliche Abfragen**
 - Punktabfrage
 - Distanzabfrage
 - Regionsabfrage
 - Nächste-Nachbarn-Abfrage
 - ...

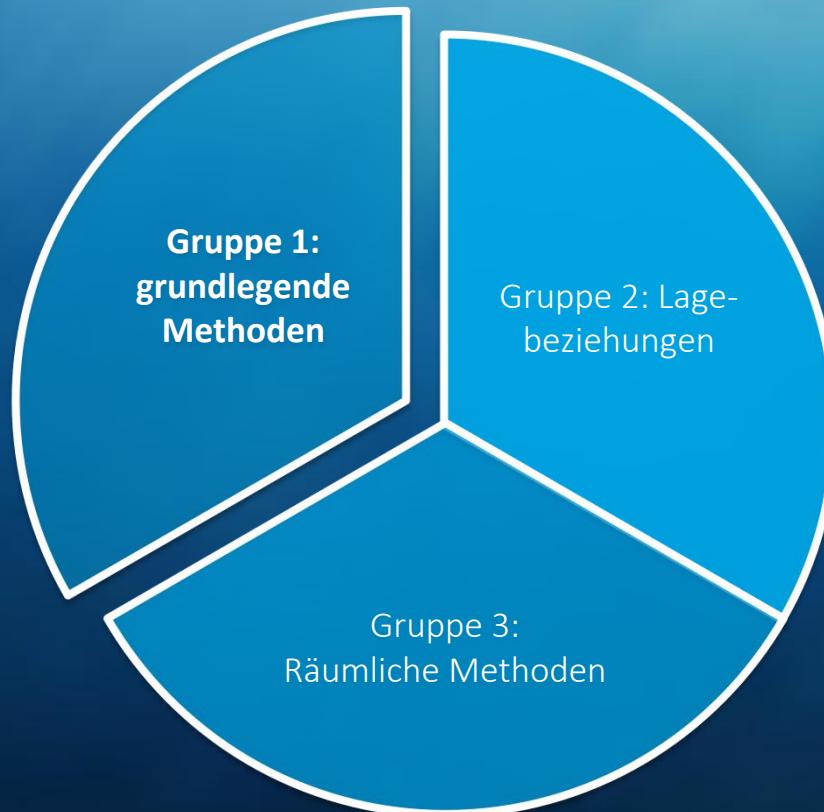
A large, stylized number '3' is centered within a circular frame. The frame is composed of numerous thin, light-colored lines that create a mesh or grid effect, resembling a wireframe circle.

Simple Features in SQL: PostGIS

Was ist alles mit Simple Features möglich?
Beispielanwendungen

3

Was mit Simple Features alles möglich ist - Teil 1



DBeaver 22.0.4 - <nkolaxidis> Script

File Edit Navigate Search SQL Editor Database Window Help

Database... Projects <nkolaxidis> Script

Enter a part of object name here

nkolaxidis - osmatrix.geog.uni-heidelberg
Databases nkolaxidis
Schemas public u2 sf22
Tables Views Materialized Views Indexes Functions Sequences Data types
Aggregate functions Event Triggers Extensions Storage System Info
Roles Administer System Info

CREATE SCHEMA IF NOT EXISTS sf22;
SET search_path='sf22', 'public';

CREATE TABLE IF NOT EXISTS geometries (name varchar, geom geometry);

INSERT INTO geometries VALUES
('Point', 'POINT(0 0)'),
('Linestring', 'LINESTRING(0 0, 1 1, 2 1, 2 2)'),
('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))'),
('PolygonWithHole', 'POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))'),
('Collection', 'GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))');

SELECT name, ST_AsText(geom) FROM geometries;

geometries 1 × Statistics 1 Output

SELECT name, ST_AsText(geom) FROM geometries

	name	st_astext
1	Point	POINT(0 0)
2	Linestring	LINESTRING(0 0, 1 1, 2 1, 2 2)
3	Polygon	POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))
4	PolygonWithHole	POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))
5	Collection	GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))

5 row(s) fetched - 176ms (1ms fetch), on May 16, 00:35:47

CET en Writable Smart Insert 13:46:537 Set: 0 | 0

DBeaver 22.0.4 - <nkolaxidis> Script.sql

File Edit Navigate Search SQL Editor Database Window Help

SQL Commit Rollback Auto nkolaxidis sf22@nkolaxidis

Database... Projects

Enter a part of object name here

nkolaxidis - osmmatrix.geog.uni-heidelberg Databases nkolaxidis Schemas public u2 sf22 Tables Views Materialized Views Indexes Functions Sequences Data types Aggregate functions Event Triggers Extensions Storage System Info Roles Administrister System Info

<nkolaxidis> Script.sql

```
SET search_path='sf22', 'public';

CREATE TABLE IF NOT EXISTS geometries (name varchar, geom geometry);

DELETE FROM geometries;

-- INSERT INTO geometries VALUES
('Point', 'POINT(0 0)'),
('Linestring', 'LINESTRING(0 0, 1 1, 2 1, 2 2)'),
('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))'),
('PolygonWithHole', 'POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))'),
('Collection', 'GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))');

-- --SELECT name, ST_AsText(geom) FROM geometries;
SELECT name, ST_AsBinary(geom) FROM geometries;
```

geometries 1

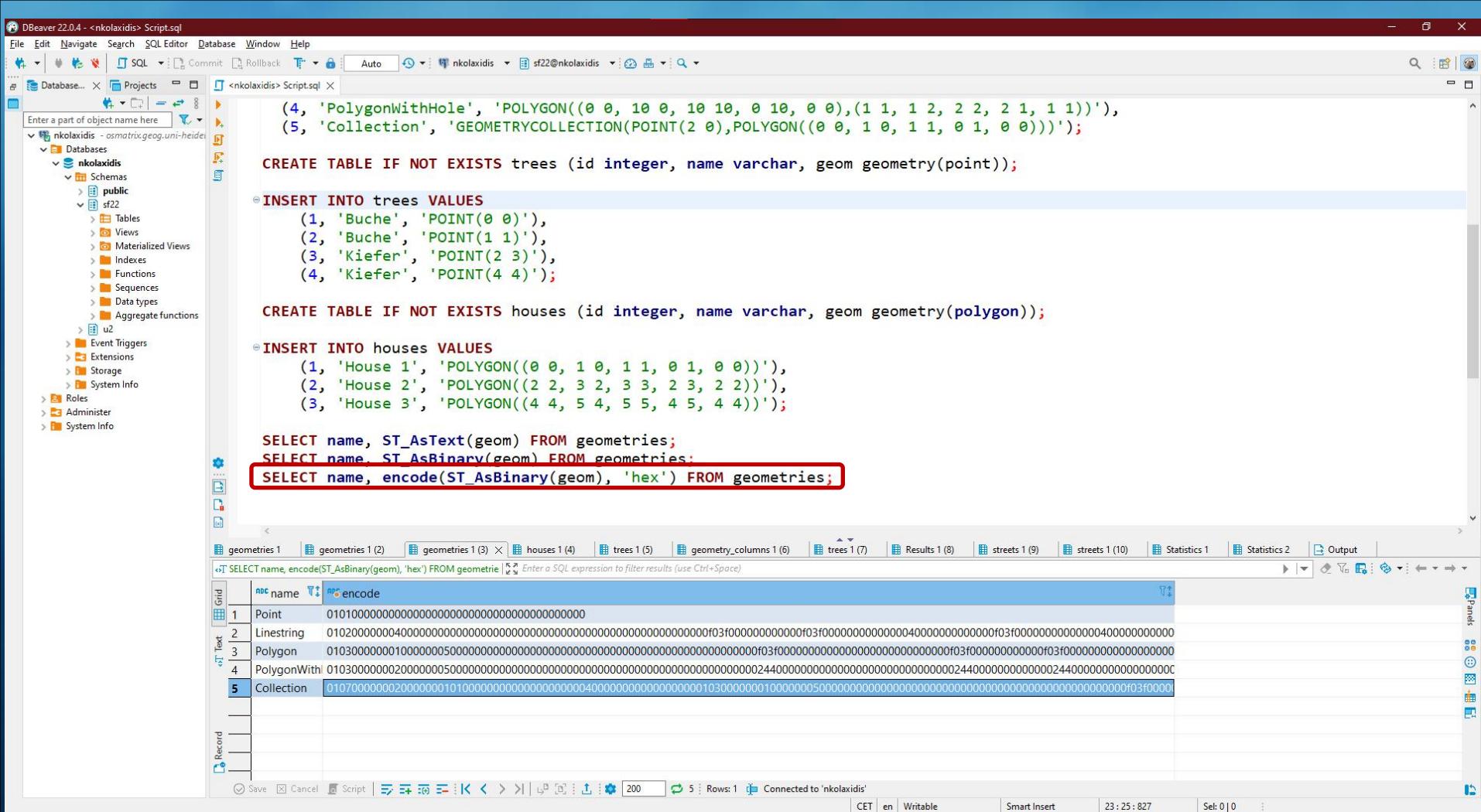
SELECT name, ST_AsBinary(geom) FROM geometries | Enter a SQL expression to filter results (use Ctrl+Space)

name	st_asbinary
1 Point	
2 Linestring	0... [73]
3 Polygon	... [93]
4 PolygonWithHole	... [177]
5 Collection	@ ... [123]

Value

00 01 02 03 04 05 06 07 08 09 0A 0B
0C:00 00 00 00 00 00 00 00 00 00 00 00
18:00 00 00 00 00 00 00 00 00 00 00 24
24:40 00 00 00 00 00 00 00 00 00 00 00
30:00 00 00 24 40 00 00 00 00 00 00 24
3C:40 00 00 00 00 00 00 00 00 00 00 00
48:00 00 00 24 40 00 00 00 00 00 00 00
54:00 00 00 00 00 00 00 00 00 00 00 00
60:00 00 00 00 00 00 F0 3F 00 00 00 00
6C:00 00 00 F0 3F 00 00 00 00 00 00 F0
78:3F 00 00 00 00 00 00 00 40 00 00 00 ?
84:00 00 00 40 00 00 00 00 00 00 00 00
90:40 00 00 00 00 00 00 40 00 00 00 00
9C:00 00 00 F0 3F 00 00 00 00 00 F0
A8:3F 00 00 00 00 00 00 F0 3F ? .?
B4:
C0:
CC:
D8:

Save Cancel Script | 200 | 5 | Rows: 1 | 5 row(s) fetched - 10ms (1ms fetch), on May 16, 01:39:24 | CET en Writable | Smart Insert | 16:48:615 | Sel: 0 | 0 |



DBeaver 22.0.4 - <nkolaxidis> Script.sql

File Edit Navigate Search SQL Editor Database Window Help

Database... Projects <nkolaxidis> Script.sql

Enter a part of object name here

nkolaxidis - osmatrix.geog.uni-heidelberg Databases nkolaxidis Schemas public sf22 Tables Views Materialized Views Indexes Functions Sequences Data types Aggregate functions u2 Event Triggers Extensions Storage System Info Roles Administrister System Info

CREATE TABLE IF NOT EXISTS trees (id integer, name varchar, geom geometry(point));

INSERT INTO trees VALUES
(1, 'Buche', 'POINT(0 0)'),
(2, 'Buche', 'POINT(1 1)'),
(3, 'Kiefer', 'POINT(2 3)'),
(4, 'Kiefer', 'POINT(4 4)');

CREATE TABLE IF NOT EXISTS houses (id integer, name varchar, geom geometry(polygon));

INSERT INTO houses VALUES
(1, 'House 1', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)'),
(2, 'House 2', 'POLYGON((2 2, 3 2, 3 3, 2 3, 2 2))'),
(3, 'House 3', 'POLYGON((4 4, 5 4, 5 5, 4 5, 4 4))');

SELECT name, ST_AsText(geom) FROM geometries;
SELECT name, ST_AsBinary(geom) FROM geometries;
SELECT name, encode(ST_AsBinary(geom), 'hex') FROM geometries;
SELECT name, ST_AsText(geom) FROM houses;
SELECT name, ST_AsText(geom) FROM trees;

SELECT name, ST_AsGeoJSON(geom) FROM trees;

geometries 1 (1) geometries 1 (2) geometries 1 (3) houses 1 (4) trees 1 (5) geometry_columns 1 (6) trees 1 (7) Results 1 (8) streets 1 (9) streets 1 (10) Statistics 1 Statistics 2 Output

SELECT name, ST_AsGeoJSON(geom) FROM trees | Enter a SQL expression to filter results (use Ctrl+Space)

name	st_asgeojson
1	Buche {"type": "Point", "coordinates": [0,0]}
2	Buche {"type": "Point", "coordinates": [1,1]}
3	Kiefer {"type": "Point", "coordinates": [2,3]}
4	Kiefer {"type": "Point", "coordinates": [4,4]}

Save Cancel Script | 200 4 4 row(s) fetched CET en Writable Smart Insert 23:25:827 Set: 0 | 0

DBeaver 22.0.4 - <nkolaxidis> Script.sql

File Edit Navigate Search SQL Editor Database Window Help

Database Navigator Projects <nkolaxidis> Script.sql

Enter a part of object name here

nkolaxidis - osmtris.geog.uni-heidelberg.de:5432

Databases nkolaxidis

Schemas public sf22

Tables geometries houses trees

Views Materialized Views

Indexes Functions Sequences Data types Aggregate functions

Event Triggers Extensions Storage System Info

Roles Administer System Info

geometries 1 houses 1 (2) trees 1 (3) geometry_columns 1 (4) Statistics 1 Output

```

('Linestring', 'LINESTRING(0 0, 1 1, 2 1, 2 2)'),
('Polygon', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))'),
('PolygonWithHole', 'POLYGON((0 0, 10 0, 10 10, 0 10, 0 0),(1 1, 1 2, 2 2, 2 1, 1 1))'),
('Collection', 'GEOMETRYCOLLECTION(POINT(2 0),POLYGON((0 0, 1 0, 1 1, 0 1, 0 0)))')

CREATE TABLE IF NOT EXISTS houses (name varchar, geom geometry(polygon));

INSERT INTO houses VALUES
    ('House 1', 'POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))'),
    ('House 2', 'POLYGON((2 2, 3 2, 3 3, 2 3, 2 2))'),
    ('House 3', 'POLYGON((4 4, 5 4, 5 5, 4 5, 4 4))');

CREATE TABLE IF NOT EXISTS trees (name varchar, geom geometry(point));

INSERT INTO trees VALUES
    ('Buche', 'POINT(0 0)'),
    ('Buche', 'POINT(1 1)'),
    ('Kiefer', 'POINT(2 3)'),
    ('Kiefer', 'POINT(4 4)');

SELECT name, ST_AsText(geom) FROM geometries;
--SELECT name, ST_AsBinary(geom) FROM geometries;
SELECT name, ST_AsText(geom) FROM houses;
SELECT name, ST_AsText(geom) FROM trees;
SELECT * FROM geometry_columns;

```

geometries 1 houses 1 (2) trees 1 (3) geometry_columns 1 (4) Statistics 1 Output

SELECT * FROM geometry_columns | Enter a SQL expression to filter results (use Ctrl+Space)

f_table_catalog	f_table_schema	f_table_name	f_geometry_column	coord_dimension	srid	type
nkolaxidis	sf22	geometries	geom	2	0	GEOMETRY
nkolaxidis	sf22	houses	geom	2	0	POLYGON
nkolaxidis	sf22	trees	geom	2	0	POINT

geometries 1 houses 1 (2) trees 1 (3) geometry_columns 1 (4) Statistics 1

SELECT name, ST_AsText(geom) FROM houses | Enter a SQL expression to filter results (use Ctrl+S)

name	st_astext
House 1	POLYGON((0 0, 1 0, 1 1, 0 1, 0 0))
House 2	POLYGON((2 2, 3 2, 3 3, 2 3, 2 2))
House 3	POLYGON((4 4, 5 4, 5 5, 4 5, 4 4))

geometries 1 houses 1 (2) trees 1 (3) geometry_columns 1 (4) Statistics 1

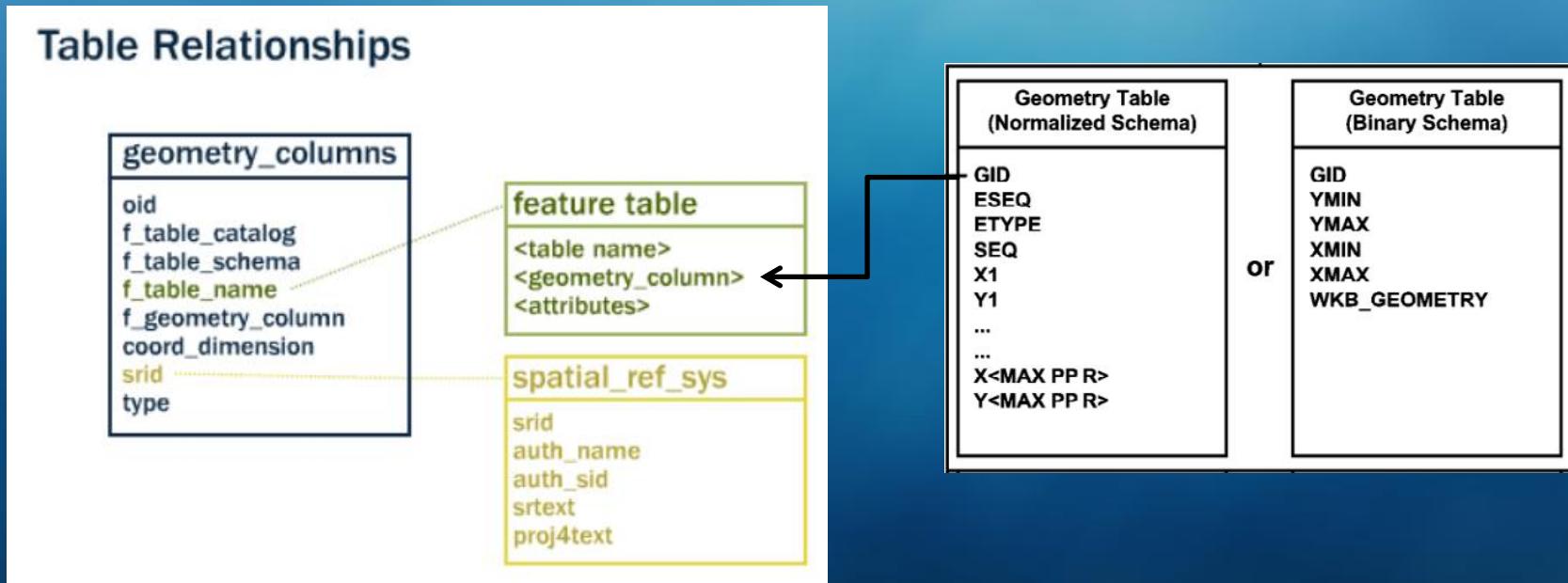
SELECT name, ST_AsText(geom) FROM trees | Enter a SQL expression to filter results (use Ctrl+S)

name	st_astext
Buche	POINT(0 0)
Buche	POINT(1 1)
Kiefer	POINT(2 3)
Kiefer	POINT(4 4)

geometries 1 houses 1 (2) trees 1 (3) geometry_columns 1 (4) Statistics 1

Rows: 1 3 row(s) fetched

Save Cancel Script | Back Forward | 200 | 3 | CET en Writable Smart Insert | 3:1:70 | Sel: 0 | 0 ...



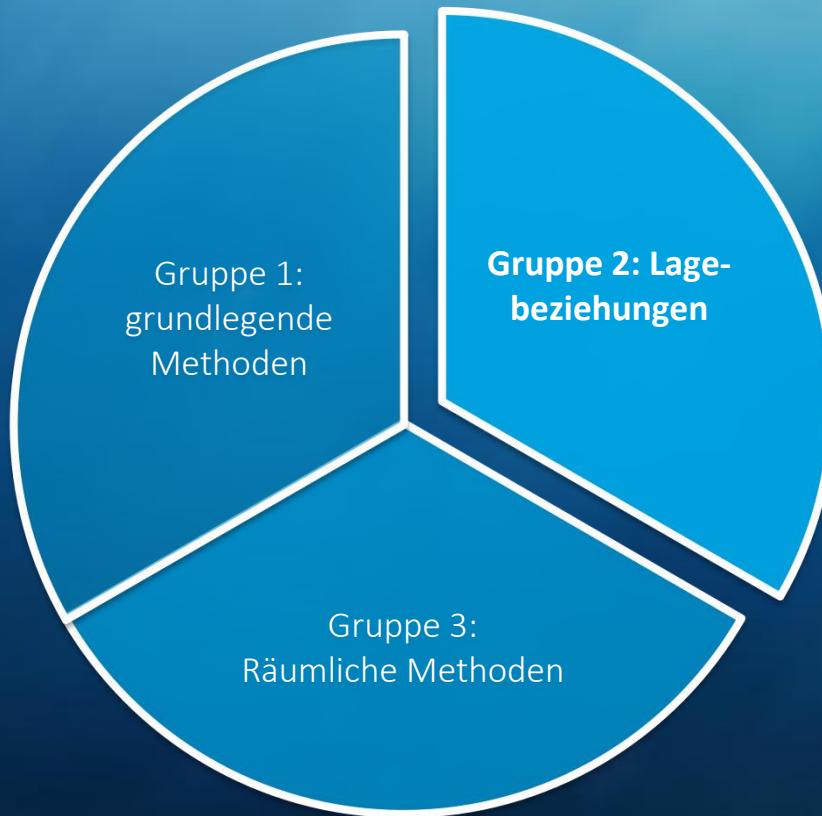
Grundlegende Methoden

SELECT Methode(Geometriespalte und weitere Attribute der Methode) FROM Tabelle;

- ST_Area: 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_AsText: Returns the Well-Known Text (WKT) representation of the geometry/geography without SRID metadata.
- ST_AsBinary: Returns the Well-Known Binary (WKB) representation of the geometry/geography without SRID meta data.
- ST_EndPoint: Returns the last point of a LINESTRING geometry as a POINT.
- ST_AsEWKB: Returns the Well-Known Binary (WKB) representation of the geometry with SRID meta data.
- ST_AsEWKT: Returns the Well-Known Text (WKT) representation of the geometry with SRID meta data.
- ST_AsGeoJSON: Returns the geometry as a GeoJSON element.
- ST_AsGML: Returns the geometry as a GML version 2 or 3 element.
- ST_AsKML: Returns the geometry as a KML element. Several variants. Default version=2, default precision=15.
- ST_AsSVG: Returns a Geometry in SVG path data given a geometry or geography object.
- ST_ExteriorRing: 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: Returns the 1-based Nth geometry if the geometry is a GEOMETRYCOLLECTION, MULTIPOINT, MULTILINESTRING, MULTICURVE or MULTIPOLYGON. Otherwise, return NULL.
- ST_GeomFromGML: Takes as input GML representation of geometry and outputs a PostGIS geometry object.
- ST_GeomFromKML: Takes as input KML representation of geometry and outputs a PostGIS geometry object
- ST_GeomFromText: Returns a specified ST_Geometry value from Well-Known Text representation (WKT).
- ST_GeomFromWKB: Creates a geometry instance from a Well-Known Binary geometry representation (WKB) and optional SRID.
- ST_GeometryType: Returns the geometry type of the ST_Geometry value.
- ST_InteriorRingN: Returns 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_Length: 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_NDims: Returns coordinate dimension of the geometry as a small int. Values are: 2,3 or 4.
- ST_NPoints: Returns the number of points (vertexes) in a geometry.
- ST_NRings: If the geometry is a polygon or multi-polygon returns the number of rings.
- ST_NumGeometries: If geometry is a GEOMETRYCOLLECTION (or MULTI*) returns the number of geometries, otherwise return NULL.
- ST_Perimeter: Returns the length measurement of the boundary of an ST_Surface or ST_MultiSurface value. (Polygon, Multipolygon)
- ST_SRID: Returns the spatial reference identifier for the ST_Geometry as defined in spatial_ref_sys table.
- ST_StartPoint: Returns the first point of a LINESTRING geometry as a POINT.
- ST_X: Returns the X coordinate of the point, or NULL if not available. Input must be a point.
- ST_Y: Returns the Y coordinate of the point, or NULL if not available. Input must be a point.

3

Was mit Simple Features alles möglich ist - Teil 2



3

Lagebeziehungen: Dimensionally Extended 9-Intersection Model (DE-9IM)

	Interior	Boundary	Exterior
Interior			
Boundary			
Exterior			

Annotations below the table:

- Interior: $\dim[I(a) \cap I(b)] = 2$
- Boundary: $\dim[B(a) \cap B(b)] = 1$
- Exterior: $\dim[E(a) \cap E(b)] = 2$
- Interior: $\dim[I(a) \cap B(b)] = 1$
- Boundary: $\dim[B(a) \cap B(b)] = 0$
- Exterior: $\dim[E(a) \cap B(b)] = 1$
- Interior: $\dim[I(a) \cap E(b)] = 2$
- Boundary: $\dim[B(a) \cap E(b)] = 1$
- Exterior: $\dim[E(a) \cap E(b)] = 2$

Equals A is the same as B	
Touches A touches B	
Overlaps A and B have multiple points in common	
Contains A contains B	
Disjoint A shares nothing with B	
Covers A covers B (or vice versa)	
Crosses A and B have at least one point in common	

3

Lagebeziehungen

- ST_Contains(geometry A, geometry B): 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_Crosses(geometry A, geometry B): Returns TRUE if the supplied geometries have some, but not all, interior points in common.
- ST_Disjoint(geometry A , geometry B): Returns TRUE if the Geometries do not “spatially intersect” - if they do not share any space together.
- ST_Distance(geometry A, geometry B): Returns the 2-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units.
- ST_DWithin(geometry A, geometry B, radius): Returns true if the geometries are within the specified distance (radius) of one another.
- ST_Equals(geometry A, geometry B): Returns true if the given geometries represent the same geometry. Directionality is ignored.
- ST_Intersects(geometry A, geometry B): Returns TRUE if the Geometries/Geography “spatially intersect” - (share any portion of space) and FALSE if they don't (they are Disjoint).
- ST_Overlaps(geometry A, geometry B): Returns TRUE if the Geometries share space, are of the same dimension, but are not completely contained by each other.
- ST_Touches(geometry A, geometry B): Returns TRUE if the geometries have at least one point in common, but their interiors do not intersect.
- ST_Within(geometry A , geometry B): Returns true if the geometry A is completely inside geometry B

The screenshot shows a database interface with two tables and a query result window.

Tables:

- geometries 1
- houses 1 (2)
- trees 1 (3)
- geometry_columns 1 (4)
- trees 1 (5)
- geometries 1
- houses 1 (2)
- trees 1 (3)
- geometry_columns 1 (4)

Query Result:

```
SELECT ST_Distance(a.geom, b.geom) FROM trees a, trees b WHERE a.id=1 AND b.id=2;
```

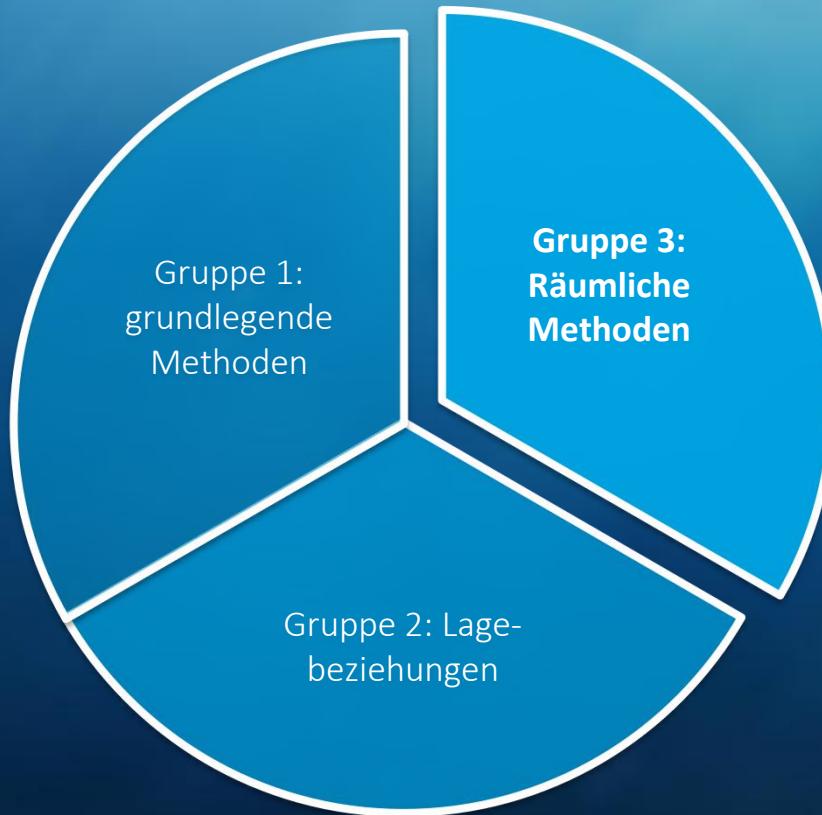
st_distance
1.4142135624

Table Data:

name	st_astext
Buche	POINT(0 0)
Buche	POINT(1 1)
Kiefer	POINT(2 3)
Kiefer	POINT(4 4)

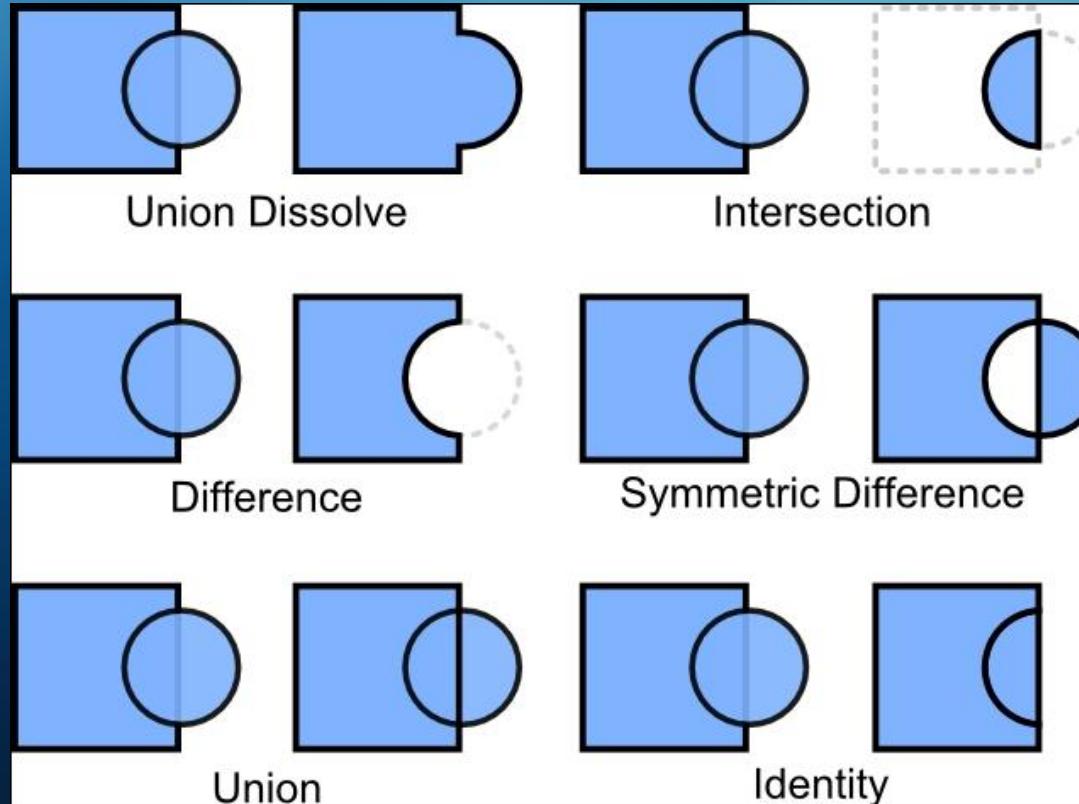
3

Was mit Simple Features alles möglich ist - Teil 3



3

Räumliche Methoden



3

Räumliche Methoden

- ST_Centroid(geometry): Returns a point geometry that represents the center of mass of the input geometry.
- ST_PointOnSurface(geometry): Returns a point geometry that is guaranteed to be in the interior of the input geometry.
- ST_Buffer(geometry, distance): 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.
- ST_Intersection(geometry A, geometry B): 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_Union(): Returns a geometry that represents the point set union of the Geometries.

-- Gruppe 3: Räumliche Analysen

```

CREATE TABLE IF NOT EXISTS streets (id integer, name varchar, geom geometry(linestring));

```

INSERT INTO streets VALUES
(1, 'Main Street', 'LINESTRING(0 0, 2 2)'),
(2, 'Second Street', 'LINESTRING(2 3, 5 4)'),
(3, 'Sub Street', 'LINESTRING(4 3, 5 4)');

```

SELECT name, ST_AsText(geom) FROM streets;
SELECT name, ST_Buffer(geom, 2) FROM streets;

```

name	st_astext
Main Street	LINESTRING(0 0, 2 2)
Second Street	LINESTRING(2 3, 5 4)
Sub Street	LINESTRING(4 3, 5 4)

name	st_buffer
Main Street	POLYGON ((0.5857864376269051 3.414213562373095, 0.8888595339607983 3.6629392246050925, 1.1429392246050925 4.14213562373095, 1.514213562373095 4.8888595339607983, 1.8888595339607983 5.6629392246050925, 2.257864376269051 5.8888595339607983, 2.6269051 6.14213562373095, 3.0000000000000003 6.414213562373095, 3.3739392246050925 6.8888595339607983, 3.747864376269051 7.257864376269051, 4.12179392246050925 7.6269051, 4.495723525314285 8.000000000000001, 4.869653168367074 8.3739392246050925, 5.243582811420863 8.747864376269051, 5.617512454473652 9.12179392246050925, 5.991442097526441 9.495723525314285, 6.36537174057923 9.869653168367074, 6.73930138363202 10.243582811420863, 7.113230926684813 10.617512454473652, 7.487160569737602 11.000000000000001, 7.861090212790391 11.3739392246050925, 8.23491985584318 11.747864376269051, 8.608849525314285 12.12179392246050925, 8.982779168367074 12.500000000000001, 9.356708811420863 12.8739392246050925, 9.730638454473652 13.243582811420863, 10.104568097526441 13.617512454473652, 10.47849774057923 14.000000000000001, 10.85242738363202 14.3739392246050925, 11.226357026684813 14.747864376269051, 11.600286669737602 15.12179392246050925, 11.974216312790391 15.500000000000001, 12.34814595584318 15.8739392246050925, 12.72207559889597 16.243582811420863, 13.09599524194876 16.617512454473652, 13.46992488495155 17.000000000000001, 13.84385452790434 17.3739392246050925, 14.21778417095713 17.747864376269051, 14.59171381391092 18.12179392246050925, 14.96564345696371 18.500000000000001, 15.3395730999165 18.8739392246050925, 15.71350274296929 19.243582811420863, 16.08743238592208 19.617512454473652, 16.46136202897487 20.000000000000001, 16.83529167192766 20.3739392246050925, 17.20922131498045 20.747864376269051, 17.58315095793324 21.12179392246050925, 17.95708059998603 21.500000000000001, 18.33099924293882 21.8739392246050925, 18.70492888599161 22.243582811420863, 19.0788585289444 22.617512454473652, 19.45278817199719 23.000000000000001, 19.82671781495098 23.3739392246050925, 20.20064745790377 23.747864376269051, 20.57457709995656 24.12179392246050925, 20.94850674291035 24.500000000000001, 21.32243638596314 24.8739392246050925, 21.69636592891593 25.243582811420863, 22.07029557196872 25.617512454473652, 22.44422521492151 26.000000000000001, 22.8181548579743 26.3739392246050925, 23.19208449992709 26.747864376269051, 23.56599914298088 27.12179392246050925, 23.940000000000001 27.500000000000001, 24.31400085701379 27.8739392246050925, 24.68799950002658 28.243582811420863, 25.06100014303937 28.617512454473652, 25.43400078605216 29.000000000000001, 25.80700042906495 29.3739392246050925, 26.180000000000001 29.747864376269051, 26.55300064296379 30.12179392246050925, 26.92600027592658 30.500000000000001, 27.29900090893937 30.8739392246050925, 27.67200054195216 31.243582811420863, 28.04500017496495 31.617512454473652, 28.41800080797774 32.000000000000001, 28.79100044099053 32.3739392246050925, 29.16400007395332 32.747864376269051, 29.53700070696611 33.12179392246050925, 29.9100003399789 33.500000000000001, 30.28300097299169 33.8739392246050925, 30.65600060595448 34.243582811420863, 31.02900023896727 34.617512454473652, 31.40200087198006 35.000000000000001, 31.77500050499285 35.3739392246050925, 32.14800013795564 35.747864376269051, 32.52100077096843 36.12179392246050925, 32.89400040398122 36.500000000000001, 33.26700003699391 36.8739392246050925, 33.6400006699567 37.243582811420863, 34.01300030291949 37.617512454473652, 34.38600093593228 38.000000000000001, 34.75900056894507 38.3739392246050925, 35.13200020195786 38.747864376269051, 35.50500083497065 39.12179392246050925, 35.87800046798344 39.500000000000001, 36.25100009999623 39.8739392246050925, 36.62400073290902 40.243582811420863, 37.09700036592181 40.617512454473652, 37.470000000000001 41.000000000000001, 37.8430006329346 41.3739392246050925, 38.21600029994739 41.747864376269051, 38.58900096696018 42.12179392246050925, 38.96200063397297 42.500000000000001, 39.33500030098576 42.8739392246050925, 39.70800096799855 43.243582811420863, 40.08100063491134 43.617512454473652, 40.45400030192413 44.000000000000001, 40.82700096893692 44.3739392246050925, 41.20000063594971 44.747864376269051, 41.5730003029625 45.12179392246050925, 41.94600096997529 45.500000000000001, 42.31900063698808 45.8739392246050925, 42.69200030390087 46.243582811420863, 43.06500097091366 46.617512454473652, 43.43800063792645 47.000000000000001, 43.81100030493924 47.3739392246050925, 44.18400097195203 47.747864376269051, 44.55700063896482 48.12179392246050925, 44.93000030597761 48.500000000000001, 45.3030009729904 48.8739392246050925, 45.67300063995319 49.243582811420863, 46.04600030696598 49.617512454473652, 46.41900097397877 50.000000000000001, 46.79200064099156 50.3739392246050925, 47.16500030790435 50.747864376269051, 47.53800097491714 51.12179392246050925, 47.91100064193093 51.500000000000001, 48.28400030894372 51.8739392246050925, 48.65700097595651 52.243582811420863, 49.0300006429693 52.617512454473652, 49.4030003109821 53.000000000000001, 49.77600097799489 53.3739392246050925, 50.14900064490768 53.747864376269051, 50.52200031192047 54.12179392246050925, 50.89500097893326 54.500000000000001, 51.26800064594605 54.8739392246050925, 51.64100031295884 55.243582811420863, 52.01400097997163 55.617512454473652, 52.38700064698442 56.000000000000001, 52.76000031399741 56.3739392246050925, 53.1330009839102 56.747864376269051, 53.50600065092309 57.12179392246050925, 53.87900032093598 57.500000000000001, 54.25200098794877 57.8739392246050925, 54.62500065496156 58.243582811420863, 55.00800031797435 58.617512454473652, 55.38100098498714 59.000000000000001, 55.75400065199993 59.3739392246050925, 56.12700032001272 59.747864376269051, 56.50000098602551 60.12179392246050925, 56.8730006560383 60.500000000000001, 57.24600032305119 60.8739392246050925, 57.61900098906408 61.243582811420863, 58.00200065907687 61.617512454473652, 58.37500032608966 62.000000000000001, 58.74800099200245 62.3739392246050925, 59.12100066201524 62.747864376269051, 59.50000033002803 63.12179392246050925, 59.87300099704082 63.500000000000001, 60.24600066405361 63.8739392246050925, 60.6190003310664 64.243582811420863, 61.0020009970792 64.617512454473652, 61.37500066409209 65.000000000000001, 61.74800033110488 65.3739392246050925, 62.12100099911767 65.747864376269051, 62.50000066113046 66.12179392246050925, 62.87300033114325 66.500000000000001, 63.24600099915604 66.8739392246050925, 63.61900066116883 67.243582811420863, 64.00200033118162 67.617512454473652, 64.37500099919441 68.000000000000001, 64.7480006612072 68.3739392246050925, 65.12100033122009 68.747864376269051, 65.50000099923288 69.12179392246050925, 65.87300066124567 69.500000000000001, 66.24600033125846 69.8739392246050925, 66.61900099927125 70.243582811420863, 67.00200066128504 70.617512454473652, 67.37500033130783 71.000000000000001, 67.74800099932062 71.3739392246050925, 68.12100066133341 71.747864376269051, 68.5000003313462 72.12179392246050925, 68.87300099935909 72.500000000000001, 69.24600066137188 72.8739392246050925, 69.61900033138467 73.243582811420863, 69.99200099940746 73.617512454473652, 70.36500066142025 74.000000000000001, 70.73800033143304 74.3739392246050925, 71.11100099944583 74.747864376269051, 71.48400066145862 75.12179392246050925, 71.85700033147141 75.500000000000001, 72.2300009994842 75.8739392246050925, 72.60300066149709 76.243582811420863, 72.97600033151088 76.617512454473652, 73.34900099952367 77.000000000000001, 73.72100066153666 77.3739392246050925, 74.09400033154945 77.747864376269051, 74.46700099956224 78.12179392246050925, 74.84000066157503 78.500000000000001, 75.21300033158782 78.8739392246050925, 75.58600099959961 79.243582811420863, 75.9590006616124 79.617512454473652, 76.3320003316252 80.000000000000001, 76.70500099963709 80.3739392246050925, 77.07800066164988 80.747864376269051, 77.45100033166267 81.12179392246050925, 77.82400099967446 81.500000000000001, 78.19700066168725 81.8739392246050925, 78.57000033169904 82.243582811420863, 78.94300099971183 82.617512454473652, 79.31600066172462 83.000000000000001, 79.68900033173641 83.3739392246050925, 80.0620009997482 83.747864376269051, 80.43500066176109 84.12179392246050925, 80.80800033177388 84.500000000000001, 81.18100099978567 84.8739392246050925, 81.55400066179846 85.243582811420863, 81.92700033181125 85.617512454473652, 82.29900099982304 86.000000000000001, 82.67200066183583 86.3739392246050925, 83.04500033184762 86.747864376269051, 83.41800099985941 87.12179392246050925, 83.7910006618712 87.500000000000001, 84.16400033188309 87.8739392246050925, 84.53700099989488 88.243582811420863, 84.91000066190667 88.617512454473652, 85.28300033191846 89.000000000000001, 85.65600099993025 89.3739392246050925, 86.02900066194204 89.747864376269051, 86.40200033195383 90.12179392246050925, 86.77500099996562 90.500000000000001, 87.14800066197741 90.8739392246050925, 87.5210003319892 91.243582811420863, 87.89400099999999 91.617512454473652, 88.26700066201178 92.000000000000001, 88.64000033202357 92.3739392246050925, 89.01300099999998 92.747864376269051, 89.38600066203537 93.12179392246050925, 89.75900033204716 93.500000000000001, 90.13200099999997 93.8739392246050925, 90.50500066205896 94.243582811420863, 90.87800033207075 94.617512454473652, 91.25100099999996 95.000000000000001, 91.62400066208255 95.3739392246050925, 92.00700033209434 95.747864376269051, 92.38000099999995 96.12179392246050925, 92.75300066210614 96.500000000000001, 93.12600033211793 96.8739392246050925, 93.50000099999996 97.243582811420863, 93.87300066212975 97.617512454473652, 94.24600033214154 98.000000000000001, 94.61900099999997 98.3739392246050925, 95.00000066215336 98.747864376269051, 95.37300033216515 99.12179392246050925, 95.74600099999998 99.500000000000001, 96.11900066217697 99.8739392246050925, 96.50000033218876 100.000000000000001, 96.87300099999999 100.3739392246050925, 97.24600066219978 100.747864376269051, 97.61900033221157 101.12179392246050925, 98.00000099999998 101.500000000000001, 98.37300066222337 101.8739392246050925, 98.74600033223516 102.243582811420863, 99.11900099999999 102.617512454473652, 99.50000066224698 103.000000000000001, 99.87300033225877 103.3739392246050925, 100.24600099999998 103.747864376269051, 100.61900066227057 104.12179392246050925, 101.00000033228236 104.500000000000001, 101.37300099999999 104.8739392246050925, 101.74600066229418 105.243582811420863, 102.11900033230597 105.617512454473652, 102.50000099999998 106.000000000000001, 102.87300066231776 106.3739392246050925, 103.24600033232955 106.747864376269051, 103.61900099999999 107.12179392246050925, 104.00000066234134 107.500000000000001, 104.37300033235313 107.8739392246050925, 104.74600099999998 108.243582811420863, 105.11900066236492 108.617512454473652, 105.50000033237671 109.000000000000001, 105.87300099999999 109.3739392246050925, 106.2460006623885 109.747864376269051, 106.61900033239929 110.12179392246050925, 107.00000099999998 110.500000000000001, 107.37300066241108 110.8739392246050925, 107.74600033242287 111.243582811420863, 108.11900099999999 111.617512454473652, 108.50000066243468 112.000000000000001, 108.87300033244647 112.3739392246050925, 109.24600099999998 112.747864376269051, 109.61900066245826 113.12179392246050925, 110.00000033247005 113.500000000000001, 110.37300099999999 113.8739392246050925, 110.74600066248184 114.243582811420863, 111.11900033249363 114.617512454473652, 111.50000099999998 115.000000000000001, 111.87300066250542 115.3739392246050925, 112.24600033251721 115.747864376269051, 112.61900099999999 116.12179392246050925, 113.00000066252898 116.500000000000001, 113.37300033254077 116.8739392246050925, 113.74600099999998 117.243582811420863, 114.11900066255256 1

Simple Features

Vielen Dank für die
Aufmerksamkeit!





Diskussion/Take-Home Messages



Diskussion/Take-Home Messages

1. Was sind die drei grundlegenden Feature-Klassen laut OpenGIS SFA und was zeichnet Simple Features aus?
2. Welche zwei Geometrie-Formate werden im OpenGIS SFA definiert und fällt dir noch ein anderes Geoformat ein?
3. Welche drei Gruppen von Methoden gibt es für Simple Features?
Nenne je zwei für jede Gruppe.
4. Wie werden räumliche Methoden & Funktionen aufgerufen (Grundstruktur)?



Literaturnachweise

- Alexandrov, O. (2007): A piecewise linear function over two dimensions (top) and the polygonal areas on which it is linear (bottom). - URL: https://en.wikipedia.org/wiki/Polyhedral_terrain#/media/File:Piecewise_linear_function2D.svg [12.05.2022].
- Auer, M. (2022): Seminar Geodatenbanken. Allgemeine Einführung. Foliensatz, Universität Heidelberg.
- OGC (2010): OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 2: SQL option.
- OGC (2011): OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture.
- OGC (2018): Features and geometry – Part 1: Feature models.
- Piorkowski, A. (2011): MySQL Spatial and PostGIS. Implementations of spatial data standards. In: Electronic Journal of Polish Agricultural Universities, 14(1), #3.
- The PostGIS Development Group (2022): Chapter 4. Data Management. PostGIS 3.2.2dev Manual. - URL: https://postgis.net/docs/using_postgis_dbmanagement.html [12.05.2022].
- The PostgreSQL Global Development Group (2022): PostgreSQL 14.3 Documentation.
- QGIS (2020): Simple-Feature-Modell. - URL: https://docs.qgis.org/3.22/de/docs/training_manual/spatial_databases/simple_feature_model.html [12.05.2022].
- Ramsey, P./Leslie, M. (2012): Introduction to PostGIS. - URL: <https://postgis.net/workshops/postgis-intro/index.html> [12.05.2022].
- Shekhar, S./Xiong, H. (2008): Encyclopedia of GIS. SpringerScience.
- Stolze, K. (o.J.): SQL/MM Spatial. The Standard to Manage Spatial Data in Relational Database Systems. Universität Jena.

Simple Features

Vielen Dank für die
Aufmerksamkeit!

Referent: Niko Kolaxidis

1

Standards für Simple Features in SQL

OpenGIS SFA - Part 1: Common architecture

OpenGIS SFA - Part 2: SQL Option

SQL/MM - Part 3: Spatial

Simple Features

- GOM SF
- Speicherung
- Erstellung
- Datenaufruf
- Datenupdates
- Lagebeziehungen

Funktionen & Befehle für raumbezogene Daten

- OpenGIS SFA 2.0
- GOM Zirkuläre Features
- erweiterte Funktionen/Methoden
- GML-Support

Raumbezogene Daten in SQL

2

Geometrietypen in WKB

Type	Code	Type	Code	Type	Code	Type	Code
Geometry	0	Geometry Z	1000	Geometry M	2000	Geometry ZM	3000
Point	1	Point Z	1001	Point M	2001	Point ZM	3001
LineString	2	LineString Z	1002	LineString M	2002	LineString ZM	3002
Polygon	3	Polygon Z	1003	Polygon M	2003	Polygon ZM	3003
MultiPoint	4	MultiPoint Z	1004	MultiPoint M	2004	MultiPoint ZM	3004
MultiLineString	5	MultiLineString Z	1005	MultiLineString M	2005	MultiLineString ZM	3005
MultiPolygon	6	MultiPolygon Z	1006	MultiPolygon M	2006	MultiPolygon ZM	3006
GeometryCollection	7	GeometryCollection Z	1007	GeometryCollection M	2007	GeometryCollection ZM	3007
CircularString	8	CircularString Z	1008	CircularString M	2008	CircularString ZM	3008
CompoundCurve	9	CompoundCurve Z	1009	CompoundCurve M	2009	CompoundCurve ZM	3009
CurvePolygon	10	CurvePolygon Z	1010	CurvePolygon M	2010	CurvePolygon ZM	3010
MultiCurve	11	MultiCurve Z	1011	MultiCurve M	2011	MultiCurve ZM	3011
MultiSurface	12	MultiSurface Z	1012	MultiSurface M	2012	MultiSurface ZM	3012
Curve	13	Curve Z	1013	Curve M	2013	Curve ZM	3013
Surface	14	Surface Z	1014	Surface M	2014	Surface ZM	3014
PolyhedralSurface	15	PolyhedralSurface Z	1015	PolyhedralSurface M	2015	PolyhedralSurface ZM	3015
TIN	16	TIN Z	1016	TIN M	2016	TIN ZM	3016

Repräsentation der Geometrie

→ Nicht darauf verlassen!

Extended Well-known Text (EWKT)

Für 3D Geometrien wird Z per Default erkannt:

OGC: POINT Z (1 2 3)
EWKT: POINT (1 2 3)

Extended Well-known Binary (EWKB)

0101000020E61000007EAA0A0DC4920DC05AF0A2AF201F4440

OGC: POINT M (1 2 3)
EWKT: POINTM (1 2 3)

Für 4D (3D + M) Geometrien werden Z und M per Default erkannt:

OGC: POINT ZM (1 2 3 4)
EWKT: POINT (1 2 3 4)

Einbezug des CRS (SRID):

SRID=32632;POINT(0 0)