

GIS features in MariaDB and MySQL

What has happened in recent years?

Hartmut Holzgraefe

Principal Support Engineer at MariaDB Inc.

hartmut@mariadb.com

August 20, 2016

- 1 GIS Introduction
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases
- 4 Performance
- 5 The End ...

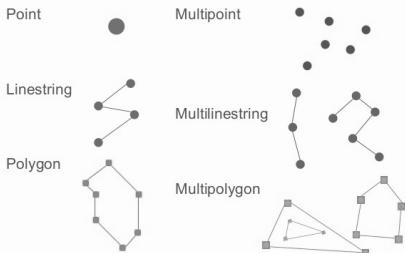
- 1 GIS Introduction
 - Examples
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases
- 4 Performance
- 5 The End ...

GIS Data Types

Geospatial Information System (GIS) data types describe geometries in a (usually) two-dimensional space. There are several different geometric subtypes:

Simple types: POINT, LINESTRING, POLYGON, GEOMETRY

Collection types: MULTIPOINT, MULTILINESTRING, MULTIPOLYGON, GEOMETRYCOLLECTION

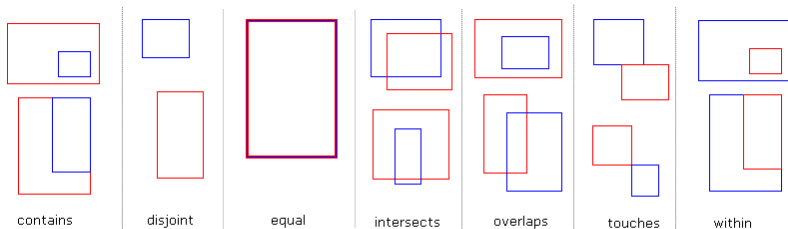


Spatial properties of a geometry can be:

- Coordinates
- Length
- Area
- Is Closed
- Bounding Rectangle
- ...

Spatial Relationships

The most important spatial relationships between two geometries:



MBRs spatial relations:  is geom1,  is geom2

Examples

- 1 GIS Introduction
 - Examples
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases
- 4 Performance
- 5 The End ...

Creating GIS columns

```
CREATE TABLE test.t1 (  
  id    INT PRIMARY KEY,  
  shape GEOMETRY,  
  poi   POINT,  
  SPATIAL INDEX (shape)  
);
```

```
CREATE TABLE test.t2 (  
  id    INT PRIMARY KEY,  
}
```

```
CALL AddGeometryColumn('catalog', 'test', 't2', 'shape', 0);
```


Adding data

```
INSERT INTO gis_table (geom)
VALUES( ST_GeomFromText('Point(51.9 8.4)'));
```

```
INSERT INTO gis_table (geom)
VALUES( ST_GeomFromText('Line(51.9 8.4,
                             52.1 8.4,
                             52.1 8.6,
                             51.9 8.6)'));
```

```
INSERT INTO gis_table (geom)
VALUES( ST_GeomFromText('Polygon((51.9 8.4,
                                   52.1 8.4,
                                   52.1 8.6,
                                   51.9 8.6,
                                   51.9 8.4))'));
```

Points in a specific rectangle

```
SELECT COUNT(*)
  FROM germany_points
 WHERE ST_CONTAINS(ST_GeomFromText('Polygon((51.9 8.4,
                                           52.1 8.4,
                                           52.1 8.6,
                                           51.9 8.6,
                                           51.9 8.4))'),
                   point);
```

Points within another geometry

```
SELECT COUNT(*)
  FROM germany_point  pt
  JOIN germany_polygon pl
 WHERE ST_CONTAINS(pl.area, pt.point)
        AND pl.name = 'Bielefeld'
        AND pt.type = 'post_box';
```

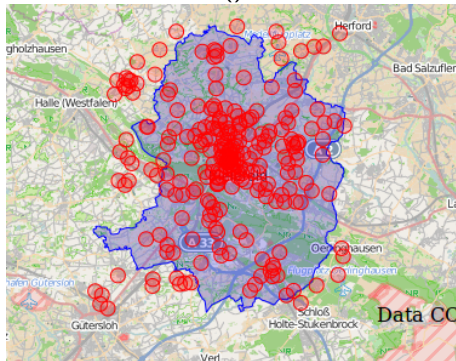
- 1 GIS Introduction
- 2 MySQL GIS History
 - How it began
- 3 Other Open Source GIS Databases
- 4 Performance
- 5 The End ...

How it began

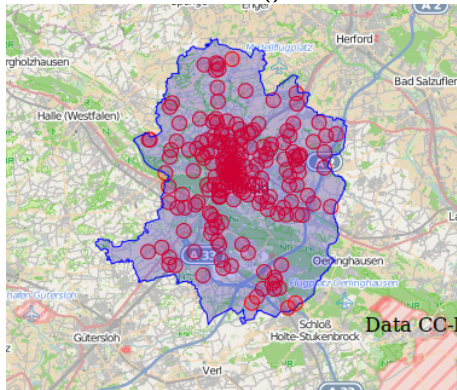
- GIS support first appeared in MySQL 4.1 (2004)
- Supports MBR (“Minimum Bounding Rectangle”) spatial relations only for now, true spatial relationships planned to be added later
- Supports spatial indexes, but only with MyISAM storage engine
- not much adoption, original developers get assigned to other features for a while

Why MBR is not enough

MBR CONTAINS()



True ST_CONTAINS()



Finally things get moving again

- In 2011 MariaDB 5.3 GA finally introduces true spatial relationship support in a production release
- In 2013 Oracle catches up with true spatial relationships in MySQL 5.6 GA
- adoption still stays low though (too little, too late?)

Things start to move into different directions

In 2015 MariaDB 10.1 GA becomes fully OpenGIS compliant (in a way)

- Adds GIS related views like `GEOMETRY_COLUMNS` to `INFORMATION_SCHEMA`
- Provides `AddGeometryColumn()` / `DropGeometryColumn()` procedures as alternatives to its native DDL syntax
- Columns can be given a default SRIDs (Spatial Reference Identifier)
- and some more ...

Meanwhile on the other side ...

MySQL 5.7 also introduces some important, but different, changes:

- begins to use Boost.Geometry instead of homegrown GIS implementation
- InnoDB now supports spatial indexes, too

What's up next?

MariaDB will get InnoDB spatial indexes as soon as it merges XtraDB 5.7
Other features currently in the MariaDB queue (but low priority):

- 3rd coordinate, e.g. for altitude
- high precision coordinates
- client side support for GIS transformations

Oracle, as usual, has not made public any further plans yet ...

Other Open Source GIS Databases

- 1 GIS Introduction
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases**
 - Advantages / Disadvantages
- 4 Performance
- 5 The End ...

GIS support for PostgreSQL comes via the PostGIS extension:

- provides many additional features beyond the basic OpenGIS requirements
- extension is licensed under GPL, not BSD like PostgreSQL itself
- installation used to be a little bit tricky ...
- ... but with `CREATE EXTENSION` it no longer is
- largest user base by far (and rightfully so)

GIS support for SQLite comes via the SpatiaLite extension

- provides a subset of OpenGIS features
- has spatial index support, but only via a special API, not plain SQL
- not very often seen in the wild (AFAICT)

With MariaDB and MySQL GIS features are part of the server itself. GIS support can be excluded at compile time, but is enabled by default.

Advantages

- no need to install / activate extra components
- mostly (MySQL 5.6+ / MariaDB 5.3+) or fully OpenGIS compliant (MariaDB 10.1)
- GIS data types and functions are 'first class citizens'
- GIS columns and indexes can be created like any other column or index
- ... no special `AddGeometryColumn()` procedure calls needed for this
- ... but is now available as an alternative in MariaDB 10.1 for OpenGIS compatibility

- not much functionality beyond OpenGIS
- so far strictly 2D only
- no support for projections / coordinate transformations yet (“the world is flat”)
- data types available in all storage engines ...
- ...but spatial indexes only in a few (MyISAM, Aria, InnoDB starting with 5.7)
- optimizer not yet too clever about combining spatial and regular indexes

- 1 GIS Introduction
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases
- 4 Performance**
- 5 The End ...

Test Data Set

To do some real world testing I have extended `osm2pgsql`, the main OpenStreetMap database import tool, with an alternative MySQL API backend and imported all German OSM data.

This provides us with a test data set of:

- 8 048 015 standalone Points
- 11 834 798 Lines
- 28 206 359 Polygons
- a total of 17GB of table data

Test data will be uploaded as `mysqldump`, MyISAM files and as InnoDB transportable table space export on <http://php-groupies.de/gis-data/> later

Query performance 1

```
SELECT COUNT(*)
  FROM planet_osm_point
 WHERE ST_CONTAINS(ST_GeomFromText('Polygon((945000 6780000,
                                     955000 6780000,
                                     955000 6820000,
                                     945000 6820000,
                                     945000 6780000)
                                     )'), way);
```

MariaDB 10.1	MySQL 5.7	PostGIS	SpatiaLite
0.08	0.08	0.03	?

Query performance 2

```
SELECT COUNT(*)
  FROM planet_osm_point n
  JOIN planet_osm_polygon p
    ON ST_CONTAINS(p.way, n.way)
 WHERE p.name = 'Bielefeld'
    AND n.amenity = 'post_box';
```

Data Set	MariaDB 10.1	MySQL 5.7	PostGIS	SpatiaLit
With spatial index only	15.7s	15.8s	—	
With extra indexes	0.95s	0.92s	0.18s	

Performance summary 1 - MariaDB vs. MySQL

- MariaDB and MySQL don't differ significantly when using MyISAM
- INSERTing and index creation are faster on MyISAM than on InnoDB
- SELECT performance is only very slightly better on MyISAM (1

Performance summary 2 - compared to others

- INSERTs on MyISAM and SpatiaLite perform roughly equally
- INSERTs on InnoDB and PostGIS also perform roughly equally well
- PostGIS has a big advantage when bulk loading with COPY though
- simple SELECTs on indexed GIS data perform equally well on MySQL/MariaDB and PostGIS
- with more complex queries the PostgreSQL optimizer takes the lead
- SpatiaLite not really comparable as index access requires spatial API

The End ...

- 1 GIS Introduction
- 2 MySQL GIS History
- 3 Other Open Source GIS Databases
- 4 Performance
- 5 The End ...**

Summary and Conclusions

- all the basic OpenGIS features are there now
- performance could be better in some situations, but is acceptable
- integrating GIS data into a MySQL ecosystem is a valid option
- if GIS is your primary focus you'd still better off with PostGIS though

Questions!

Contact hartmut@skysql.com

MariaDB GIS <https://mariadb.com/kb/en/gis-functionality/>

MySQL GIS <http://dev.mysql.com/doc/refman/5.6/en/spatial-extensions.html>

PostGIS <http://postgis.net/>

SpatiaLite <https://www.gaia-gis.it/fossil/libspatialite/index>

Table Files <http://php-groupies.de/gis-data/> (soon)

The End?

Or just the beginning?