

BEYOND POSTGIS

NEW DEVELOPMENTS IN OPEN SOURCE SPATIAL DATABASES

Karsten Vennemann



TERRA GIS
TERRESTRIAL ENVIRONMENT REGIONAL ANALYSIS
Seattle

- Intro

- Relational Databases

- PostGIS
 - JASPA
 - INGRES Geospatial
 - MySQL Spatial Support
 - HatBox – a user space extension

- File Based

- SpatiaLite

- Document Based DB

- GeoCouch

- Comparison - Summary

- Resources



- PostGIS is an extension for PostgreSQL
- Adds support for geographic objects to PostgreSQL
- Enables PostgreSQL server to be used as a backend spatial database for GIS
- Spatial operations and analysis simply mean running a (spatial) SQL query in the database
- Similar functions as SDE and much more

JASPA “**JAVA SPATIAL**”

by José Carlos Martínez,
Univ. Politécnica de Valencia.



- Beta testing phase not released (planned for June)
- Written in Java, built on top of JTS, Geotools as an alternative to PostGIS
needs PL/Java language in PostgreSQL
- Not restricted to PostgreSQL
can easily ported to other Java based databases
- For Windows and Linux; PostgreSQL and H2 databases
- Goal is to be almost 100% compatible with PostGIS
- 200 functions total: all functions that PostGIS 1.4 has completed plus some additional functions (clean polygons, create feature nodes..)

- JASPA “**JAVA SPATIAL**”
- Spatial Indexing borrowed from GIST in PostgreSQL
- First performance comparisons to PostGIS 1.4
JASPA faster: ST_Union
PostGIS faster: read/write geometries from text and binary
- Currently UMN Mapserver only as a front end using PostGIS connection
gvSIG, ogr and JASPA JDBC planned

The screenshot shows a PostgreSQL database browser interface. On the left, a tree view displays the database structure. The 'Databases (7)' folder is expanded to show 'osgis' and 'osgis_jaspa'. Under 'osgis_jaspa', the 'Schemas (6)' folder is expanded to show 'jaspa'. The 'Functions (559)' folder under 'jaspa' is expanded, and the function '_dwithin(bytea, bytea, double precision)' is selected and highlighted in blue.

On the right, the 'Properties' tab is active, showing the following details for the selected function:

Property	Value
Name	_dwithin
OID	17902
Owner	postgres
Argument count	3
Arguments	bytea, bytea, double precision
Signature arguments	bytea, bytea, double precision
Return type	boolean
Language	javau
Returns a set?	No
Source	org.cartosig.jaspa.SQL.ST_DWithin

Below the properties, the 'SQL pane' displays the following SQL code:

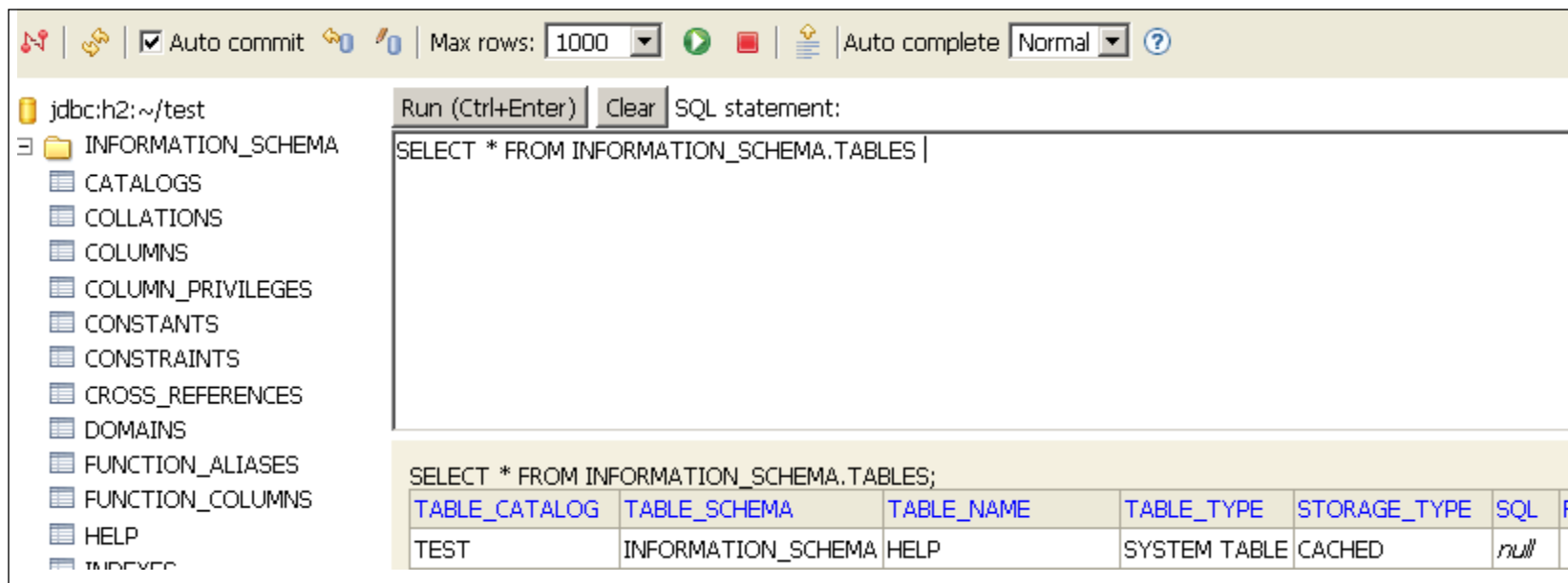
```
-- Function: jaspa._dwithin(bytea, bytea, double precision)
-- DROP FUNCTION jaspa._dwithin(bytea, bytea, double precision);
CREATE OR REPLACE FUNCTION jaspa._dwithin(bytea, bytea, double precision)
  RETURNS boolean AS
'org.cartosig.jaspa.SQL.ST_DWithin'
  LANGUAGE 'javau' IMMUTABLE STRICT
  COST 100;
ALTER FUNCTION jaspa._dwithin(bytea, bytea, double precision) OWNER TO postgres;
```

JASPA and PostGIS in the same installation of PostgreSQL

H2 Database

- Java SQL database
- Very fast, open source, JDBC API
- Embedded and server modes; in-memory databases
- Browser based Console application
- Small footprint: around 1 MB jar file size

Web based Admin tool



Auto commit Max rows: 1000 Auto complete Normal

jdbc:h2:~/test

- INFORMATION_SCHEMA
 - CATALOGS
 - COLLATIONS
 - COLUMNS
 - COLUMN_PRIVILEGES
 - CONSTANTS
 - CONSTRAINTS
 - CROSS_REFERENCES
 - DOMAINS
 - FUNCTION_ALIASES
 - FUNCTION_COLUMNS
 - HELP
 - INDEXES

Run (Ctrl+Enter) Clear SQL statement:

```
SELECT * FROM INFORMATION_SCHEMA.TABLES |
```

```
SELECT * FROM INFORMATION_SCHEMA.TABLES;
```

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE	STORAGE_TYPE	SQL	F
TEST	INFORMATION_SCHEMA	HELP	SYSTEM TABLE	CACHED	<i>null</i>	



- Great-grandmother of many DBs
Sybase, MS SQL Server, PostgreSQL
- Since 1974, became open source in 2005
- Geospatial Branch available from SVN
Community release planned 2010
- Implementation Phase 1
two-dimensional data types implemented
point, linestring, polygon, multipoint, multilinestring, multipolygon, geometrycollection
- Focus OGC SFS compliance
<http://www.opengeospatial.org/standards/sfs>
- All functions OGC SFS SQL v1.1 working



- [OGR](#) driver for Ingres – already usable with MapServer
- Rtree indexing.
- WKT / WKB support
- Work in Progress as we speak: Coordinate system support using [Proj.4](#) (completion expected in May)
- Ingres/VectorWise project cooperation with CWI Amsterdam ¹
-> Greatly improved performance
- Planned to release as OS - 2010?

¹ Research Institute in Mathematics and Computer Science



Hat Box



Apache Derby 

H2

- User Space Spatial Extension for Apache Derby and H2
- Avoiding license incompatibilities JTS – Derby – H2)
- Independent of Derby and H2 version
- User regular user visible tables and functions etc. to implement spatial functionality (same way as ArcSde)
- RTree indexing but need to join to spatial index table in SQL
- Supports the full range of OGC spatial filters, including Distance Within (DWithin) and Beyond
- Relies on JTS and GeoTools libraries
- Geometries Well Known Binary (WKB)
- Allows one geometry column per table



- Popular and fast OS database
- Spatial Functionality limited
- Only some
- Not OpenGIS compliant spatial data types and functionality
- Useful Spatial indexes only on "MyISAM" table type (R-tree)
- Spatial operations only on minimum bounding boxes (mbr) for most functions with some exceptions
BUFFER, DIFFERENCE, DISTANCE, INTERSECTION,
SYMDIFFERENCE, UNION, WITHIN
- Despite the limitations can be a good option for storage of large spatial databases



Spatialite

By Alessandro Furieri a.furieri@lgt.it

- File based portable light weight spatial DBMS
- Built on top of SQLite
- Related tools include the RasterLite library to handle Raster data and spatialitegis a simple GIS tool
- Depends on the PROJ and GEOS libraries
- QGIS can read the format, ogr driver in gdal ≥ 1.7
- Spatialite <http://www.gaia-gis.it/spatialite/>
- Potential to replace shape files as a simple data exchange format



Spatialite

- Supports standard WKT and WKB formats
- Implements SQL spatial functions
AsText(), GeomFromText(), Area() etc...
Example: `select name from counties where intersects(counties.Geometry,setsrid((MakePoint(1622794, 150532)),2285));`
- OpenGIS functions via GEOS
Overlaps(), Touches(), Union(), Buffer() etc
- Coordinate reprojection via PROJ.4
- Spatial metadata along the OpenGIS specifications
Spatial Index based on the SQLite's RTree extension

GeoCouch



- Spatial extension for Apache CouchDB
- Document oriented DB – schema free
- Collection of JSON documents
- RESTful HTTP API (PUT, POST, GET, DELETE)
- Highly concurrent- designed for local replication
- Currently SpatiaLite backend (PostGIS possible)
- Geo-spatial queries: bounding box, radius & polygon searches
- Features: (Multi-)Points, (Multi-)LineStrings, and (Multi-)Polygons
- Vertically scalable



Schema Free

```
{
  "_id": "950da89b4748cc6d08bc2f86fa2860c9",
  "_rev": "3-77f17a55f6ab11f7f6668e63a75f2281",

  "name": "Station-294",
  "date": "2009-10-20",
  "location": [140.39583, -37.48272],
  "state": "SA",
  "temperature": 18,
  "rainfall": 3,
  "atmospheric_pressure": 1021
}
```

Figure taken from Volker Mische FOSS4G 2009 Sydney



Web GIS – Typical 3 Tier Architecture



Client

JavaScript
(e.g. OpenLayers)



Server

Web Map/Feature Server
(e.g. GeoServer, MapServer)



Database

Geospatial database
(e.g. PostGIS, SpatiaLite)

Figure taken from Volker Mische FOSS4G 2009 Sydney



Web GIS – 2 Tier Architecture with Geocouch

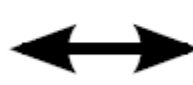


Client

JavaScript
(e.g. OpenLayers,
code to access CouchDB)



CouchDB



GeoCouch

(with spatial index)

Figure taken from Volker Mische FOSS4G 2009 Sydney

Comparison

DB system	depends on	db type	usage	Lang.	connectors			OpenGIS SFS	functionality	spatial index	storage format
					GIS	other	MapServer				
PostGIS	PostgreSQL	relational	Enterprise	C	ogr	ODBC	postgis	yes	comprehensive spatial functions	yes GIST	extended WKB
JASPA	PostgreSQL H2	relational	Academic, pot.Enterprise	JAVA	gySIG in the works	JDBC	postgis	yes	comprehensive spatial functions	yes (GIST) no	WKB
INGRESS GEOSPATIAL	INGRESS	relational	Enterprise	C, C++	ogr	ODBC	ogr	yes	nearing OpenGIS compliance	yes	WKB
MySQL	MySQL	relational	Enterprise	C, C++	ogr	ODBC	ogr	no	limited; not OpenGIS compliant	MyISAM tables R-tree indexes	MySQL specific
Hat Box	H2 Apache Derby	relational, user-space extension	Large DBs smaller to mid size systems	JAVA JAVA		JDBC JDBC		yes	OpenGIS compliant.	RTree join to spatial index table in SQL	WKB
Spatialite	SQLite	file	personal, great data exchange format	C	ogr, QGIS	ODBC	ogr	yes	almost complete OpenGIS compliant	Rtree via SQLite	WKB
GeoCouch	Couch DB	document oriented	simplifying web development, easy recovery	Erlang				partial implementation		yes	JSON

OpenGIS Implementation Specification for Geographic information - Simple feature access - SQL option:
<http://www.opengeospatial.org/standards/sfs>

WKB = “Well Known Binary”

JSON = “JavaScript Object Notation”, text-based, human-readable data interchange format

GiST = “Generalized Search Tree“, a generic form of indexing

■ OGC Definitons

<http://www.opengeospatial.org/ogc/glossary/w>

■ Well-Known Text

Representation of Spatial Reference Systems. Format that provides a standard textual representation for spatial reference system information.

■ Well-Known Binary

Representation for Geometry (WKBGeometry). Data format that provides a portable representation of a Geometry value as a contiguous stream of bytes.

■ OpenGIS® Simple Features Specifications For SQL

<http://www.opengis.org/docs/99-049.pdf>Ingres

■ Ingres

<http://community.ingres.com/wiki/GeoProjectPlan>

■ MySQL

<http://dev.mysql.com/doc/refman/5.5/en/spatial-extensions.html>