Database management II.

Objectrelational Databases
Geographical Databases

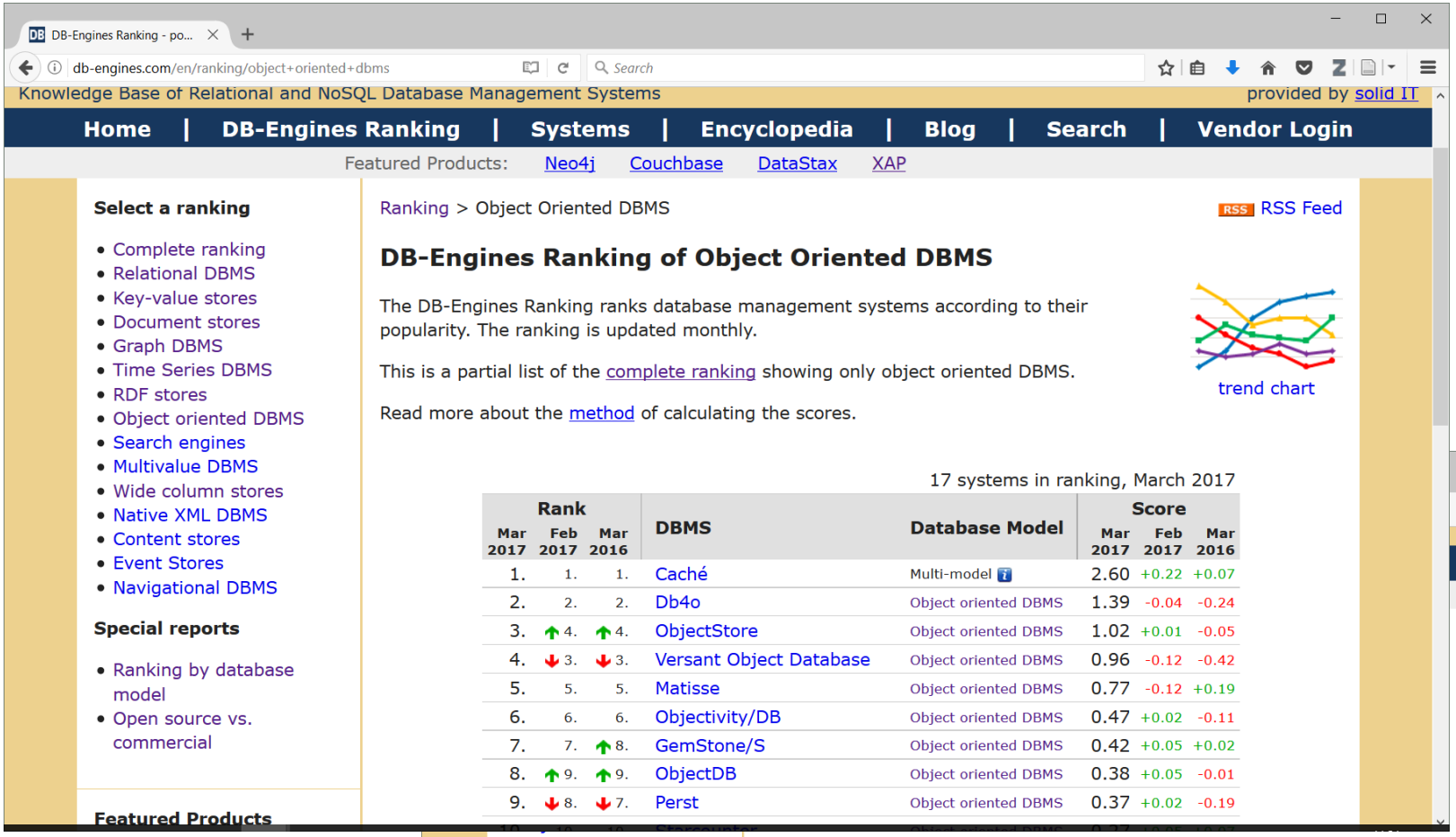
Gergely Lukács
Pázmány Péter Catholic University
Faculty of Information Technology
Budapest, Hungary
lukacs@itk.ppke.hu

Object-Relational Databases

Object-oriented databases („object databases”)

- Object-oriented concepts
  - encapsulation, inheritance, …
- Early 1990s: Research on object-oriented databases
- Object-oriented DBMS’s failed because they did not offer the efficiencies of well-entrenched relational DBMS’s.
- Niche application areas only

http://db-engines.com/en/ranking



DB-Engines Ranking - po...

db-engines.com/en/ranking/object+oriented+dbms

Search

Knowledge Base of Relational and NoSQL Database Management Systems

provided by solid IT

Home | DB-Engines Ranking | Systems | Encyclopedia | Blog | Search | Vendor Login

Featured Products: Neo4j Couchbase DataStax XAP

Select a ranking

- Complete ranking
- Relational DBMS
- Key-value stores
- Document stores
- Graph DBMS
- Time Series DBMS
- RDF stores
- Object oriented DBMS
- Search engines
- Multivalue DBMS
- Wide column stores
- Native XML DBMS
- Content stores
- Event Stores
- Navigational DBMS

Special reports

- Ranking by database model
- Open source vs. commercial

Featured Products

Ranking > Object Oriented DBMS

RSS RSS Feed

DB-Engines Ranking of Object Oriented DBMS

The DB-Engines Ranking ranks database management systems according to their popularity. The ranking is updated monthly.

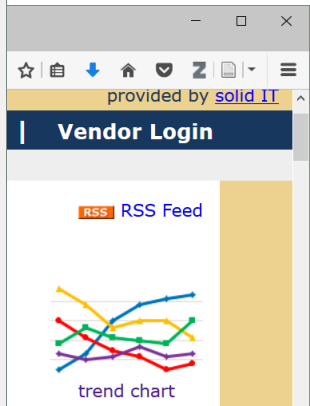
This is a partial list of the complete ranking showing only object oriented DBMS.

Read more about the method of calculating the scores.

trend chart

17 systems in ranking, March 2017

Rank Mar 2017	Rank Feb 2017	Rank Mar 2016	DBMS	Database Model	Score Mar 2017	Score Feb 2017	Score Mar 2016
1.	1.	1.	Caché	Multi-model	2.60	+0.22	+0.07
2.	2.	2.	Db4o	Object oriented DBMS	1.39	-0.04	-0.24
3.	↑ 4.	↑ 4.	ObjectStore	Object oriented DBMS	1.02	+0.01	-0.05
4.	↓ 3.	↓ 3.	Versant Object Database	Object oriented DBMS	0.96	-0.12	-0.42
5.	5.	5.	Matisse	Object oriented DBMS	0.77	-0.12	+0.19
6.	6.	6.	Objectivity/DB	Object oriented DBMS	0.47	+0.02	-0.11
7.	7.	↑ 8.	GemStone/S	Object oriented DBMS	0.42	+0.05	+0.02
8.	↑ 9.	↑ 9.	ObjectDB	Object oriented DBMS	0.38	+0.05	-0.01
9.	↓ 8.	↓ 7.	Perst	Object oriented DBMS	0.37	+0.02	-0.19



provided by solid IT

Vendor Login

RSS RSS Feed

trend chart

- Search engines
- Multivalue DBMS
- Wide column stores
- Native XML DBMS
- Content stores
- Event Stores
- Navigational DBMS

Special reports

- Ranking by database model
- Open source vs. commercial

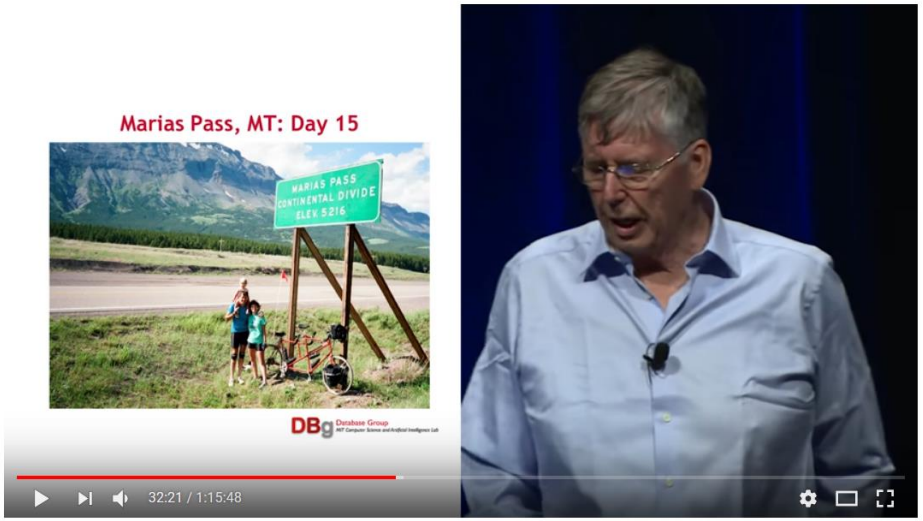
Featured Products

322 systems in ranking, March 2017

Rank Mar 2017	Rank Feb 2017	Rank Mar 2016	DBMS	Database Model	Score Mar 2017	Score Feb 2017	Score Mar 2016
1.	1.	1.	Oracle	Relational DBMS	1399.50	-4.33	-72.51
2.	2.	2.	MySQL	Relational DBMS	1376.07	-4.23	+28.36
3.	3.	3.	Microsoft SQL Server	Relational DBMS	1207.49	+4.04	+71.00
4.	4.	↑ 5.	PostgreSQL	Relational DBMS	357.64	+3.96	+58.01
5.	5.	↓ 4.	MongoDB	Document store	326.93	-8.57	+21.60
6.	6.	6.	DB2	Relational DBMS	184.91	-2.99	-3.02
7.	↑ 8.	7.	Microsoft Access	Relational DBMS	132.94	-0.45	-2.09
8.	↓ 7.	8.	Cassandra	Wide column store	129.19	-5.19	-1.14
9.	9.	↑ 10.	SQLite	Relational DBMS	116.19	+0.88	+10.42
10.	10.	↓ 9.	Redis	Key-value store	113.01	-1.03	+6.79

Object-relational databases

- Michael Stonebraker: 2014 ACM A.M. Turing Lecture, June 13 2015
  https://www.youtube.com/watch?v=BbGeKi6T6QI&t=2867s



- https://amturing.acm.org/award_winners/stonebraker_1172121.cfm
  https://en.wikipedia.org/wiki/Turing_Award

Object-relational databases

- Object-oriented extensions to relational DBMS's
  - advantages of OO, yet retain the relation as the fundamental abstraction.
  - Object-oriented models support interesting data types --- not just flat files.
    - ***Maps, XML, multimedia, etc.***
  - The relational model supports very-high-level queries.

SQL-99 and Oracle Features

- SQL-99 includes many of the object-relational features.
- However, different DBMS's use different approaches.

Oracle: User Defined Types

- A user-defined type, or UDT, is essentially a class definition, with a structure and methods.
- Two uses:
  1. As the type of an attribute of a relation.

```
SQL> CREATE TABLE states (
  2      state         VARCHAR2(30),
  3      totpop        NUMBER(9),
  4      geom          SDO_GEOMETRY);
```

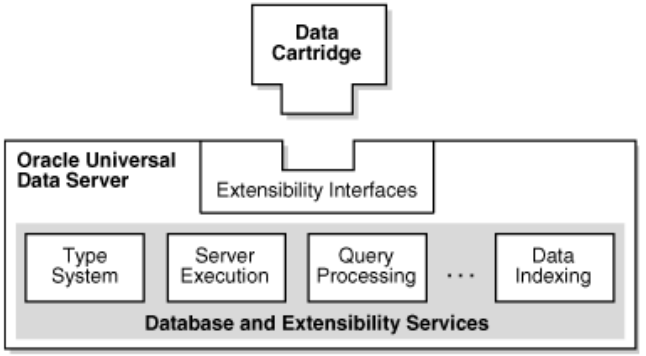
  2. As a rowtype, that is, the type of a relation.

```
create type emp_t as object (empno number, ename varchar2(10), ...);
create table emp of emp_t;
```

Oracle Types, Data Cartride Developers Guide

- In addition to the efficient and secure management of data ordered under the relational model, Oracle provides support for data organized under the object model.

1. Object types, large objects (LOBs),
2. external procedures,
3. extensible indexing
4. query optimization



can be used to build powerful, reusable server-based components called data cartridges.

Objectrelational DBMS (!)

- Extensible data types
- Methods/Operations
- Extensible indexing
  - Cost-based query optimization

How widely used are Oracle objects? (!)

- http://stackoverflow.com/questions/5767200/how-widely-used-are-oracle-objects

- some standard Oracle functionality uses Types, for instance XMLDB and Spatial (which includes declaring columns of Nested Table data types)

- What is **not** commonly done, except it would sadly appear in some college courses, is to use object-based tables instead of regular relational tables to hold regular data like employees and departments
  - **(->object-relational mapper, Hibernate**)
- While these may be nice simple examples to teach the concepts, I fear they may lead to a new generation of database developers who think this approach is suitable, more modern and therefore better than "old-fashioned" relational tables. It emphatically is not.

- Object relational technology adds a huge amount of complexity

Oracle XML DB

XML

Oracle XML DB and XML Developer's Kit enable you to develop high performance applications that process XML content and manage XML stored in the database. XDK and XML APIs allow you to generate and store XML data in the database or in documents outside the database.

- XML DB Developer's Guide
- XML Developer's Kit Programmer's Guide
- XML Java API Reference (Javadoc)
- XML C API Reference
- XML C++ API Reference

Oracle Spatial, …

Oracle Spatial and Location Information

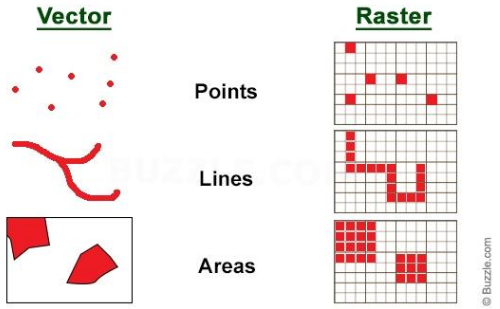
Use features described in these manuals to implement applications that manage data with spatial organization.

Spatial Developer's Guide

Spatial GeoRaster Developer's Guide

Spatial Topology and Network Data Models Developer's Guide

Spatial Java API Reference (Javadoc)



- Oracle Multimedia
- Oracle Text

Table 1-1 Data Cartridge Domains; Content and Scope

Content	Scope: Cross-Industry Uses	Scope: Industry-Specific Extensions
Scalar Data	Statistical conversion	Financial and Petroleum
Multimedia and Complex Unstructured Data	Text	Image
Audio/Video	Spatial	Legal
Medical	Broadcasting	Utilities

PostgreSQL vs Oracle

- Oracle
  - Define type, create table using type
  - Methods
  - No inheritance between tables
  - No multiple parents
  - Smart column types
- PostgreSQL
  - Table inheritance
  - …

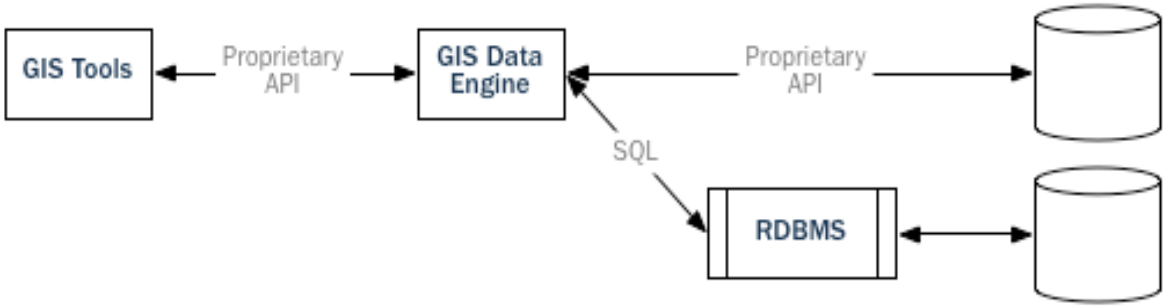
Spatial Databases

Evolution of GIS Architectures

First-Generation GIS:

Geometry, indexes, and attributes stored in file system

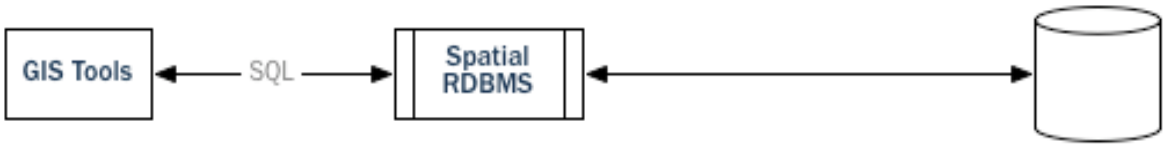
Second-Generation GIS:



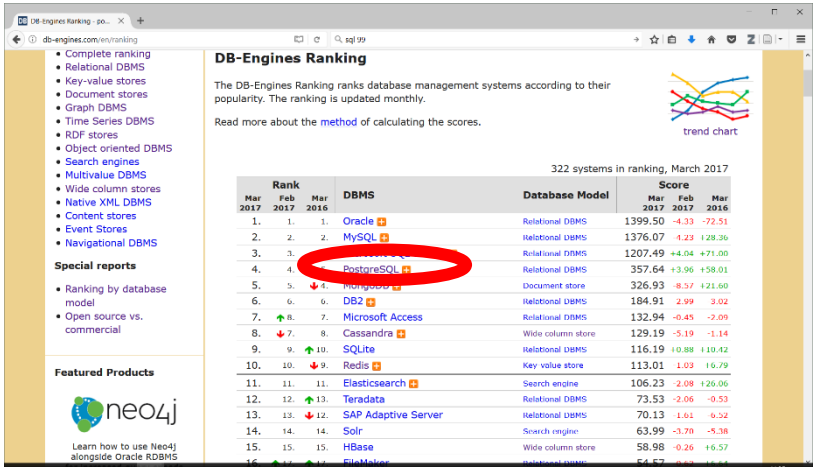
Geometry, indexes, and attributes stored in file system (may also be stored in BLOBs in RDBMS)

Attributes stored in RDBMS tables

Third-Generation GIS:



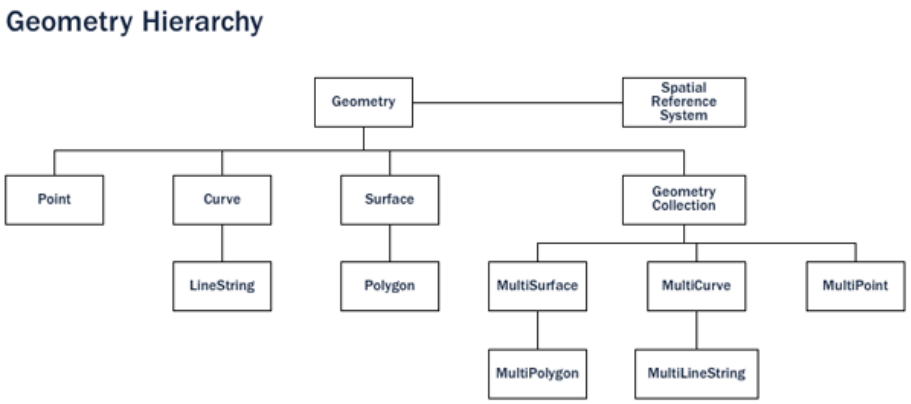
Geometry stored in ADTs in RDBMS tables with related business data

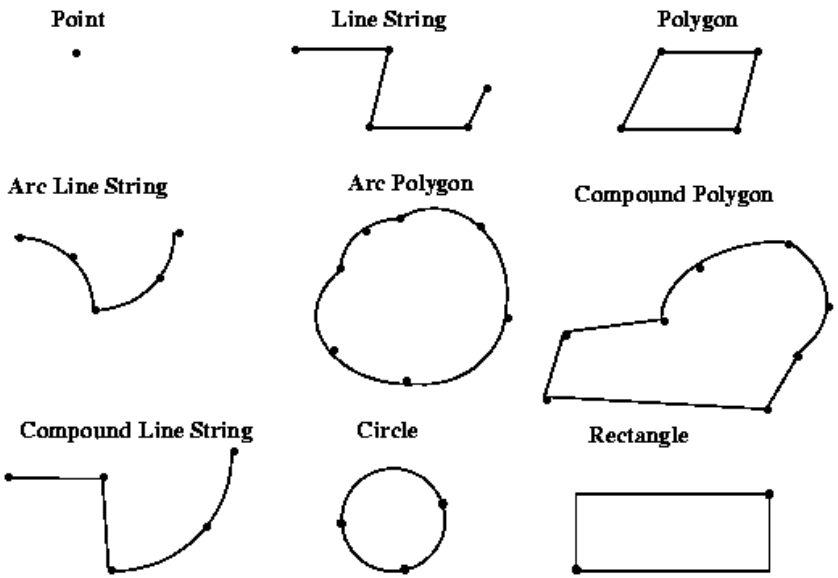


Because the development path for adding types to PostgreSQL was so straightforward, it made sense to start there. When MySQL released basic spatial types in version 4.1, the PostGIS team took a look at their code, and the exercise reinforced the original decision to use PostgreSQL. Because MySQL spatial objects had to be hacked on top of the string type as a special case, the MySQL code was spread over the entire code base. Development of PostGIS 0.1 took under a month. Doing a “MyGIS” 0.1 would have taken a lot longer, and as such, might never have seen the light of day.

PostGIS geographic data types

```
CREATE TABLE testgeog (
 gid serial PRIMARY KEY,
 the_geog geometry( point, 4326 )
);
```





Spatial Reference System, SRID

- World Geodetic System
  - WGS 1984
  - SRID 4326
  - Google, KML

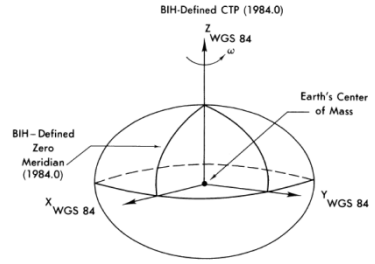
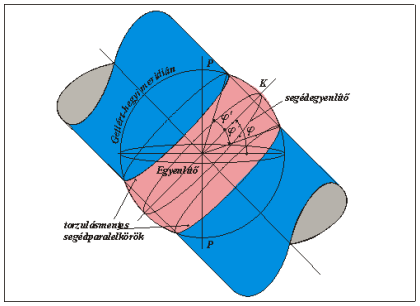
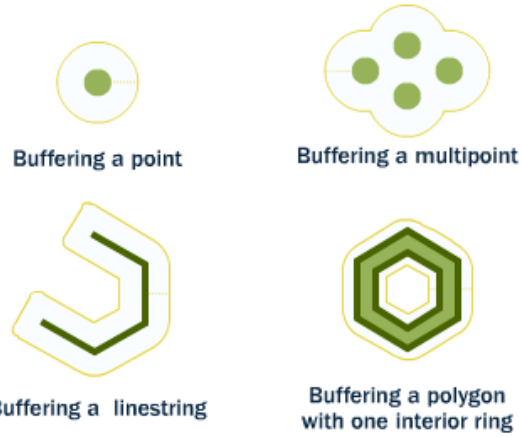


Figure 1.1. WGS 84 Reference Frame

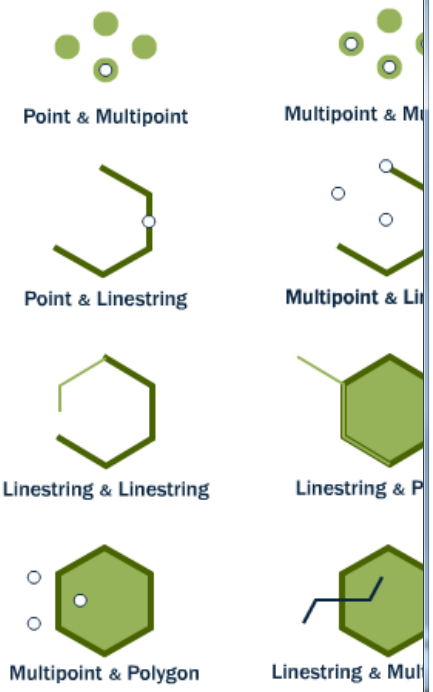


- Hungary: Egységes Országos Vetület
  - SRID 23700
  - https://hu.wikipedia.org/wiki/Egys%C3%A9ges_orsz%C3%A1gos_vet%C3%BClet

ST_Buffer



Intersects



Firefox

Chapter 8. PostGIS Reference

postgis.net/docs/referen

Google

Chapter 8. PostGIS Reference

Prev Next

Chapter 8. PostGIS Reference

Table of Contents

zotero

Firefox

Chapter 8. PostGIS Reference

postgis.net/docs/reference.html#Operators

8.7. Operators

&& — Returns TRUE if A's 2D bounding box intersects B's 2D bounding
&&& — Returns TRUE if A's 3D bounding box intersects B's 3D bounding
&< — Returns TRUE if A's bounding box overlaps or is to the left of B's
&<| — Returns TRUE if A's bounding box overlaps or is below B's.
&> — Returns TRUE if A' bounding box overlaps or is to the right of B's
<< — Returns TRUE if A's bounding box is strictly to the left of B's.
<<| — Returns TRUE if A's bounding box is strictly below B's.
= — Returns TRUE if A's bounding box is the same as B's. Uses double
>> — Returns TRUE if A's bounding box is strictly to the right of B's.
@ — Returns TRUE if A's bounding box is contained by B's.
|&> — Returns TRUE if A's bounding box overlaps or is above B's.
|>> — Returns TRUE if A's bounding box is strictly above B's.
~ — Returns TRUE if A's bounding box contains B's.
~= — Returns TRUE if A's bounding box is the same as B's.
<-> — Returns the distance between two points. For point / point che
types the distance between the floating point bounding box centroids
<#> — Returns the distance between bounding box of 2 geometries. F
geometries are double precision). Useful for doing distance ordering ar

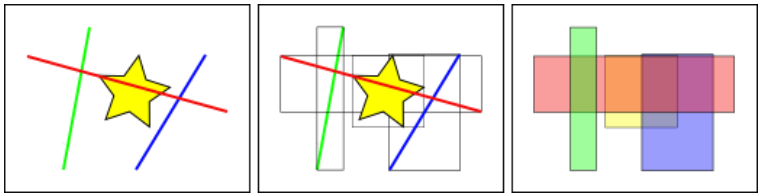
8.8. Spatial Relationships and Measur

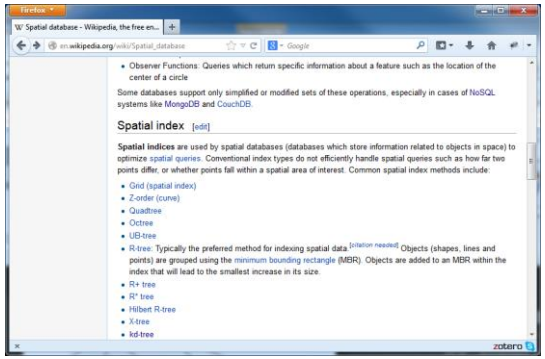
ST_3DClosestPoint — Returns the 3-dimensional point on g1 that is cl
ST_3DDistance — For geometry type Returns the 3-dimensional cartes
ST_3DDWithin — For 3d (z) geometry type Returns true if two geomet
ST_3DDFullyWithin — Returns true if all of the 3D geometries are with
ST_3DIntersects — Returns TRUE if the Geometries "spatially intersect
ST_3DLongestLine — Returns the 3-dimensional longest line between
ST_3DMaxDistance — For geometry type Returns the 3-dimensional c
ST_3DShortestLine — Returns the 3-dimensional shortest line betwee
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_Azimuth — Returns the north-based azimuth as the angle in radians measured clockwise from the vertical on pointA to pointB.
ST_Centroid — Returns the geometric center of a geometry.
ST_ClosestPoint — Returns the 2-dimensional point on g1 that is closest to g2. This is the first point of the shortest line.
ST_Contains — 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_ContainsProperly — Returns true if B intersects the interior of A but not the boundary (or exterior). A does not contain properly itself, but does contain itself.
ST_Covers — Returns 1 (TRUE) if no point in Geometry B is outside Geometry A
ST_CoveredBy — Returns 1 (TRUE) if no point in Geometry/Geography A is outside Geometry/Geography B
ST_Crosses — Returns TRUE if the supplied geometries have some, but not all, interior points in common.
ST_LineCrossingDirection — Given 2 linestrings, returns a number between -3 and 3 denoting what kind of crossing behavior. 0 is no crossing.
ST_Disjoint — Returns TRUE if the Geometries do not "spatially intersect" - if they do not share any space together.
ST_Distance — For geometry type Returns the 2-dimensional cartesian minimum distance (based on spatial ref) between two geometries in projected units. For geography type defaults to return spheroidal minimum distance between two geographies in meters.
ST_HausdorffDistance — Returns the Hausdorff distance between two geometries. Basically a measure of how similar or dissimilar 2 geometries are. Units are in the units of the spatial reference system of the geometries.
ST_MaxDistance — Returns the 2-dimensional largest distance between two geometries in projected units.
ST_Distance_Sphere — Returns minimum distance in meters between two lon/lat geometries. Uses a spherical earth and radius of 6370986 meters. Faster than ST_Distance_Spheroid ST_Distance_Spheroid, but less accurate. PostGIS versions prior to 1.5 only implemented for points.
ST_Distance_Spheroid — Returns the minimum distance between two lon/lat geometries given a particular spheroid. PostGIS versions prior to 1.5 only support points.
ST_DFullyWithin — Returns true if all of the geometries are within the specified distance of one another

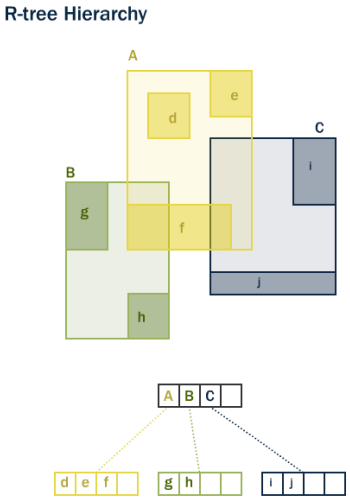
zotero

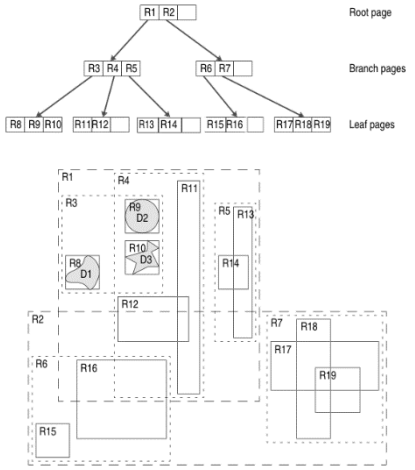
Geographic index (multidimensional index)

- Bounding box
- Tree
  - R-Tree

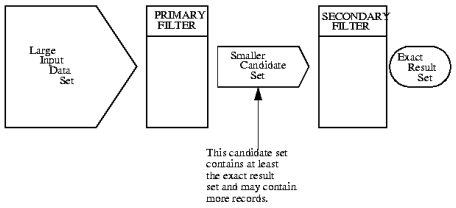


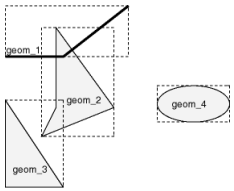


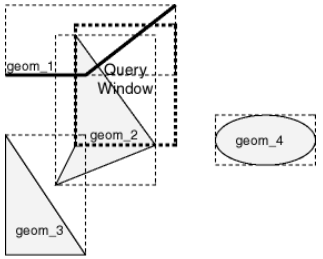




- Query processing
  - Multiple filter sets

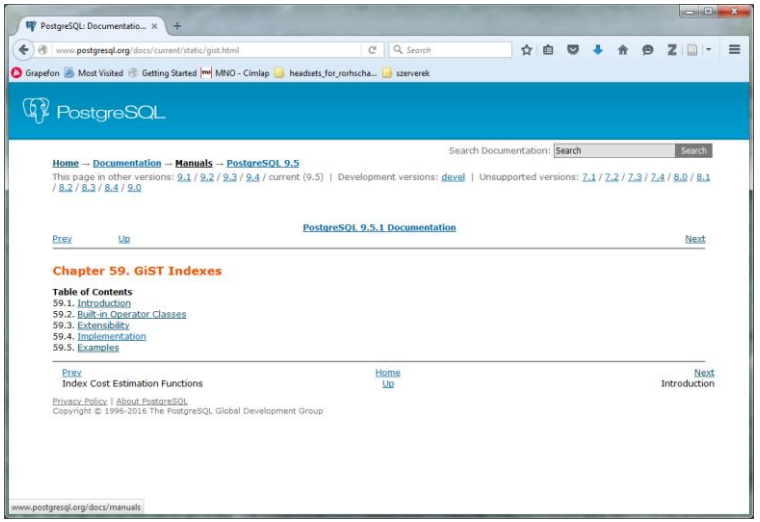






PostgreSQL – Generalized Search Tree

- balanced, tree-structured access method
- template for implementing indexes: B-tree, R-tree,…
- development of custom data types with the appropriate access methods, by an expert in the domain of the data type, rather than a database expert



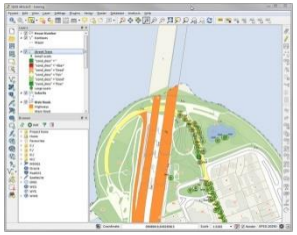
Spatial Database Offerings

- ESRI ArcSDE (on top of several different DBs)
- Oracle Spatial
- IBM DB2 Spatial Extender
- Informix Spatial DataBlade
- MS SQL Server (with ESRI SDE)
- Geomedia on MS Access
- PostGIS / PostgreSQL

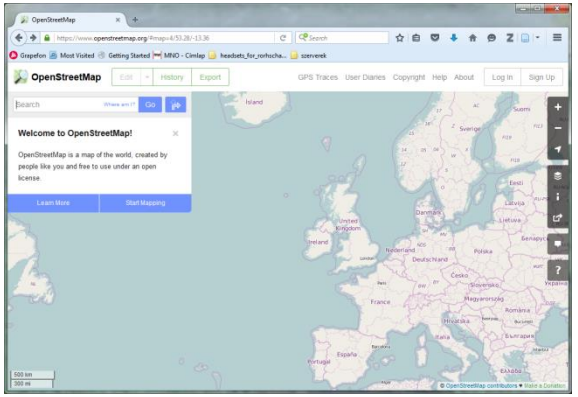
Practical hints

- PostgreSQL + PostGIS (+ pgRouting)

- Visualization: QGIS - Desktop GIS vagy Google



- Data: OpenStreetMap



- Oracle: Check the definition of SYS.XMLTYPE, allowing to manage XML data.

- Use PostgreSQL (9.1) and PostGIS (1.5) for the following tasks!
- Client: pgadmin (locally installed)
- Server connection
  - Host: csquared2.itk.ppke.hu
  - Port: 5432
  - Username: csquared2_db2
  - Maintenance db: csquared2

- It is recommended to use http://sqlformat.darold.net/ for formatting your SQL codes!

- Select a point in a Hungarian city and note its cooordinate values, using Google Maps.

- Check the location 47.501947, 19.034393 on Google Maps!

- Create a point object with SRID set to 4326, using the coordinates 47.501947, 19.034393 (create == query it with a SELECT without a FROM clause, PostgreSQL allows this, Oracle needs the DUAL table).
  - ST_Point
  - Coordinate values swapped! SRID: 4326
  - ST_SetSRID

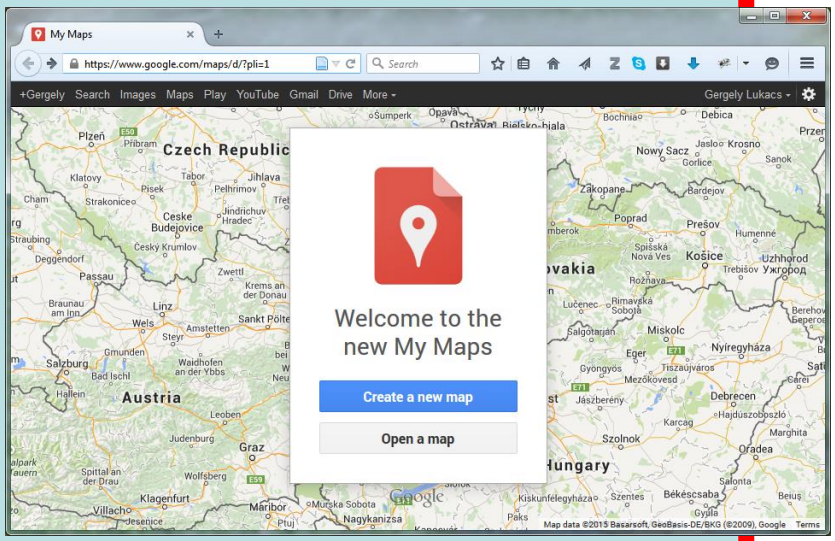
  - CAST the created geometry object to a geography object! http://postgis.net/workshops/postgis-intro/geography.html

  - Check the Well known text (WKT) format of the created geographic object (ST_AsText)

- Calculate (=query) the 2500 meters buffer zone (a geometry!) of the point!
  - (In case of a geogmetry object: ST_Transform;
    Distance can be measured in meters using SRID=23700
    for geography objects not needed!)
  - ST_Buffer

- Create a KML file from the previous buffer zone geometry and visualize it with Google Maps!
  - Creating the KML fragment with PostGIS: **ST_Askml**
  - KML Header, Footer (KML fragment comes to the place of „…”)

```
<?xml version="1.0" encoding="UTF-8"?>
<kml xmlns="http://www.opengis.net/kml/2.2">
<Document id="utvonal">
    <name>route</name>
    <Placemark>
        <name>Utvonalnév</name>
        <styleUrl>#LineStyle00</styleUrl>
<MultiGeometry>
…
</MultiGeometry>
<Style id="LineStyle00">
        <LabelStyle>
            <color>00000000</color>
            <scale>0.000000</scale>
        </LabelStyle>
        <LineStyle>
            <color>ff0098e6</color>
            <width>3.000000</width>
        </LineStyle>
    </Style>
  </Placemark>
</Document>
</kml>
```



- Google My Maps, Create a new map, Import
- Check what happens in case of a geometry type without using st_transform! (use 2.5 as radius instead of 2500)

- Query the streets containing in their name „lo” (case-insensitive) and are in the buffer zone calculated previously!
- Aggregate the geometry of the streets and visualize it with Google Maps!
  - Table with streets: hu_2po_4pgr (Open Street Map adat)
  - Attribute
    - name: hu_2po_4pgr.osm_name
    - geometry: hu_2po_4pgr.geom_way
  - ST_Intersects
  - (If data type geometry use ST_Transform)
  - ST_Collect

- Modify the previous query so that the streets (street sections) are ordered accourding to their distance from (47.501947, 19.034393) in increasing order! The result shall also contain the distance in meters!
  - ST_Distance