Modeling Spatial Data within Object Relational-Databases

Iuliana BOTHA, Anda VELICANU, Adela BÂRA
Academy of Economic Studies, Bucharest, Romania
iuliana.botha@ie.ase.ro, anda.velicanu@ie.ase.ro, bara.adela@ie.ase.ro

Spatial data can refer to elements that help place a certain object in a certain area. These elements are latitude, longitude, points, geometric figures represented by points, etc. However, when translating these elements into data that can be stored in a computer, it all comes down to numbers. The interesting part that requires attention is how to memorize them in order to obtain fast and various spatial queries. This part is where the DBMS (Data Base Management System) that contains the database acts in. In this paper, we analyzed and compared two object-relational DBMS that work with spatial data: Oracle and PostgreSQL.
Keywords: *database, object-relational database, spatial data, GIS (Geographic Information System), spatial index.*

1 Introduction

Application that work with spatial data can be developed using various technologies (web services, desktop application, client-based application), architectures (SOA - Service Oriented Architecture, EDA - Event-Driven Architecture, cloud computing) or DBMS. Also the domains in which spatial database applications can have applicability are very vast: meteorology, secret services, army, agriculture (irrigations), geology, urbanism, utilities (energy, transport, telecommunications), health care, environment etc. An example of an application developed as a web service that uses spatial data as GPS coordinates is mentioned in the article [3].
Thus, in recent years are developed rapidly some new ways to store and manipulate spatial data. Since relational databases (RDB) have limitations in the case of spatial data, the most effective way proves to be the use of object-relational databases (ORDB) [4].

2. Some aspects regarding the object-relational databases

The existence of complex and comprehensive databases is an important requirement of the new type of economy, the digital one. Currently, the most widely used since the 80s are relational databases, characterized by simplicity, relatively easy implementation and easy data retrieval facilities through a powerful query language, SQL. In time, however, the complexity of the real world has led to failed attempts to represent it by simple models.
The relational databases limits led research in a new direction in programming that began to dominate - object-oriented technology, leading to a new generation of DBMS called object-oriented.
Developing object-oriented model was due to inability of the relational model to successfully deal with very large data volumes, of great complexity, encountered most often in new types of computer applications (multimedia, Internet, spatial applications etc.). However, although OODBMS (Object-Oriented DBMS) appear to meet the needs for better software required by the new economy, markets for their use remains relatively low, the reason most often cited being the difficult query with a large consumption of computational resources.
An intermediary level between relational and object-oriented databases comes through the new hybrid type, namely object-relational databases, which presents object-oriented features (in particular the

fundamental characteristics of objects: encapsulation, inheritance, polymorphism, etc.) as extensions of the relational model [8]. Thus, ORDB combines the benefits of both the relational and the object-oriented models such as scalability and support for complex data types (large objects, multimedia data, spatial data, user defined object types, etc.). Also called extended-relational, the object-relational data model is exemplified by the querying language version SQL-99.

In Fig. 1 is presented a very suggestive summary graph with a classification of the types of DBMS. This view was proposed in 1996 by Stonebraker [9], the ORDBMS being called “the next wave”. However, the schema does not include the pre-relational models (hierarchical model and network model), considered obsolete at this stage of databases development.

The graph is simple and suggests how complex data and complex queries influence each other. It is a two-dimensional graphical representation: the abscissa refers to the ability to define complex data types and the ordinate presents the ability to query databases.

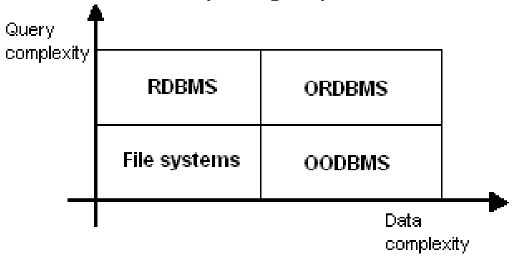


Fig.1. DBMS types classification - adapted from [9]

The bottom left quadrant contains the types of applications that process simple data types and requires no data query. These types of applications use directly the file system of the operating system for persistent data storage.

The top left quadrant refers to the relational database management systems (RDBMS), which processes simple data types, but allow complex queries.

Going forward, the lower right quadrant presents the object-oriented database management systems (OODBMS), which are handling with complex data types. However, resolving queries is quite difficult, since for each query must be provided the necessary relationships in the structure of objects.

In the upper right quadrant are included the object-relational database management systems (ORDBMS), which allow complex processing and resolving complex queries. Object-relational model is obviously the most complete, because allows both user-defined data types and complex queries. For this reason, the object-relational data model is also called extended-relational or universal and it can be seen as an attempt to capture as many as possible of the object-oriented concepts.

Using this hybrid type of database has its main reason for that:

- In many cases, the existing applications are already based on a relational data model. This calls for coexistence with the relational model as long as we do not want to redesign the applications based on a common object model to be included in a single OODB;
- Performance and scalability are important properties of an application, and in this respect, OODBMS have not yet shown advantages over RDBMS.

The current trends involve the transition from relational to the object-relational. Generally, this transition is achieved by gradually adding object model features to the relational one.

Regarding the ORDBMS, the offer is very generous and covers a wide scale of cost and performance, going from the DBMS that can be used free (unlicensed or with public license, such as PostgreSQL) to the commercial ones such as Oracle 10g, DB2 UDB 8, and SQL Server 2005. They all have adopted the object-relational data model to supports relational tables, abstract data types, and functions on abstract data types.

3. Some aspects regarding the spatial databases

Spatial objects consisting of lines, surfaces, volumes and objects of higher dimensions are commonly used in computer aided design applications, cartography or geographic information systems. These are described both by spatial attributes (length, pattern, perimeter, area, volume, etc.) and by non-spatial attributes (the time, the owner, membership of a superior structure, etc.). The values of objects' spatial attributes represent spatial data.

Spatial data can be divided into specific data and regional data. The specific data is a point, which is completely characterized by its location in a multidimensional space. It can be obtained directly from measurements or by converting for it to be more easily stored and retrieved.

The regional data is characterized by location (one fixed point in a region) and destination (a line in 2D or a surface in 3D).

Spatial data can also have a certain rank (e.g. "Cities in Europe"), approach (e.g. "Three nearby lakes in France"), junction (e.g. "Pairs of city, in Romania, located 50 km apart) [5].

The queries performed for spatial data can be: local, in the area or in the neighborhood (most difficult because it requires the evaluation of proximity - e.g. determining the road which passes closest to a specified region) [11].

Examples of spatial DBMS are:

- Boeing Spatial Query Server;
- PostgreSQL, which uses PostGIS spatial extension to implement standard "geometric" data and corresponding functions;
- MySQL, which implements "geometric" data type and some spatial functions. The functions that test spatial relations are limited to working with rectangles;
- Spatial Databox is a spatial front end to a relational system that offers spatial queries. It also offers real-time response to queries on interactive maps;
- Oracle Spatial (a feature of Oracle Database 11g Enterprise Edition) includes full support for Web and 3D services that allow managing all types of geospatial data, including raster and vector data, topologies and network models. It was designed to satisfy GIS requirements for applications such as territorial management, utilities, defense;
- SQL Server 2008 (spatial data stored in two types of data: GEOMETRY and GEOGRAPHY, for plane models and ellipsoid data).

The types of applications that work with spatial database systems are Geographic Information Systems (GIS), Computer Aided Design (CAD), Multimedia, etc.

A geographic information system is used to create, store, analyze and process information from the geometric space using a computer automated process. GIS technology can be used in various scientific fields such as resource management, studies on environmental impact, mapping, planning routes.

GIS has a unique way of organizing the database. There are two databases: one to store spatial distribution of geographic elements (through a system of x, y coordinates) and another to store the attributes of these elements (e.g. the length, the width, the number of bands, and the construction material of a road).

There are several ways to represent geographic data such as raster or vector. There is GIS software that works in either raster or vector mode, although most of them accept both types of formats.

4. Using spatial data for modeling object-relational databases

As shown above the object-relational databases are an extension of the relational ones. Thus, the logic of storing and retrieving data is the same like in the relational case. The main difference consists in new types of data, some of them user defined (like object classes), and in the ability of manipulate them. Spatial data are important resources, which need to be manipulated, in order to use them in GIS applications.

However, one can observe that, in a table of such database, the main part of the columns have nothing in particular, being just standard columns. The exceptions (and the extensions) come from these columns that contain complex data: large objects, object types (with specific properties) or spatial data.

Therefore, regarding these features of the object-relational databases, several interesting metrics can be defined in order to use them for optimizations or just as statistics. Some examples of metrics are:

(1) – N_{OT}: the number of object types (known as object classes) involved in the definition of the columns from a user schema [6]:

$$N_{OT} = \sum_{i \in schema} ObjectType_i$$

(2) – P_{OT}: the percentage of inherited object types in the total number of object types (N_{OT}) that are involved in the definition of the columns from a user schema:

$$P_{OT} = \frac{\sum_{i \in Schema} InheritedObjectType_i}{N_{OT}} * 100$$

(3) – P_{CS}: the percentage of columns with spatial data type in the total number of columns from a user schema:

$$P_{CS} = \frac{\sum_{i \in Schema} SpatialTypeColumn_i}{\sum_{i \in Schema} Column_i} * 100$$

(4) – P_{CC}: the percentage of complex columns in a user schema [6]:

$$P_{CC} = \frac{\sum_{i \in Schema} ComplexColumn_i}{\sum_{i \in Schema} Column_i} * 100$$

(5) – P_{SC}: the percentage of spatial data in the total number of complex columns:

$$Psc = \frac{\sum_{i \in Schema} SpatialTypeColumn_i}{\sum_{i \in Schema} ComplexColumn_i} * 100$$

Using spatial data into object-relational databases is an important characteristic for the enterprises that develop or use GIS projects. Below, we provide an overview of spatial characteristics of Oracle and PostgreSQL, the ORDBMS mostly used.

Oracle and spatial data

Oracle DBMS has a feature that is called Oracle Spatial, which allows users to manage regional and geographic data in a native data type in the Oracle database. Thus, there can be developed a wide range of applications that may include: automated mapping, management facilities, geographic information systems or wireless location services.

Oracle Spatial provides facilities for storing, updating, querying spatial information from an Oracle database and it includes the following elements: MDSYS (Multi Dimensional SYStem) schema that governs the storage, syntax and semantics of data types supported by the geometric database; a mechanism for indexing spatial data by different ordering criteria; operators, functions and procedures for querying the areas of interest, the meeting space and other spatial analysis operations; functions and procedures specific for spatial data and adjustment operations; topographic data models for working with nodes, edges and faces in a topology; data networks models for representing the objects which are modeled as nodes and links in a network; GeoRaster, a feature that enables storage, indexing, querying, analysis and transferring the GeoRaster data.

Oracle Spatial uses a two-tier query model to resolve spatial queries and unions. The term is used to indicate that two distinct operations are used to perform queries. The result of combining the two operations is precisely the set of results. The two operations are: primary operation – meaning that the primary filter permits fast selection of records to the secondary filter; secondary operation – is based mostly on the operations performed by the secondary filter on the set of results from the primary filter and is usually more expensive.

Indexing spatial data is a mechanism to

decrease the number of searches, and a spatial index (considered logic) is used to locate objects in the same area of data (window query) or from different locations (spatial junction). Oracle Spatial uses two types of indexing: R-Tree (SDO_INDEX_TABLE table, maintaining the SDO_RTREE_SEQ_NAME sequence in the virtual table USER_SDO_INDEX_METADATA) and QuadTree (a tree structure, whose nodes have up to four children and is used to divide two-dimensional space, by recursively subdividing itself in four regions) [1].

To determine spatial relations, Oracle Spatial has several methods of secondary filtering:

- SDO_RELATE operator evaluates topological criteria;
- SDO_WITHIN_DISTANCE operator determines if two spatial objects are within a certain distance;
- SDO_INTERSECTION operator determines the topological intersection between two spatial elements;
- SDO_AREA operator calculates the area of a geometric figure;
- SDO_MAX_MBR_ORDINATE operator determines the maximum value for a coordinate (x or y);
- SDO_LENGTH operator calculates the length of a geometric figure;
- SDO_DIFFENECE operator determines the geometric element that results from the difference of two other spatial objects;
- SDO_CENTROID operator gets the centre of a polygon;
- SDO_NN operator identifies the nearest neighbor of a spatial object.

PostgreSQL and spatial data

PostgreSQL is an object-relational database developed by online community as an open-source alternative to the commercial databases like Oracle and Informix. PostgreSQL has native geometric types, built for academic research purposes, but they are too limited for using them in GIS projects and in spatial analysis.

The spatial extension to the PostgreSQL is called PostGIS and provides the ability to store relational attributes as well as spatial properties of the objects inside the database server. PostGIS supports the simple spatial features proposed by the Open GeoSpatial Consortium (OGC) - point, line, polygon, multipoint, multiline, multipolygon, and geometry collection [10]. In addition, PostGIS provides mechanisms for high speed spatial indexing using three types of access methods for indexes: B-Tree, R-Tree indexes and GiST (Generalized Search Tree).

In addition to normal B-Tree indexing, which is used only for data that can be sorted along one axis, R-Tree and GiST are used to speed up searches on all kinds of spatial data. R-Tree indexing algorithm breaks up data into smaller polygons, but according [7] the PostgreSQL R-Tree implementation is not as efficient and robust as the GiST implementation. GiST indexes decompose data according their spatial representation (e.g. "things to one side", "things which overlap", "things which are inside").

The SQL statements and functions provide the standard features for updating and retrieving spatial data from the database, for perform spatial operations (like area, length, union, intersection) and for determine spatial relations.

In terms of spatial analysis, one can make powerful queries such as location based queries (nearest neighbor) and sub-queries (searches based upon linear or spatial indexing of the subsets).

To determine spatial relations, PostGIS has a series of functions, of which we can specify:

- ST_INTERSECTION returns a geometry that represents the portion between two spatial elements;
- ST_AREA returns the area of a geometric figure;

- ST_LENGTH calculates the length of a geometric figure;
- ST_DIFFERENCE returns a geometry that represents a part of a spatial object that does not intersect with another object;
- ST_UNION returns a geometry that represents the common part of two spatial objects;
- ST_GEOMETRYTYPE returns the geometry type of an object.

Oracle vs. PostgreSQL

One can compare the two DBMS presented above considering their spatial characteristics and object-relational features.

Both Oracle and PostgreSQL with their spatial extensions, OracleSpatial and PostGIS, provides transactional spatial databases. Multiple users can access and edit the data stored simultaneously without file locking or data corruption issues. A main difference is that, unlike Oracle Spatial, PostGIS is an open-source project, which does not need licenses or restrictions for usage.

In Table 1 are presented some features that refer to Oracle and PostgreSQL historic, types of spatial data accepted in the database, spatial restrictions, spatial applicability, steps in developing a spatial application, object-relational functionalities.

Table 1. Oracle vs. PostgreSQL - synthetic comparison

	Oracle	**PostgreSQL**
Object-relational features	- type objects (similar with class objects) - each type has specific attributes and methods - there are implemented the following properties: type inheritance, polymorphism, encapsulation	- type objects (class objects) - each type has specific attributes and methods - there are implemented the following properties: type inheritance, polymorphism, encapsulation
Spatial features (**Oracle Spatial** vs. **PostGIS**)	*Oracle and PostgreSQL spatial history*	
	- Oracle 4 had spatial-data capability - Oracle 7 had "Spatial Data Option" or "SDO". - Since Oracle 8, "Oracle Spatial" extension exists. The primary spatial indexing system uses a standard R-tree index.	- PostgreSQL has its own native geometric data type, but this is unable for being used in GIS projects - spatial features since PostgreSQL 7.1.x database server, when it was released PostGIS 0.1
	Steps in building a spatial application in Oracle and PostgreSQL	
	1. Create the table in which the spatial data will be stored. 2. Adding the appropriate entries in the table. SDO_GEOMETRY is the spatial field. 3.Updating the USER_SDO_GEOM_METADATA view to reflect the dimensional information for spatial data. 4. Creating the spatial index (an R-	1. Create the table in which the spatial data will be stored. 2. Adding the appropriate entries in the table. GEOGRAPHY is the spatial field. 3. Updating the SDE_LAYERS, SDE_TABLE_REGISTRY, SDE_GEOMETRY_COLUMNS, SDE_COLUMN_REGISTRY tables to reflect the dimensional information

	Oracle	PostgreSQL
	tree index). 5. Executing spatial queries.	for spatial data. 4. Creating the spatial indexes (R-Tree or GiST). 5. Executing spatial queries.
	Operating systems on which Oracle and PostgreSQL can run	
	- Windows - Linux - Unix - Solaris - Mac OS	- Windows - Linux - Unix - Solaris - Mac OS
	Oracle and PostgreSQL spatial data type	
	SDO_GEOMETRY	GEOGRAPHY

In Oracle the spatial field has some parameters: SDO_GEOMETRY (polygon_dimension, latitude_longitude, SDO_POINT_TYPE, SDO_ELEM_INFO_ARRAY (SDO_STARTING_OFFSET, SDO_ETYPE, SDO_INTERPRETATION), SDO_ORDINATE_ARRAYV (variable number of parameters)). SDO_STARTING_OFFSET represents the offset from which to begin storing in the SDO_ORDINATE vector and starts at 1, not 0.
The step 3 in developing an Oracle Spatial application is required to be performed before the index is created. It is performed once for each level (table-column combination). It contains information about the name of the table that contains spatial data, the column from the table that is SDO_GEOMETRY type, the size of the geometry and a number (SID) which specifies the value of the coordinates system [12].
Spatial indexing is a mechanism that helps to increase the search performance on a table based on spatial criteria. An R-tree index approximates each geometry with the smallest rectangle that contains the geometry (called MBR - Minimum Bounding Rectangle). When there are more geometries, an R-tree index consists of a hierarchical indexing of MBR rectangles [2].

5. Conclusions
The integration of the spatial data into object-relational databases is an absolutely necessary characteristic for today's enterprises that uses a GIS. The paper provide a synthetic look of spatial features of two ORDBMS, namely PostgreSQL and Oracle. These two products cover the sector of open source and commercial object-relational technology today mostly used.

6. Acknowledgments
This paper is a result of the project POSDRU/6/1.5/S/11 „Doctoral Program and PhD Students in the education research and innovation triangle". This project is co funded by European Social Fund through The Sectorial Operational Program for Human Resources Development 2007-2013, coordinated by The Bucharest Academy of Economic Studies.
This paper presents some results of the research project PN II, TE Program, Code 332: "Solutii informatice pentru asistarea procesului decizional in mediile incerte si cu evolutii putin predictibile in vederea integrarii in retele de tip Grid", financed within the framework of People research program.

References

[1] D. Abugov, N. Alexander, *Oracle Spatial User's Guide and Reference, 10g Release 1 (10.1)*, Oracle Corporation, 2003.

[2] T. Brinkhoff, H.P. Kriegel, B. Seeger, „Efficient Processing of Spatial Joins Using R-trees", International Conference on Management of Data, *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, Volume 22, Issue 2, pp. 237 - 246, 1993, Washington D.C., ISSN 0163-5808.

[3] A. Dioşteanu, L. Cotfas, "Agent Based Knowledge Based Management Solution using Ontology, Semantic Web Services and GIS", *Informatica Economica Journal*, Vol. 13, No. 4 / 2009, pp. 90-98, ISSN 1453-1305.

[4] W. Huibing, "Extending object-relational database to support spatio-temporal data", *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B2. Beijing 2008, ISSN 1682-1750.

[5] J. LeSage, S. Banerjee, M. M. Fischer, P. Congdon, „Spatial statistics: Methods, models & computation", *Computational Statistics & Data Analysis*, Volume 53, Issue 8, 15 June 2009, pp. 2781-2785, ISSN: 0167-9473.

[6] M. Piattini, C. Calero, H. Sahraoui, "An empirical study with object-relational databases metrics", *Proceedings of the 7th International Conference on Software Quality*, London, 2002, pp. 298–309, ISBN 3-540-43749-5.

[7] PostGIS 1.5.1 Manual - http://postgis.refractions.net/docs/index.html

[8] Gh. Sabau, "Comparison of RDBMS, OODBMS and ORDBMS", *Proceedings of the 8th International Conference on Informatics in Economy,* Bucharest, 2007, pp. 792-796, ISBN 978-973-594-921-1.

[9] M. Stonebraker, D. Moore, *Object-Relational DBMS - The Next Great Wave*, Morgan-Kaufmann, 1996, ISBN:155-860-397-2.

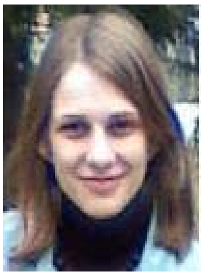
[10] http://www.refractions.net/products/postgis

[11] http://www.spatial.cs.umn.edu/

[12] A. Yeung, B. Hall, *Spatial Database Systems: Design, Implementation and Project Management*, Springer, 2006, ISBN 1-4020-5393-2.

[13]

Iuliana BOTHA is an Assistant Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Cybernetics, Statistics and Economic Informatics in 2006 and the Databases for Business Support master program organized by the Academy of Economic Studies of Bucharest in 2008. Currently, she is a PhD student in the field of Economic Informatics at the Academy of Economic Studies. She is co-author of two books, 5 published articles (one article ISI indexed and another three included in international databases), 9 scientific papers published in conferences proceedings (among which 3 paper ISI indexed). She participated as team member in 3 research projects that have been financed from national research programs. From 2007, she is the scientific secretary of the master program *Databases for Business Support* and she is also a member of INFOREC professional association. Her scientific fields of interest include: Databases, Database Management Systems, Design of Economic Information Systems, Grid Computing, e-Learning Technologies.



Anda Velicanu is a Pre-Assistant Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Economic Cybernetics, Statistics and Informatics of the Bucharest Academy of Economic Studies, in 2008. She is a PhD student in the field of Economic Informatics at the Academy of Economic Studies and since January 2009, she is a Pre-Assistant Lecturer. She teaches Database, Database Management Systems and Economic Informatics seminars at the following faculties: Economic Cybernetics, Statistics and Informatics, Commerce, Marketing and International Business and Economics. Her research activity can be observed in the following achievements: 5 diplomas, 2 scientific awards, 3 proceedings, 2 articles published in scientific reviews, 1 research contract, 1 book and 1 research grant. She is a member of INFOREC professional association. Her scientific fields of interest include: Databases, Database Management Systems, Programming, Information Systems.

Adela BÂRA is a Lecturer at the Economic Informatics Department at the Faculty of Cybernetics, Statistics and Economic Informatics from the Academy of Economic Studies of Bucharest. She has graduated the Faculty of Economic Cybernetics in 2002, holds a PhD diploma in Economics from 2007. She is the author of 7 books in the domain of economic informatics, over 40 published scientific papers and articles (among which over 20 articles are indexed in international databases, ISI proceedings, SCOPUS and 2 of them are ISI indexed). She participated (as director or as team member) in 4 research projects that have been financed from national research programs. She is a member of INFOREC professional association. From May 2009, she is the director of the Oracle Excellence Centre in the university, responsible for the implementation of the Oracle Academy Initiative program. Domains of competence: Database systems, Data warehouses, OLAP and Business Intelligence, Executive Information Systems, Decision Support Systems, Data Mining.