Using Oracle Spatial towards the effective handling of spatial data of cadastre

Karel Janecka[1]

[1]Department of Mathematics, Faculty of Applied Sciences, University of West Bohemia, Univerzitni 22,
306 14, Pilsen, Czech Republic
kjanecka@kma.zcu.cz

Abstract.

The aim of the article is to show the possible usage of Oracle Spatial for storing of spatial data, concretely the digital cadastral map, towards the effective spatial data handling. Digital cadastral map is stored in the Cadastral Information System which is maintained by the Czech Office for Surveying, Mapping and Cadastre in Prague (COSMC). As it was already shown the actual physical data model of the Cadastral Information System for storing of vector spatial data has some important shortcomings. These shortcomings do not allow the using of the stored data in some types of web applications, e.g. in applications when we prefer a fast retrieving and the visualization of all requested data. On this account I tried to find out whether the use of Oracle Spatial data structures would be more appropriate. Concretely I tested the object-relational data model (which uses the SDO_GEOMETRY object data type for storing the geometry) and the winged-edge structure (which uses the SDO_TOPO_GEOMETRY data type). The software for transformation of spatial data was created. This software allows transform source data coming from the Cadastral Information System to the mentioned Oracle Spatial data structures. After importing of transformed data into Oracle database the time needed for selecting of spatial data by query window was measured. Also the space for storing of spatial data was monitored. In article is shown that if we have data stored by using of the SDO_GEOMETRY object data type that the data are selected from the database much faster than if we use for storing of spatial data SDO_TOPO_GEOMETRY data type. As well a less of disc space is required in the case of storing digital cadastral map by using of the SDO_GEOMETRY data type. Due to the INSPIRE initiative there is a strong demand on the effective way of storing and retrieving of mentioned spatial cadastral data in a web environment. Mainly for the purpose of accessing cadastral data via WFS or WMS so called Publication database has been created at COSMC. The creation of the Publication database has been based on the results mentioned in this paper which means that the spatial data are stored through the use of the SDO_GEOMETRY object data type.

Keywords: Oracle Spatial, cadastre, object-relational data model, winged-edge structure, INSPIRE

1 Introduction

The Cadastral Information System (CIS) was built on the Oracle database system. The extension for processing of spatial data – Spatial, is also used. It is quite necessary to mention the fact that the standard data type `SDO_GEOMETRY` available in Spatial for storing of geometry of spatial feature is used but as it was already shown that the actual physical data model of the Cadastral Information System for storing of vector spatial data has some important shortcomings [2]. These shortcomings do not allow the using of the stored data in some types of web applications, e.g. in applications when we prefer a fast retrieving and the visualization of all requested data. The reason is that geometries of spatial features like parcels are stored in an inappropriate way. Very important activity regarding with the way of storage of spatial data is the INSPIRE directive. Accordance with this directive the provider of spatial data has to offer spatial data among others via the Web Feature Service (WFS). Therefore it is necessary to find the appropriate data model in which cadastral data will be stored. Upon this data model will be build Publication database which will be maintained by the Czech Office for Surveying, Mapping and Cadastre in Prague. This database will act as the technical base for INSPIRE related activities and will be built upon the Oracle database. Therefore there is also used Oracle 10g Enterprise Edition in this paper.

The remainder of this paper is structured as follows. In section 2 is described the actual data model for storing of vector spatial data – digital cadastral map, in the Cadastral Information System. Also there is an example why the actual data model is not appropriate for providing data for example via WFS. In section 3 is generally described Oracle Spatial and two ways how to store vector spatial data using Oracle Spatial. In the section 4 is described software DKM2Spatial which was created for data conversion between the actual data model and the new proposed solution. Section 5 gives a comparison between the object-relational data model and topology data model in Oracle Spatial. There are mentioned results of performed tests. Conclusions can be found in section 6.

2 Cadastral information system

The cadastral information system, maintained by the Czech Office for Surveying, Mapping and Cadastre in Prague (COSMC), is the main source of spatial data in the Czech Republic. Spatial data inside this information system contain the highest level of detail. Due to this fact CIS is very important source of data for other information systems working with spatial data. It is necessary to mention the fact that for using of cadastral data in some other application one has to make some conversion. As was stated in [2] the actual data model is not suitable for applications which require fast displaying of data. For example – if we want to get the geometry of one concrete parcel we have to join four tables. All needed attributes for identification of concrete parcel are stored in table `PARCELY`. Each parcel boundary consists from single parts which together constitute the final cadastral boundary. The coordinates of points which make one single part of cadastral boundary are stored in table `SOURADNICE_OBRAZU`. For storing of coordinates of points is used the `SDO_GEOMETRY` object data type. As you can see on figure 1 we have to join tables `PARCELY`, `HRANICE_PARCEL`, `SPOJENI_B_POLOH` and `SOURADNICE_OBRAZU` – four tables for getting the geometry of parcel.

Well, but how the way of storage of spatial data deals with the INSPIRE activity? As it was mentioned in the introduction the provider of spatial data has to offer spatial data among others via the Web Feature Service (WFS). For this reason the data must be stored in a more appropriate way. I propose the solution to store the data in Oracle Spatial. But the Cadastral Information System is already built on the Oracle database system and uses Spatial. So what is the difference between older and new solution? The fundamental difference is in way of using of Spatial. The older solution uses Spatial only for storing of points coordinates. The new proposed solution will use Spatial more widely. The first basic idea for the new solution was to use the `SDO_GEOMETRY` object data type and store complete geometry (e.g. boundary of parcel) using the `SDO_GEOMETRY` data type as an object in one place – in one cell of table. The second idea was to use the `SDO_TOPO_GEOMETRY` data type for storing of geometry. It means [3] to use topology data model, concretely winged-edge structure. I tested both approaches – object-relational data model by using the `SDO_GEOMETRY` object data type and topology approach by using the `SDO_TOPO_GEOMETRY` data type. I was interested mainly in the requirements on the disk space for storage of data and the time needed for spatial query evaluation.

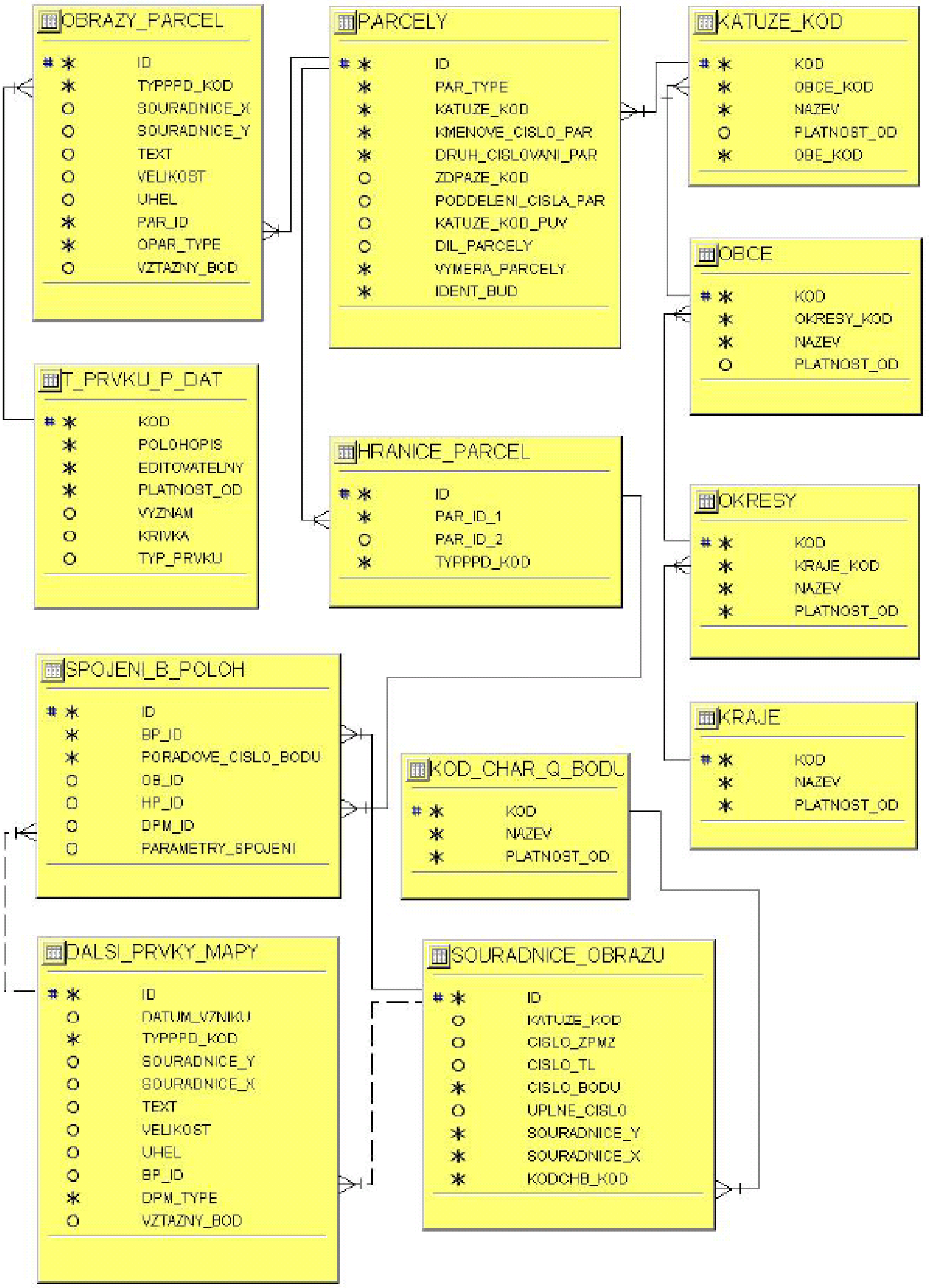

Fig. 1. The part of the data model of the Cadastral Information System for parcels

3 Oracle Spatial

Oracle Spatial is an integrated set of functions and procedures that enables spatial data to be stored, accessed, and analyzed quickly and efficiently in an Oracle database. Oracle Spatial provides a SQL schema and functions that facilitate the storage, retrieval, update, and query of collections of spatial features in an Oracle database. Spatial consists for example of the following [4]:

- A schema (MDSYS) that prescribes the storage, syntax, and semantics of supported geometric data types
- A spatial indexing mechanism
- Operators, functions, and procedures for performing area-of-interest queries, spatial join queries, and other spatial analysis operations
- Functions and procedures for utility and tuning operations
- Topology data model for working with data about nodes, edges, and faces in a topology [3]

3.1 Object-relational data model

Spatial supports the object-relational model for representing geometries. This model stores an entire geometry in the Oracle native spatial data type for vector data, `SDO_GEOMETRY`. An Oracle table can contain one or more SDO_GEOMETRY columns. The object-relational model corresponds to a "SQL with Geometry Types" implementation of spatial feature tables in the Open GIS ODBC/SQL specification for geospatial features. The benefits provided by the object-relational model include [4]:

- Support for many geometry types, including arcs, circles, compound polygons, compound line strings, and optimized rectangles.
- Ease of use in creating and maintaining indexes and in performing spatial queries.
- Geometries modeled in a single row and single column.
- There spatial data can by associated with Cartesian, Geodetic, Projected and Local coordinate system in Spatial. There we can find many functions for the manipulation with individual coordinate systems and their transformation in Spatial but Spatial does not support S-JTSK coordinate system in 10g Release 2 version.

3.2 Topology data model

The topology data model of Oracle Spatial lets work with data about nodes, edges and faces in a topology. We can store information about topological elements and geometry layers in Oracle Spatial tables and metadata views. We can then perform certain Spatial operations referencing the topological elements [3].

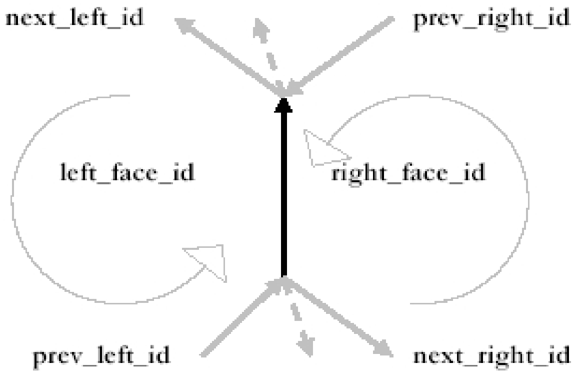

Fig. 2. Winged-edge structure in Oracle Spatial

As we can see at figure 2 topology data model works with oriented edges and references on the most neighbouring edges. Next there are references on the left and right face for each edge. For storing all needed topology information we need three tables, as it is shown in the figure 3.

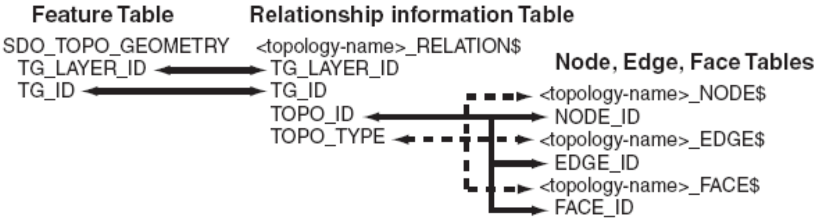

Fig. 3. Mapping between feature table and tables with topology primitives [3]

There is the relationship information table connecting information in a feature table with information in its associated node, edge or face table.

4 Conversion between data models

As it was mentioned in this paper above, the actually data model of cadastral information system is not suitable for WFS purposes. Two proposed solutions were tested – the first one – to store geometry of features using the `SDO_GEOMETRY` data type and the second one – to store geometry of features using the winged–edge structure. To be able make some tests of the new solutions I proposed and implemented software for conversion between the actually data model and the new proposed models. This software was created in Java programming language. Software is able to load data from the text file containing data as a result of SQL query which select needed data from the actually production database. The name of the software is `DKM2Spatial`. DKM means Digital Cadastral Map[2]. As we can see from the name of the software we are able to transform digital cadastral map which makes the crucial part of CIS. Useful sources for this work were [6] and [7] but the situation in the Czech cadastre is different. It must be considered the fact that we have to consider among point and lines also circles, circular arcs and curves.

4.1 Software DKM2Spatial

The program is simply triggered from the command line:

```
java -jar DKM2Spatial.jar –data_model
```

```
where the choice –data_model is:
```

- `g – for transformation of source data to the object-relational model`
- `t – for transformation of source to the winged-edge structure`
- `a – for transformation of source data to the both data models`

The output is the text file containing SQL DML commands `INSERT` and the data prepared for their loading to the testing database.

4.2 Loading data to the object-relational data model

The program `DKM2Spatial` is able to process input data and as a result to give the text files with the data prepared for their loading to the testing database, concretely to the object-relational data model based on the `SDO_GEOMETRY` object data type.

On the side of the testing database we have first of all to propose data model for storage of transformed data. For simplicity suppose that we will have only two tables – table `PARCELS` containing boundaries of parcels plus identification attributes and the table `DEFINITION_POINTS` containing the numbers and the geometries of definition points of parcels. Next, each parcel has just only one

[2] Czech equivalent of Digital Cadastral Map is Digitální Katastrální Mapa.

definition point. Each parcel boundary (and the location of definition point as well) is stored by using of the SDO_GEOMETRY object data type.

The content of the output file looks like this (for parcels):

```
INSERT INTO PARCELS VALUES(
  4966580301, -- Parcel ID
  81450301,   -- Definition point ID
  6,
  3,
  13,
  600032,
  2,
  MDSYS.SDO_GEOMETRY( -- geometry of cadastral boundary
    2003,
    NULL,
    NULL,
    SDO_ELEM_INFO_ARRAY(1,1003,1),
    SDO_ORDINATE_ARRAY(1163850990,750699120,
                       1163851750,750701170,
                       1163853560,750704580,
                       ...
                       1163850990,750699120)

  )
);
```

Such structured data we are able to load to the testing database, concretely to the PARCELS table. For definition points the situation is quite similar. When the data are loaded in the testing database we must perform next necessary steps as is described in detail in [4]. It means concretely to update metadata tables, create a spatial index and in the last step to check the data validity. The R-tree structure [1] is supported by Oracle Spatial 10g Release 2 for spatial data indexing. Finally, after making all previous steps we are able to visualize the data. Figure 4 shows one cadastral territory in which boundaries of parcels are stored by using the SDO_GEOMETRY object data type (table PARCELS).

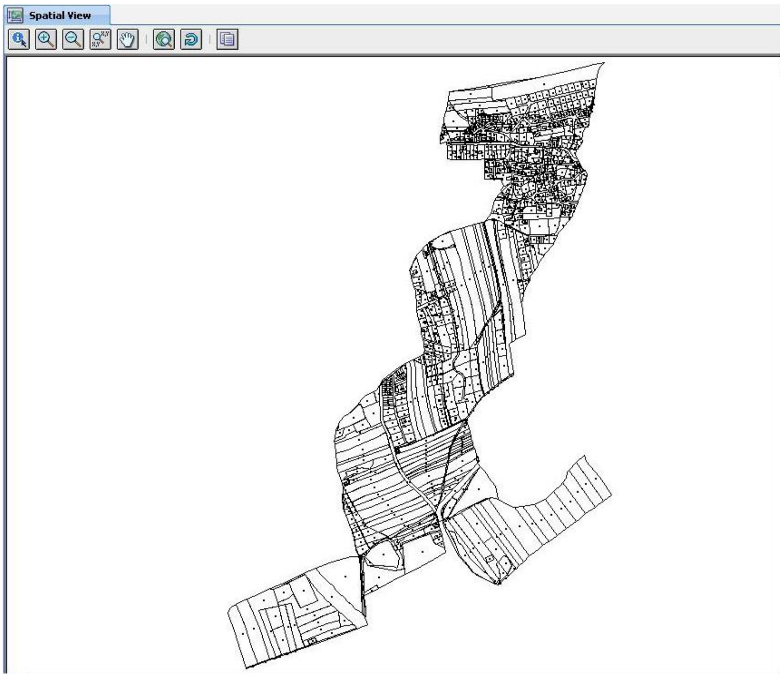

Fig. 4. Visualization of parcels which boundaries are stored by using the SDO_GEOMETRY data type

4.3 Loading data to the winged-edge structure

The software is also able to prepare data in specified form for their inserting to the edge, node and face tables because for using the Spatial topology capabilities, we must first insert data into special edge, node and face tables.

The edge, node and face tables are described in detail in [3]. These tables are created by Spatial when we create a topology. The data to the edge, node and face tables were loaded using SQL `INSERT` statements (this is the output of run of the `DKM2Spatial` program):

```
INSERT INTO cadastre_edge$ VALUES(
  6811226301,  -- Edge ID
  10532960301, -- ID of the start point
  10532961301, -- ID of the end point
  6811224301,  -- next_left edge
  6811228301,  -- prev_left_edge
 -6811228301,  -- next_right edge
 -6811224301,  -- prev_right_edge
  4966587301,  -- face on the left side
  4966585301,  -- face on the right side
  SDO_GEOMETRY( -- geometry of edge
               2002,
               NULL,
               NULL,
               SDO_ELEM_INFO_ARRAY(1,2,1),
               SDO_ORDINATE_ARRAY(1163964810,750670440,
                                  1163957780,750670900)
  )
);
```

As we can see from the example above all needed references on the most neighbouring edges had to be computed. This part was maybe most difficult to implement. Next feature tables were created. For simplicity we will have only two feature tables – table `TOPO_PARCELS` containing the attribute `boundary` of the `SDO_TOPO_GEOMETRY` data type plus identification attributes and the table `TOPO_DEFINITION_POINTS` containing the number and the attribute `boundary` of the `SDO_TOPO_GEOMETRY` data type containing location of definition point. Next, each parcel has just only one definition point. Each parcel boundary (and the location of definition point as well) is stored by using of the `SDO_TOPO_GEOMETRY` object data type. Created feature tables were associated with the topology. This caused the `(<topology-name>_RELATION$` table was created – in our case concretely the `CADASTRE_RELATION$` table. Spatial automatically maintains a relationship information `CADASTRE_RELATION$` table for each topology. Topology metadata are initialized by using the `SDO_TOPO.INITIALIZE_METADATA` [3] procedure. This procedure also creates spatial indexes on the `<topology-name>_EDGE$`, `<topology-name>_NODE$` and `<topology-name>_FACE$` tables. Next to last step is the load of data to the feature tables. After it we can already query the topology data. In figure 5 we can see a detail of cadastre territory which is stores by using the winged-edge structure. In figure 6 is simplified scheme of testing database.

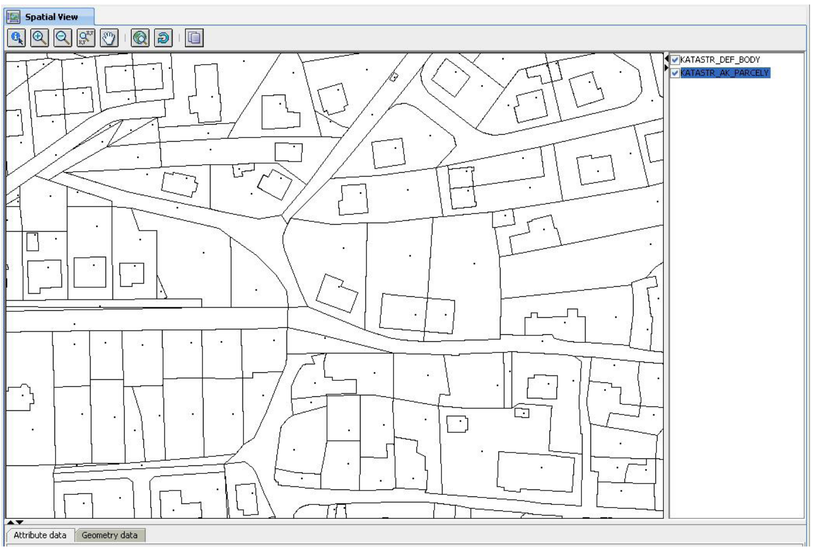

Fig. 5. Visualization of parcels which boundaries are stored by using the winged-edge structure

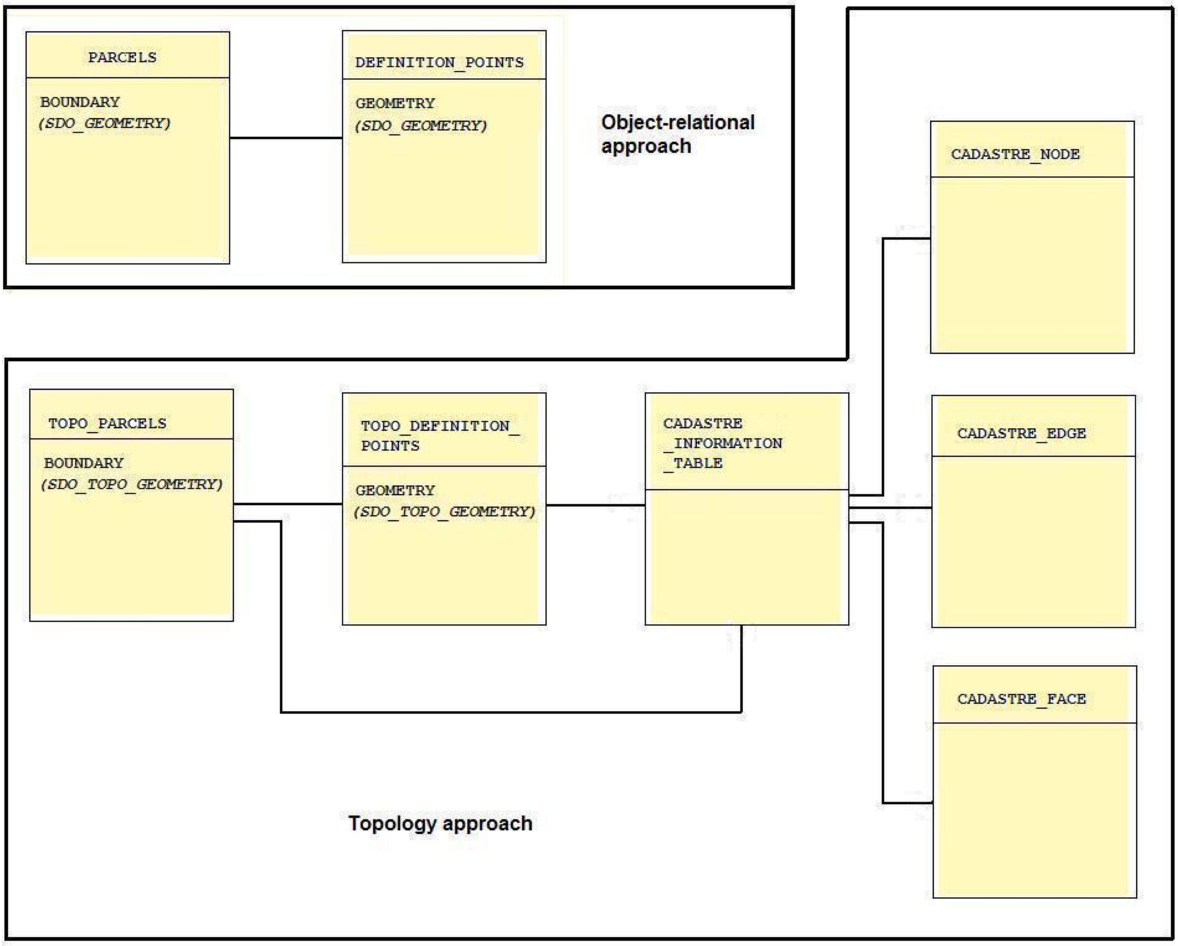

Fig. 6. Simplified data model of testing database

5 Testing Oracle Spatial for cadastral data

In this chapter will be described the testing of requirements on the disk space needed for storage of our data sample and the time needed for spatial query evaluation. It was expected that the object-relational data model will be more appropriate for our purpose – to provide data via WFS. We need to retrieve requested data from database very quickly. As regards a disk space needed for the sample data storage I think that in this time it is not so much important. So the preferred criterion is the time needed for evaluation of spatial queries.

5.1 Storage requirements

In table 1 is mentioned how many disk space (in kilobytes) require single tables.

Table 1. Disk space needed for single tables.

Table	Disk space [KB]
PARCELS	295
DEFINITION_POINTS	30
TOPO_PARCELS	311
TOPO_DEFINITION_POINTS	49
CADASTRE_EDGE$	1070
CADASTRE_FACE$	236
CADASTRE_NODE$	219
CADASTRE_RELATION	16

It stands to reason (see numbers in table 1) that preferable way for storage of data is the using of the object-relational data model (`SDO_GEOMETRY` object data type for storage of geometries). In the case of the testing data was required approximately six times less of disk space for the `SDO_GEOMETRY` object data type (tables `PARCELS` and `DEFINITION_POINTS`). Of course, we have also to consider the disk space for indexes. Here the situation is also better for the `SDO_GEOMETRY` data type as is evidently from the values in table 2.

Table 2. Disk space needed for single indexes.

Index	Table/Attribute	Disk space [KB]
PARCELS_SPATIAL_IDX	PARCELS/BOUNDARY	295
DEF_POINTS_SPATIAL_IDX	DEFINITION_POINTS/GEOMETRY	30
CADASTRE_ED_SIDX	CADASTRE_EDGE$/GEOMETRY	1070
CADASTRE_FC_SIDX	CADASTRE_FACE$/GEOMETRY	236
CADASTRE_ND_SIDX	CADASTRE_NODE$/MBR_GEOMETRY	219

5.2 Spatial queries evaluation

The decision which data model (object-relational versus topology data model) to use for the Publication database depends on the time needed for the spatial query evaluation. I carried out test with query window of various sizes. The decision to use the object-relational data model for Publication database was made on the base of resulting times. Concrete values are mentioned in table 3.

Table 3. Time needed for the spatial query evaluation.

Query Window [m]	SDO_GEOMETRY	SDO_TOPO_GEOMETRY
10x10	0.016 (1)	0.125 (3)
50x50	0.032 (12)	0.140 (12)
100x100	0.032 (27)	0.141 (27)
250x250	0.047 (146)	0.297 (146)
500x500	0.047 (486)	0.375 (486)

1000x1000	0.047 (690)	1.016 (690)

We can see that for the SDO_GEOMETRY object data type is required a much less time for spatial query evaluation than for the SDO_TOPO_GEOMETRY data type. I also tested the function-based index over the attribute BOUNDARY in the TOPO_PARCELS table but the results were not so much good as in the case of the object-relational data model. This test with the function-based index was inspired by [5].

6 Conclusions

The aim of the presented paper was to decide which data model to choose for the Publication database which is presently created at the Czech Office for Surveying, Mapping and Cadastre in Prague. This database will serve for INSPIRE related activities. It was shown that the actual data model is not suitable for such activities. Therefore it was decided to store data in some other more appropriate structure. On the basis of results stated in this paper the object-relational data model is used in the Publication database. It means that spatial features are stored as objects inside this database.

Reference

[1] GUTTMAN, A. *R-trees: a dynamic index structure for spatial searching*. In Proceedings of the ACM SIGMOD international konference on Management of data, 1984, pp. 47 – 57 ISBN: 0-89791-128-8.

[2] JANECKA, K. *Modelling of geospatial data base on the data model of cadastre level*. Diploma Thesis. 2005, Pilsen.

[3] MURRAY, Ch. *Oracle Spatial Topology and Network Data Model, 10g Release 2*. Oracle. 2005.
Oracle. http://www.oracle.com/technology/documentation/spatial.html

[4] MURRAY, Ch. *Oracle Spatial User's Guide and Reference, 10g Release 2*. Oracle. 2003.
Oracle. http://www.oracle.com/technology/documentation/spatial.html

[5] OOSTEROM, P., STOTER, J., QUAK, W., ZLATANOVA, S. *The Balance Between Geometry and Topology*. In Proceedings of the ISPRS Technical Commission IV Symposium, 2002, Canada. Vol. XXXIV, part 4, ISSN 1682-1750.

[6] OOSTEROM, P., TIJSSEN, T., PENNINGA, F. *Topology storage and use in the context of consistent data management*. GISt Report No. 33. 2005, TU Delft. 54 p.

[7] PENNINGA, F. *Oracle 10g Topology; Testing Oracle 10g Topology using cadastral data*. GISt Report No. 26. 2004, Delft. 48 p.