

Advanced Applications of Spatial Database for Geographical Information System (GIS)

Mr.M.Jayakameswaraiah, Dr.S.Ramakrishna

Abstract— The drastic ascent in the volume of spatial data owes its growth to the technical advancements in technologies that aid in spatial data acquisition, mass storage and network interconnection. Thus the necessity for automated detection of spatial knowledge from voluminous spatial data arises. This paper explains the concepts of Oracle spatial as a way to store and analyze spatial data. Oracle spatial is designed to make spatial data management easier and more natural to users of location-enabled applications and Geographic Information System (GIS) applications. Once spatial data is stored in an Oracle database, it can be easily manipulated, retrieved, and related to all other data stored in the database. The data that indicates the Earth location (such as longitude and latitude) of these rendered objects is the spatial data. When the map is rendered, this spatial data is used to project the locations of the objects on a two-dimensional piece of paper. A GIS is often used to store, retrieve, and render this Earth-relative spatial data. Types of spatial data (other than GIS data) that can be stored using Oracle spatial include data from Computer Aided Design (CAD) and Computer Aided Manufacturing (CAM) systems.

Index Terms—oracle spatial; Spatial Database; Geographical Information System; object relational model; spatial query; Web Data model.



1 INTRODUCTION

The advancements in scientific data collection have led to huge and continuously ascending amount of data making it unfeasible to be interpreted manually. Hence, the development of new techniques and tools in support of humans, aiding in the transformation of data into beneficial knowledge has been the center of the comparatively new and interdisciplinary research area called the “knowledge discovery in databases (KDD)”. Classical database techniques used for spatial data X and Y columns from various geometries, but this method allows only limited indexing and querying and the database access and data manipulation is difficult. Oracle spatial in Oracle Database10g includes native data types for storing vector, raster, and persistent topology data. This document outlines some best practices when working with Oracle’s native vector data type, SDO_GEOMETRY, in Oracle spatial for Oracle Database 10g. Some of the major GIS vendors have legacy proprietary alternatives to store vector data in Oracle, for example, in Oracle’s LONG RAW data type. These legacy alternatives were introduced prior to the availability of the SDO_GEOMETRY data type. In the marketplace, there is sometimes the misconception that using SDO_GEOMETRY as the underlying vector data store may compromise features or performance in products offered by the major GIS vendors. Oracle spatial performance may far exceed 15% better

performance over proprietary “LONG RAW” data type solutions[1]. This is because Oracle does not support table partitioning on tables that contain LONG or LONG RAW columns. Tables with SDO_GEOMETRY columns can leverage Oracle table partitioning, which can significantly help performance, scalability, and manageability. Most business information has a location component, such as geocoded customer addresses, sales territories, and wireless service boundaries. Businesses can take advantage of their geographic information by incorporating location analysis into their information systems.

2 ORACLE SPATIAL

The geospatial data features support complex geographic information systems (GIS) applications, enterprise applications, and location-based services applications, augmenting the Oracle Database Locator feature, which provides storage, analysis, and indexing of 2D location data accessible through SQL and standard programming languages. 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 of the following [2]:

- (1) A schema (MDSYS) that prescribes the storage, syntax, and semantics of supported geometric data types;
- (2) A spatial indexing mechanism;
- (3) Operators, functions, and procedures for performing area-of-interest queries, spatial join queries, and other spatial analysis operations;
- (4) Functions and procedures for utility and tuning operations;

- Author: Mr.M.Jayakameswaraiah, Research Scholar, Department of Computer Science, Sri Venkateswara University, Tirupati, Andhrapradesh, India.E-mail: mjayakameswaraiah@gmail.com
- Co-Author: Dr.S.Ramakrishna, Professor, Department of Computer Science, Sri Venkateswara University, Tirupati, Andhrapradesh, India E-mail: drsramakrishna@yahoo.com

(5) Topology data model for working with data about nodes, edges, and faces in a topology Network data model for representing capabilities or objects that are modeled as nodes and links in a network;

(6) GeoRaster, a feature that lets you store, index, query, analyze, and deliver GeoRaster data, that is, raster image and gridded data and its associated metadata;

(7) The spatial component of a spatial feature is the geometric representation of its shape in some coordinate space. This is referred to as its geometry.

3 OBJECT RELATIONAL MODEL

The ORM (Object-Relationship Model) is the "static" part of OSA (Object Oriented System Analysis). That is, the ORM describes the "database" of a model: what objects may exist, what object sets they belong to, and what relationships exist between objects.

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 [5]. The benefits provided by the object-relational model include:

(1) Support for many geometry types, including arcs, circles, compound polygons, compound line strings, and optimized rectangles;

(2) Ease of use in creating and maintaining indexes and in performing spatial queries;

(3) Indexing is maintained by the Oracle database server;

(4) Geometries modeled in a single column;

(5) Performance is improved.

4 THE WEB GIS ARCHITECTURE

The architecture of Web-GIS is composed of several key functional parts, which are abstracted based on communication platform, system service providing by Internet computing environment, functions of GIS and data management. Every part includes several components which implement certain function.

At present, there are two tier and three tier client/server two computing mode. Two tier client/server architecture is formed by server side software and client side software[9]. Three tier client/server Web GIS architecture is divided into three levels from its function logic, like in Fig. 1.

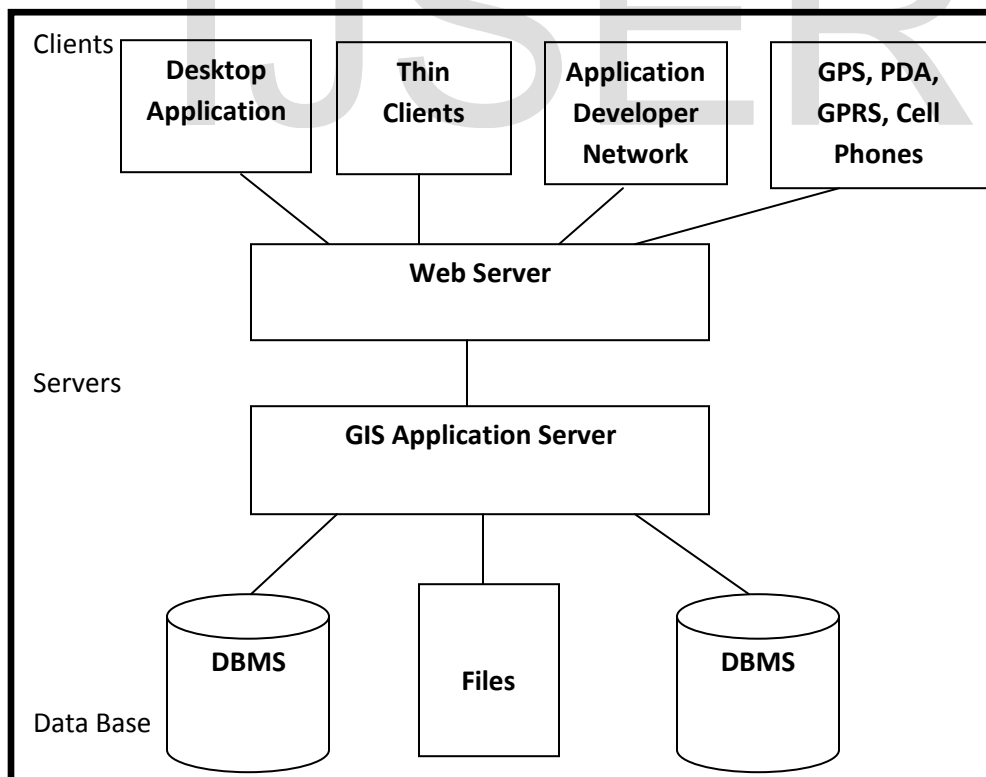


Fig.1.Web GIS Architecture

The first level is user interface, offers user interface function; the second level is GIS transaction processing logic level offers accomplishment all of business logic and data access function; the third level is GIS spatial data storage level offers functional data continuous data storage function. Servers level offers service for GIS data, it logically isolates client side from data level, thereby, it reduces the client site complexity of data access, at the same time, it enhances the security of database.

Traditionally, tools devoted to the management of geographic data use client-server architectures composed of desktop clients and either conventional or spatial Database Management Systems (DBMS) at their servers. The architecture of the system is extensible in the sense that it can be adapted to the particular characteristics of a wide variety of heterogeneous mobile devices.

5 THE WEB DATA MODEL

The spatial data model is a hierarchical structure consisting of elements, geometries, and layers. Layers are composed of geometries, which in turn are made up of elements. An element is the basic building block of geometry. The supported spatial element types are points, line strings, and polygons. For example, elements might model star constellations (point clusters), roads (line strings), and county boundaries (polygons). Each coordinate in an element is stored as an X,Y pair. The exterior ring and the interior ring of a polygon with holes are considered as two distinct elements

6 DATABASE OBJECTS AND SPATIAL QUERY

Oracle spatial can store and analyze four-dimensional data; however, for the sake of clarity, this article discusses only two-dimensional. The basic building block in Oracle spatial is an element: a Point, a LineString, or a Polygon.

that together make up a complex polygon. A geometry (or geometry object) is the representation of a spatial feature, modeled as an ordered set of primitive elements. A geometry can consist of a single element, which is an instance of one of the supported primitive types, or a homogeneous or heterogeneous collection of elements. A multipolygon, such as one used to represent a set of islands, is a homogeneous collection. An example of a geometry might describe the buildable land in a town. This could be represented as a polygon with holes where water or zoning prevents construction. A layer is a collection of geometries having the same attribute set. For example, one layer in a GIS might include topographical features, while another describes population density, and a third describes the network of roads and bridges in the area (lines and points). The geometries and associated spatial index for each layer are stored in the database in standard tables. A coordinate system (also called a *spatial reference system*) is a means of assigning coordinates to a location and establishing relationships between sets of such coordinates. It enables the interpretation of a set of coordinates as a representation of a position in a real world space. Any spatial data has a coordinate system associated with it. The coordinate system can be *georeferenced* (related to a specific representation of the Earth) or not georeferenced (that is, Cartesian, and not related to a specific representation of the Earth). When performing operations on geometries, Spatial uses either a Cartesian or curvilinear computational model, as appropriate for the coordinate system associated with the spatial data.



Oracle spatial defaults to a Cartesian coordinate system with dimensionless units. For large geodetic data sets (such as

maps of the earth's surface), a geodetic coordinate system is needed. This coordinate system uses longitude and latitude for the ordinates and automatically handles issues with the curvature of the earth's surface. Another piece of information needed to map is the accuracy of ordinates. Spatial data almost always has an associated tolerance or error associated with it. However, Oracle spatial defaults to a tolerance of zero[3]. With a coordinate system and tolerance, it's possible to query and map spatial data correctly. For the map show in Fig. 1 will be present an example of creating a spatial database structure, a data line inserting and a spatial query. To create a table called street with an indexed spatial column, the following SQL statements are needed:

```
CREATE TABLE street (  
    street_id NUMBER PRIMARY KEY,  
    street_name VARCHAR2(32),  
    shape MDSYS.SDO_GEOMETRY);  
INSERT INTO user_sdo_geom_metadata VALUES (  
    'street',  
    'shape',  
    MDSYS.SDO_DIM_ARRAY(  
        MDSYS.SDO_DIM_ELEMENT('X', 0, 100, 0.05),  
        MDSYS.SDO_DIM_ELEMENT('Y', 0, 100, 0.05)),  
    NULL);  
CREATE INDEX street_idx ON street (shape);
```

The first statement creates the desired table street with a column named shape to hold the spatial data. The second statement tells Oracle spatial about the spatial data in the street table. The user_sdo_geom_metadata table describes the coordinate system and tolerance for each spatial column that a user owns.

The table takes four values:

- (1) The table name that has a spatial column,
- (2) The column name for the spatial data,
- (3) An array describing the minimum, maximum, and tolerance value for each dimension, and a number stating the coordinate system. street uses two-dimensional data that can range from 0 to 100 units with a tolerance of 0.05.
- (4) The null value for the coordinate system tells Oracle spatial to use the default Cartesian coordinate system.

The last statement creates the spatial index street_idx, which is required for spatial queries.

Oracle spatial uses the MDSYS.SDO_GEOMETRY type to store spatial data, which is defined as:

```
CREATE TYPE sdo_geometry AS OBJECT (  
    SDO_GTYPE NUMBER,  
    SDO_SRID NUMBER,
```

```
    SDO_POINT SDO_POINT_TYPE,  
    SDO_ELEM_INFO  
    MDSYS.SDO_ELEM_INFO_ARRAY,  
    SDO_ORDINATES  
    MDSYS.SDO_ORDINATE_ARRAY);
```

SDO_GTYPE is a number that defines the overall shape; that is, a line, a sphere, and so forth. It describes the end result of the ordered combination of elements. SDO_GTYPE is a four-digit integer. The first digit represents the number of dimensions. The second digit represents the linear representation, which is important for a three- or four-dimensional shape. In a two-dimensional shape the value is zero.

The SQL command for insert a street in database is:

```
INSERT INTO street  
VALUES(  
    10, -- feature id  
    'Traian Street', -- name  
    MDSYS.SDO_GEOMETRY(  
        2006,  
        NULL,  
        NULL,  
        MDSYS.SDO_ELEM_INFO_ARRAY(1,1003,1),  
        MDSYS.SDO_ORDINATE_ARRAY(2,1, 6,2, 6,5,  
        9,7, 8,9, 5,7, 5,4, 1,3, 2,1) ));
```

To obtain an exact query, SDO_FILTER and SDO_RELATE functions must be executed. SDO_RELATE looks at two geometries and determines if they interact in a specified way. It is important to note that SDO_RELATE only works on two-dimensional data[5]. It is defined below:

```
SDO_RELATE(  
    geometry1 MDSYS.SDO_GEOMETRY,  
    geometry2 MDSYS.SDO_GEOMETRY,  
    params VARCHAR2)
```

SDO_FILTER arguments are the same as those for SDO_RELATE with the exception of the last argument. The parameters argument has a mask type value in addition to the query type value. To select all objects that are inside a defined rectangle the following spatial query would work:

```
SELECT street_name FROM street s  
WHERE street_id = 12  
AND SDO_FILTER (s.shape,
```

```
mdsys.sdo_geometry(2003,NULL,NULL,  
mdsys.sdo_elem_info_array(1,1003,1),  
mdsys.sdo_ordinate_array(2,1, 6,2, 6,5, 9,7, 8,9, 5,7,  
5,4, 1,3, 2,1)),  
'querytype=WINDOW') = 'TRUE'  
AND SDO_RELATE(s.shape,  
mdsys.sdo_geometry(2003,NULL,NULL,  
mdsys.sdo_elem_info_array(1,1003,1),  
mdsys.sdo_ordinate_array(2,1, 6,2, 6,5, 9,7, 8,9, 5,7,  
5,4, 1,3, 2,1)),  
'masktype=INSIDE
```

```
querytype=WINDOW')='TRUE'
```

There are many more query operators and functions in Oracle spatial.

7 CONCLUSIONS

Oracle Spatial can be used to design very powerful and robust spatial applications. It was not possible in case of traditional database to store, analyze, visualize, and integrate spatial data in spatial applications. Oracle spatial makes it possible to combine the relational power of a database with spatial data. The ability to use indexes, various queries, and functions means complex spatial calculations may be pushed back onto large database servers. As mobile applications and technologies increase, so will the demands to store and analyze spatial data in a transactional setting. Oracle spatial are powerful core features of the Oracle database.

For complex GIS applications Oracle spatial provides the required functionality, like spatial functions (including area, buffer, centroid calculations), advanced coordinate systems support, linear referencing systems, and aggregate functions. Oracle spatial provides performance, ease of use, and an architecture that scales to meet the needs of Internet and wireless solutions. This technical document describes some best practices, tips and general information that can help utilize Oracle spatial to increase productivity, decision support, and cost savings in your everyday business practices.

REFERENCES

- [1] Bauer M. *Mapping Geometric Data with Oracle Spatial*, O'Reilly Network, 2003.
- [2] Hall B. *Oracle Spatial and Database Tuning*, Oklahoma, 2007.
- [3] Kothuri R., Beinat E., Godfrind A. *Pro Oracle Spatial*, Apress L. P. Hardcover, 2004
- [4] Tomlinson, Roger F. *Thinking About GIS: Geographic Information System Planning for Managers*. Environmental Systems Research Institute, Inc., Redlands, CA, USA, 2003.
- [5] Ravi Kothuri, Albert Godfrind and Euro Beinat "Pro Oracle Spatial for Oracle Database 11g", Apress publishers, 2007.
- [6] "Advance Spatial Data Management for Enterprise Applications", September, 2009. [Online]. Available:
- [7] Heechae Jin, Open GIS: The Interoperability of Spatial Data, Journal of Information Science, Vol. 16 No.3, 23-27, 1998.
- [8] Pedersen, T. B. and Tryfona, N. Pre-Aggregation in Spatial DataWarehouses. In *Proceedings of the Seventh International Symposium on Spatial and Temporal Databases*, to appear, 2001.
- [9] Xiong hanjiang, Gong jianya. The Design of Three tier Client/Server Web GIS [J], Journal of Wuhan University Science Information Version. Vol 26(2), 2000.
- [10] http://www.oracle.com/technology/products/spatial/pdf/11gr2_collateral/spatial11gr2_wp_0922.pdf