GEO-DBMS : DATA MANAGEMENT OF 3D SPATIAL OBJECTS

Harith Fadzilah Abd Khalid, Nor Suhaibah Azri and Alias Abdul Rahman

Department of Geoinformatics,
Faculty of Geoinformation Science and Engineering,
Universiti Teknologi Malaysia, 81310 UTM Skudai, Johor, Malaysia.
E-mail: hfadzilah2@siswa.utm.my, alias@fksg.utm.my, norsuhaibah@gmail.com

Abstract

This paper describes one aspect of 3D GIS research that is the data management of 3D objects. Current geo-database can manage large spatial data sets and able to provide access to multi users. Lately there are growing tendency towards the multifunctional use of space i.e. above and below the ground surface. Applications such as urban planning, telecommunication and 3D cadastre depend on 3D data. Thus, the needs for managing the 3D spatial datasets especially for large area are inevitable. Besides this, data acquisition technique has also motivates us for a more practical data structure(s) towards a data model for 3D GIS. A discussion on how to establish these data structures for managing 3D spatial objects form major discussion of this paper. 3D primitives like points, lines, polygons and solids are part of the primitives. The mapping of this structure in Oracle Spatial will be highlighted as well. Since geo-spatial database can handle large amount of data, we propose a practical data management and test by using real world datasets. In the experiment, the management of the 3D objects and appropriate indexing method is also discussed. Our experiment will only use simple 3D objects and will be visualized through other application software such as Autodesk Map3D or Bentley MicroStation. Simple query within the SQL for the 3D objects is also possible. The paper also will highlight the future outlook of the data structuring approach for managing large volume of 3D spatial objects and relates with current research development in 3D GIS.

KEYWORDS: Data structures, Data management, 3D spatial objects, Geo-DBMS, 3D GIS

1.0 INTRODUCTION

Since GIS was introduced, it has played a vital role to maintain, manipulate, analyze and presenting information about spatial phenomena. GISs are widely used in many government, business and private activities; which fall into three major categories (Maguire et al. 1991): a) socio-economic applications (e.g., urban and regional planning, cadastrial registration, archaeology, natural resources, market analysis, etc.); b) environmental applications (e.g., forestry, fire and epidemic control, etc.); and c) management applications (e.g., organization of pipeline networks and other services, such as electricity and telephones, real-time navigation for vessels, planes and cars, etc.). The role of GISs in these applications is to provide the users and decision-makers with effective tools for solving the complex and usually ill or semi-structured spatial problems, while providing an adequate level of performance.

The recent advances in geo-processing change the understanding of GIS and become a sophisticated system for handling spatial and thematic information of real world spatial objects. In this regard the Database Management System (DBMS) has becomes an important medium for spatial data storage, maintenance, operation and integration purposes.

Lately the availability of 3D data through digital photogrammetry and fast data acquisition such as Light Detection and Ranging (LIDAR) have made applications related to 3D information such as urban planning, telecommunication and 3D cadastre have increased (Stoter 2004, A. Rahman et.al. 2002). Complex structures have been built, be it on the ground, under the ground or even hanging above the ground. Current 2D system to tackle 3D situation is no longer applicable. But the availability of accepting 3D coordinates for spatial objects which defined by the 2D geometric type in the DBMS made possible to describe 3D points, 3D lines and 3D polygons (Zlatanova, 2006). Managing 3D spatial object in the geo-database for a large area is challenging. This motivates us to embark in this research.

The paper is organized in the following order: first, short discussion for the geo-database. Then, the 3D data model and structured the 3D object in Oracle. The experiment and some discussions presented in section four and the conclusion in section five.

2.0 GEO-DATABASE

Initially DBMS was designed for non-spatial data. Until the last several years, developments of DBMS were significant to cater the spatial data. According to Bruenig et.al.(2004), there are two approaches of development in DBMS linking to spatial data. One of this approach is the 'bottom-up' approach. In this approach, the DBMS supports the spatial data type; meaning that it extends 'low level' DBMS data type and indexes to use them in the upper level of GIS applications. Data analysis on the spatial and non-spatial parts of objects can be executed. More and more commercial DBMSs provide spatial extensions to support spatial objects.

Most current mainstream DBMSs offer spatial data type and spatial functions similar to the OpenGIS Consortium (OGC, 1998) Simple Features Specification for SQL (OGC, 1999). The spatial data stored can be accessed by many front-end engines such as MicroStation Bentley, ArcGIS and Autodesk Map 3D. According to the OpenGIS specifications, the spatial object is represented by two structures, i.e. *geometrical (i.e. simple feature specifications)* and *topological (i.e. complex feature specifications)*. The geometric structure provides direct access to the coordinates of individual objects, while the topological structure encapsulates information about their spatial relationships. Presently, geometrical model has been implemented in mainstream DBMSs.

In other word, a full-fledged DBMS which has capabilities for handling spatial data is called Geo-database or Geo-DBMS (Gunting, 1994). A Geo-DBMS knows primitive and composed geometric data types i.e. point, line and polygon, in the same way as its primitive standard data types such as character, string, integer, real etc. The implementations of spatial data type in mainstream DBMSs are basically 2D and the spatial features are stored in a geometrical model without the internal topology. Topological relationships between geometries can be retrieved by the use of spatial operators.

3.0 3D DATA MODELS

According to Zlatanova, et.al.(2004), there are four 3D data model introduced for modeling 3D objects. i.e. 3D Format Data Structure (3DFDS) model, TEtrahedral Network (TEN) model, Simplified Spatial Model (SSM) and Urban Data Model (UDM). The 3DFDS was introduced by Molenaar (1990) consist of three fundamental levels: *feature* (related to a thematic class), four *elementary objects* (*point*, *line*, *body* and *surface*) and four *primitives* (*node*, *arc*, *face* and *edge*). It has all the necessary data to visualise the geometry of objects but in term of performance issues, it has disadvantages. The second model, TEN introduced by Pilouk (1996) as an alternative to the 3DFDS which have some difficulties in modeling objects with indiscernible boundaries. The model is appropriate for representing irregularities in the real world, such as terrain, soil, air, geological objects, etc. TEN has four constructive objects (*tetrahedron*, *triangle*, *edge*, *node*). The general rule in this model is based on the fact that each node is part of an arc, each arc is part of a triangle and each triangle is part of a tetrahedron. A body object is composed of *tetrahedrons*, a surface object of *triangles*, a line object of *arcs* and a point object of *nodes*. The third model, Simplified Spatial Model (SSM) introduced by Zlatanova (2000). It was designed to serve web-oriented applications where spatial queries need to be visualized on the screen as 3D models. Interestingly it has four simple objects but with two primitives i.e. node and face. The faces represent the 3D object. The last model is the Urban Data Model (UDM) introduced by Coors (2003).

In this research, the SSM model approach is adapted since it does not require full partition of space i.e. all of the objects are embedded in 3D. The model is designed to 1) be appropriate for urban modeling (faces with arbitrary shape and non-partitioned bodies), 2) to provide consistent data for visualization (convex, planar faces without nodes in the interior), and 3) be able to distinguish between a large number of 3D topological relations.

3.1 3D object in DBMS

Based on the SSM model approach, 3D object can be stored in the database. Figure 1 shows the UML class diagram of Simplified Spatial Model (Zlatanova 2000). The data types embedded in 3D space, i.e. point, line and polygon can be represented with their 3D coordinates. In this paper the approach used to represent the 3D objects through a set polygons (with 3D coordinates) composing the objects. This approach also has the advantage of being 'understandable' for all the front-ends (GIS/CAD/AEC) since it is supported by the DBMS.

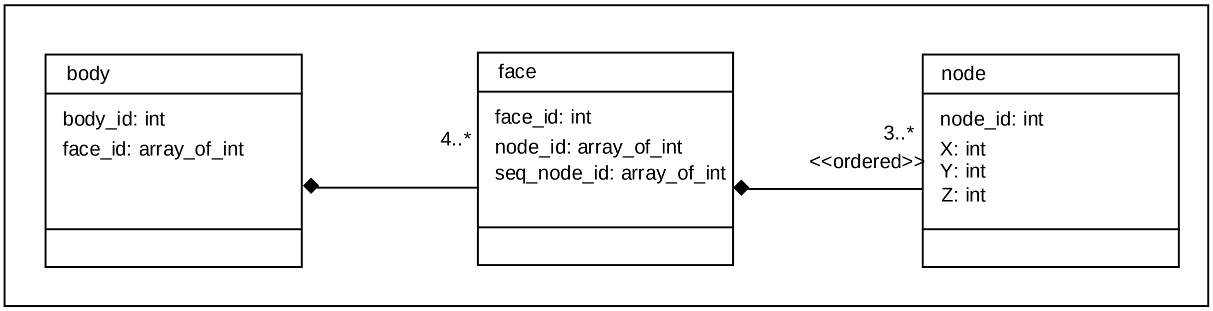


Figure 1: UML class diagram of Simplified Spatial Model (Zlatanova, 2000)

In Oracle Spatial, the geometric description of a spatial object is stored in a user-defined table i.e. in a single row and in a single column of object type SDO GEOMETRY. Any tables that have a column of type SDO GEOMETRY must have another column, or set of columns, that define a unique primary key for that table. Tables of this sort are referred to as geometry tables. Figure 2 shows the SDO GEOMETRY in Oracle Spatial 10g.

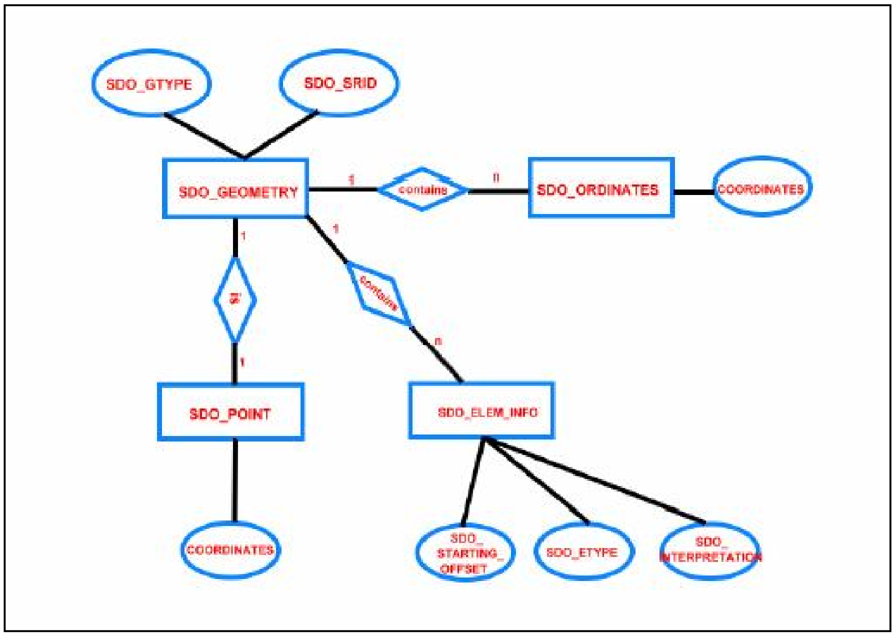


Figure 2: SDO_GEOMETRY in Oracle Spatial

3D object is represented as **polyhedron** (body with flat faces) which is 3D polygon stored as a list of polygons. The face of the object represented by the polygon where the polygon built by the sequence ordered list of X,Y,Z coordinates. Figure 3 shows how the structure is.

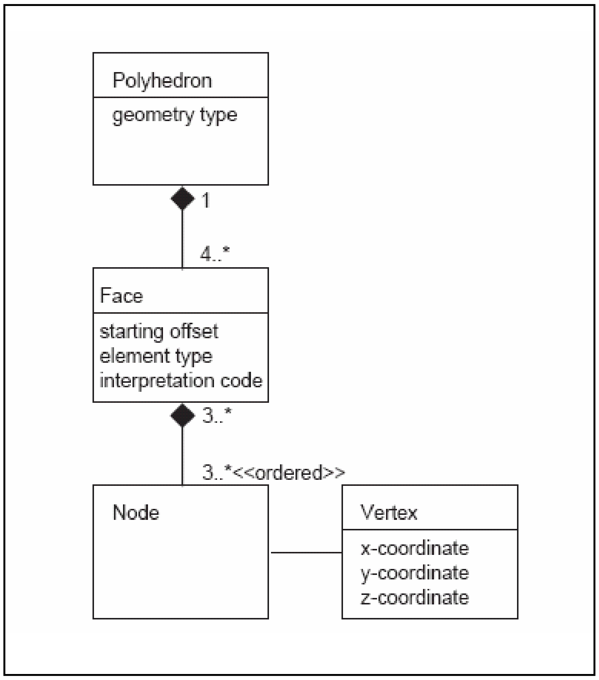


Figure 3: Polyhedron Data Structure

3.2 Spatial Indexing

Spatial index deals with the method of searching, which often involves mathematical algorithms. Spatial indexes are used in DBMS for fast search especially when spatial functions are applied. Without indexing, any searches for a feature would require a sequential scan of every record in the database. Indexing speeds up searching by organizing the data into a search tree that could be quickly traversed to find a particular record. A R-Tree is a depth-balanced tree extending the B-tree for n-dimensions. The index stores the minimum bounding boxes as representations, not the objects themselves. It is equally referred to as a minimum bounding rectangle (MBR). There is no standard syntax/command/structure stated by OGC that enables any DBMS to be implemented. Only the DBMSs themselves provide their own syntax/command/structure that establishes the spatial index. The concept of sample R-tree structure is given in Figure 4 and Figure 5 in two and three-dimensions.

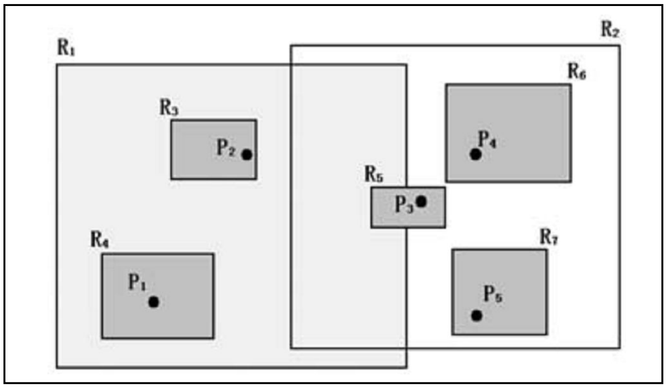


Figure 4: A planar representation of an R-tree

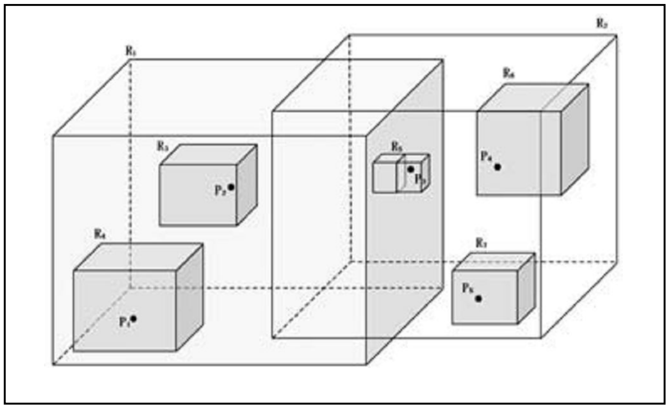


Figure 5: A 3D representation of an R-tree

Oracle Spatial, is able to provide 3D indexing for 3D object (MULTIPOLYGON). For Spatial, the metadata that maintains the lower and upper bounds and tolerance of 3D object needs to be created. Later, a spatial index (R-tree in 3D) could be created on tables to speed up spatial queries. The following example denotes the sample in creating a 3D spatial index within Spatial.

```
-- Inserting metadata for 3D object: MULTIPOLYGON

INSERT INTO user_sdo_geom_metadata VALUES
 (‘Obj3d’, ‘shape’,
 mdsys.sdo_dim_array(
  mdsys.sdo_dim_element(‘X’, 0, 100, 0.1),
  mdsys.sdo_dim_element(‘Y’, 0, 100, 0.1),
  mdsys.sdo_dim_element('Z', 0, 100, 0.1))
 , NULL);

-- Creating 3D Spatial Index
CREATE INDEX Obj3d_I on Obj3d(shape)
  INDEXTYPE IS mdsys.spatial_index
  PARAMETERS('sdo_index_dims=3');  -- Dimension = 3

ANALYZE TABLE Obj3d COMPUTE STATISTICS
```

4.0 EXPERIMENT AND DISCUSSIONS

In the experiment, we start with the buildings and roads within the 'Lingkaran Ilmu' area in University Teknologi Malaysia. Figure 6 shows the conceptual UML class diagram of the database.

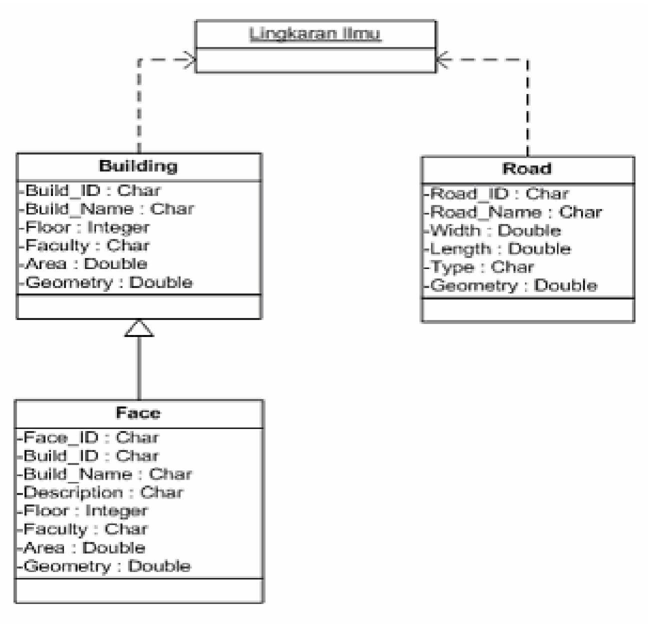


Figure 6: Shows the UML class diagram of the database

The UML diagram shows that the 3D objects (Building) make-up of the combination of faces, where the individual faces being assign with a building *id*. At the other hand, the road considered as a plane 2D objects, which part of Lingkaran Ilmu distinguish objects. Figure 7 shows the buildings which are in 3D and the roads surrounding the area. All the spatial objects stored in Oracle Spatial can be viewed through Autocad Map 3D after the linkage was established. Furthermore, the table of the spatial objects stored can also be displayed in the Autocad Map 3D as in Figure 8.

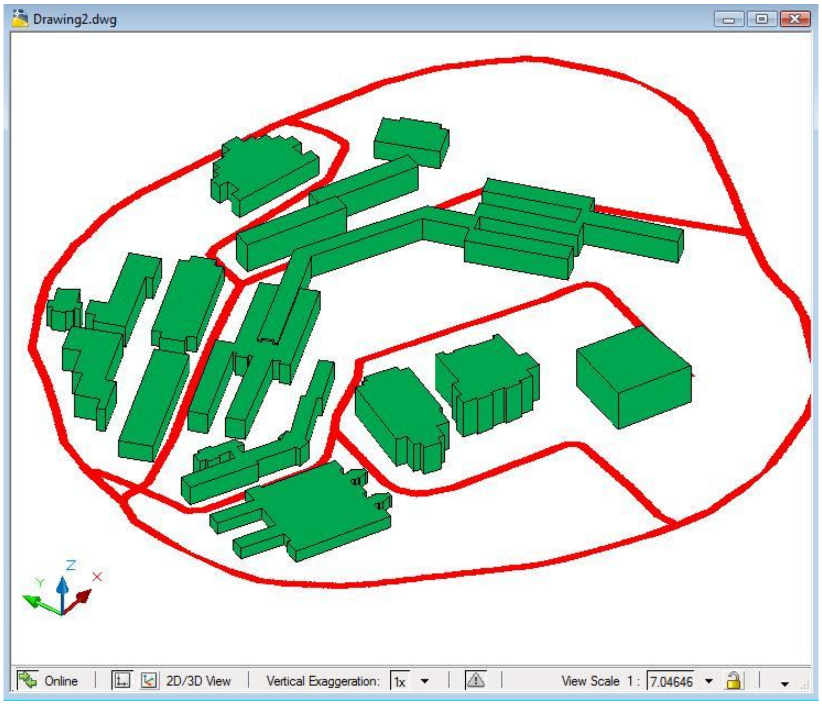
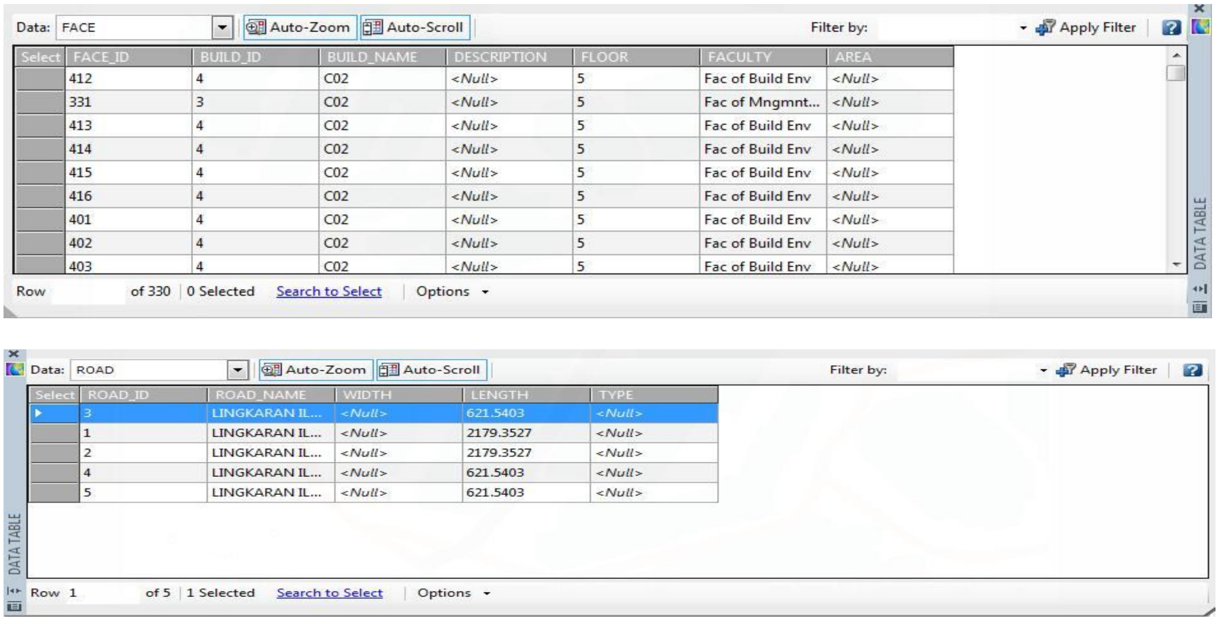


Figure 7: 3D buildings and roads stored in the Oracle Spatial displayed in Autocad Map 3D



Data: FACE

Select	FACE_ID	BUILD_ID	BUILD_NAME	DESCRIPTION	FLOOR	FACULTY	AREA
	412	4	C02	<Null>	5	Fac of Build Env	<Null>
	331	3	C02	<Null>	5	Fac of Mngmnt...	<Null>
	413	4	C02	<Null>	5	Fac of Build Env	<Null>
	414	4	C02	<Null>	5	Fac of Build Env	<Null>
	415	4	C02	<Null>	5	Fac of Build Env	<Null>
	416	4	C02	<Null>	5	Fac of Build Env	<Null>
	401	4	C02	<Null>	5	Fac of Build Env	<Null>
	402	4	C02	<Null>	5	Fac of Build Env	<Null>
	403	4	C02	<Null>	5	Fac of Build Env	<Null>

Row of 330 | 0 Selected

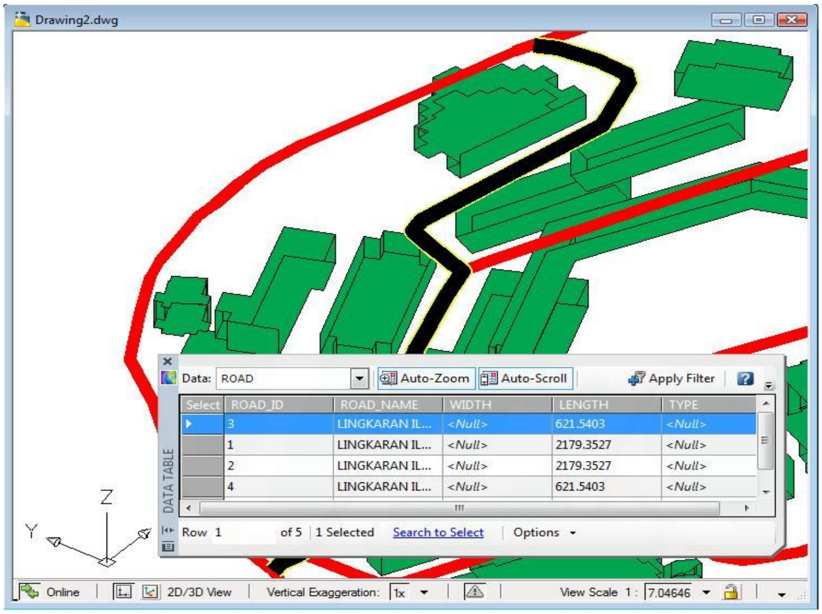
Data: ROAD

Select	ROAD_ID	ROAD_NAME	WIDTH	LENGTH	TYPE
▸	3	LINGKARAN IL...	<Null>	621.5403	<Null>
	1	LINGKARAN IL...	<Null>	2179.3527	<Null>
	2	LINGKARAN IL...	<Null>	2179.3527	<Null>
	4	LINGKARAN IL...	<Null>	621.5403	<Null>
	5	LINGKARAN IL...	<Null>	621.5403	<Null>

Row 1 of 5 | 1 Selected

Figure 8: Buildings and roads table in Oracle Spatial displayed in Autocad Map 3D

In the experiment the building considered as rectangular boxes. The coordinates of the building's foot print generated from the survey plan drawings and the buildings height produced from the architecture building plan. A simple program used to convert these coordinates into the Oracle Spatial SDO GEOMETRY format. The more complicated the building is, the more points needed to represent them, meaning that the coordinates input is enormous. It is also possible to give the real texture for the buildings, since the 3D objects make-up of faces which can contained the real texture as part of their attribute column in the table. A simple SQL query can be carried out as shown in Figure 9.



Data: ROAD

Select	ROAD_ID	ROAD_NAME	WIDTH	LENGTH	TYPE
▶	3	LINGKARAN IL...	<Null>	621.5403	<Null>
	1	LINGKARAN IL...	<Null>	2179.3527	<Null>
	2	LINGKARAN IL...	<Null>	2179.3527	<Null>
	4	LINGKARAN IL...	<Null>	621.5403	<Null>

Row 1 of 5 | 1 Selected

Vertical Exaggeration: 1x

View Scale 1 : 7.04646

Figure 9: Simple SQL query

5.0 CONCLUSION

GIS is changing and becoming an integration of database management, powerful editing and realism visualization inherited from the advanced computing development This experiment shows the combination of software used to stored and visualized 3D objects in the step to realized the 3D GIS. The powerful commercial DBMS which capable of storing enormous of data; in this particular spatial 3D data and the flexibility of the 3D Cad software to view realism is part of the future trends in GIS. In this experiment the buildings can be visualized in 3D. Any attribute concerning the buildings can be attached to via the attribute column in the feature table. SQL query can be done based on the spatial object itself or their attribute. The real look of the buildings is also possible by applying real texture to the building (face) which will be part of this research in future.

Acknowledgement

I would like to convey my deepest acknowledgement firstly to my PhD Research Supervisor, Prof. Dr. Alias Abdul Rahman for his excellent support. He is also the Project Leader for E-Science MOSTI Research Project, Vot No. 79232, entitle Developing 3D Data Management For 3D Spatial Objects in Large Area which cover my expenses for this conference. Secondly, my sincere appreciations to Ministry of Science, Technology and Innovations (MOSTI) for providing us the research grant and last but not least is Universiti Teknologi Malaysia (UTM) for maintaining well services for our research facilities needs.

REFERENCES

Bruenig, M and Zlatanova, S. (2004). *3D-GeoDBMS*, Directions Magazine

Coors, V, (2003). *3D-GIS in networking environments*, In Computers, Environment and Urban Systems, Pergamon. Pp345-357.

Güting, R.H, (1994). *An Introduction to Spatial Database Systems*. VLDB Journal Vol. 3 no. 4, pp357-399

Maguire, David J., Goodchild, M.F, Rhind, D.W (1991). *Geographical Information Systems: Principles and applications.* Longman Group UK

Molenaar, M., (1990). *A formal data structure for 3D vector maps*, in: Proceedings of EGIS'90, Vol. 2, Amsterdam, The Netherlands, pp. 770-781

OGC (1998). *The OpenGIS Guide, Third edition. An introduction to Interoperable Geo-processing*. The OGC Project Technical Committee of OpenGIS Consortium, edited by Buhler and K. McKee, L., Wayland, Mass., USA

OGC (1999). *OpenGIS Simple Features Specification for SQL. Revision 1.1*, OpenGIS Project Document 99-049

Pilouk, M., (1996). *Integrated modelling for 3D GIS*, PhD thesis, ITC, The Netherlands

Rahman, A.A, Zlatanova, S., Pilouk M., (2002). *Trends in 3D GIS Development*. In Journal of Geospatial Engineering, Vol.2, No.2 pp. 1-10.

Stoter, J. (2004). *3D Cadastre*. PhD. Thesis. TU Delft/ Netherlands

Zlatanova, S. (2006). *3D Goemetries in Spatial DBMS*. 3DGeoinfor'06 International Workshop on 3D Geoinformation, Kuala Lumpur

Zlatanova, S. (2000). *3D GIS For Urban Development*. PhD Thesis. University of Graz, Austria.

Zlatanova, S., Rahman, A.A., Wengzhong, S., (2004). *Topological models and frameworks for 3D spatial objects.* In Computers & Geosciences 30 (2004) 419–428