ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

Performance Analysis of Three-Dimensional Objects in Database System as Computer Aided Design

Kazi Shamsul Arefin[1], Prof. Dr. Mohammed Salim Bhuyan[2]

Dept. of Computer Science, American World University, CA, USA (Bangladesh Study Centre) [1,2]

ABSTRACT: Relational database management system has inadequate resource to preserve data in three-dimensional architecture. Besides, three-dimensional scanning is required for an object which needs to deal with very large data sets as resolution is also high. On the other hand, there are a lot of potential applications that would be greatly benefited utilizing three-dimensional data. As there is no currently available three-dimensional data type therefore this research leads to identify the three-dimensional objects and develop the data type accordingly.The purpose of this research is to integrate three-dimensional architecture into relational database management system and hence implementation in computer aided design.

KEYWORDS: Database Management System (DBMS); Three-Dimensional (3D); Computer Aided Design (CAD).

I. Introduction

After the 1970s, there was rapid growth in the use of CADtechnologies by the mass development of silicon chips and the microprocessor, more affordable computers. Today, CADtechnologies became popular in all the major platforms producing consumer products from molded plastics, electronics products:graphics card, high speed CPUsand RAM and many more electronic components. Computers are also being used to control the manufacturing processes. This is because the control data are not based on geometrical parameters.

A. *STATEMENT OF THE PROBLEM*

In case of 2D data, we have two dimensions, such as length and width; however, 3D data have an additional dimension, such as height. In DBMS, height information was always overlooked. With this, the height is approximated and included in calculations for these operations. There are some weaknesses for 3D data model in DBMS:

- DBMS does not differentiate 3D objects
- Inconsistency in continuous flat topology characteristic in GIS systems
- Problematic in the execution of spatial analyses in the GIS systems
- Problematic to provide solution for spatial overlap

B. *BENEFITS OF CAD*

The benefits of CAD include lower product development costs, increased productivity, improved product quality and faster time-to-market.

- Better visualization of product design
- It does speed up the design process
- It offers greater accuracy
- Low risk rate as errors free production
- Complex tasks can be done easily and more robust

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

- Data can bere-using for design
- Money, Time and Manpower all can be saved

C. *BENEFITS OF 3D-DBMS DEVELOPMENT*

There are a lot of advantages utilizing 3D data type in order to preserve 3D information. Now-a-days, some DBMS are trying to preserve 3D information introducing spatial concepts. As mentioned before, depth (Z-coordinate) is ignored or missing in DBMS. Introducing 3D information is a good approach although it is still insufficient. This concept is developed on the basis of geometry models. There are many advantages using 3D DBMS model:

- Avoids redundant storage
- Consistency in data editing
- Easier to maintenance
- Efficient in visualization
- Efficient in query operations
- Natural data model for applications

D. *OBJECTIVES OF THIS RESEARCH*

Some researchers are working on 3D geometry data types. However, there are still not recognized 3D data types by the spatial extent in DBMS. The objectives of this research are as follows:

1. To offer 3D data type in DBMS;
2. To develop a data structure that can handle with large data and offers support for analyzing, validation and querying;
3. Propose the next generation topology in DBMS for CAD.

II. RELATED WORK

All columns of tables havetheir own unique name with appropriate data type. Developer decides what type of data will be stored inside the column based on the attribute. In this section, a table: T_MATRIX_INFO is created with 4 (Four) columns: id, row, column, and value for this research that leads to the 3D data type in section: III.

We gathered some geo information and inserted data (id, row, column and value) into T_MATRIX_INFO. After the execution of 'INSERT' statements, 'COMMIT' command is placed in order to save into database permanently then data availability is cross-checked using 'SELECT' statement.

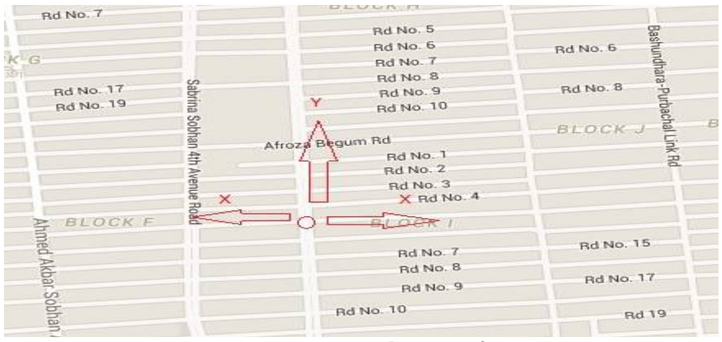


Figure 2.1: Sample 2D information

A. *LIMITATIONS*

Now if we would like to preserve 3D information forCAD (see figure 5.3) in the table: T_MATRIX_INFO then it is not possible in the traditional way. Besides, there is no such data type in oracle database that is able to hold this information. In this case what a traditional DBMS does:

- Additional table will be created to preserve information
- Use referential keys (Foreign key) in order to make join in different tables

Hence, there are some limitations mentioned bellows:

- Consuming more memory (Redundant information)
- Need to join in both table (Primary and Foreign key)
- Perform slower as both not in the same table
- Need to indexing in order to fetching data
- Complexity in maintaining 2 (two) different physical tables
- Complexity in finding related information

III. **THREE-DIMENSIONAL DATA TYPE DESIGN**

A Cartesian coordinate system is a coordinate system that specifies each point uniquely by a pair of numerical coordinates, which are the signed distances to the point from two fixed perpendicular directed lines, measured in the same unit of length. Each reference line is called a coordinate axis or just axis of the system, and the point where they meet is its origin, usually at ordered pair (0, 0). The coordinates can also be defined as the positions of the perpendicular projections of the point onto the two axes, expressed as signed distances from the origin.

Alternatively, the coordinates of a point P can also be taken as the (signed) distances from P to the three planes defined by the three axes. If the axes are named x, y, and z, then the x-coordinate is the distance from the plane defined by the y and z axes. The distance is to be taken with the + or − sign, depending on which of the two half-spaces separated by that plane contains P. The y and z coordinates can be obtained in the same way from the xz- and xy-planes respectively.

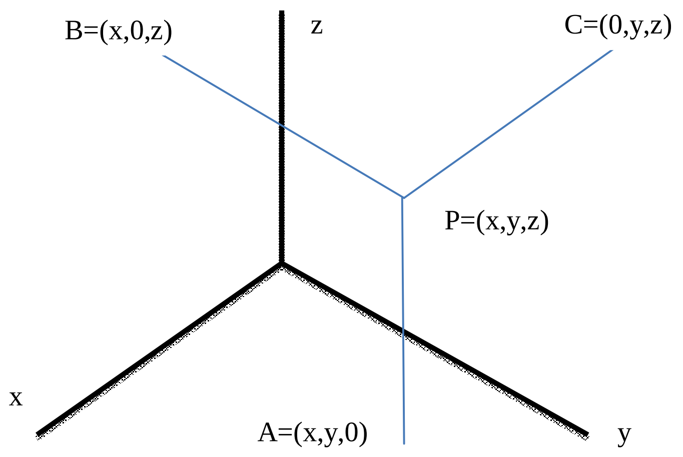


Figure 3.1: 3D (X, Y and Z) co-ordinate information

In the figure 3.1 and 3.2, we are getting 3D (X, Y and Z) co-ordinate information. Therefore, a new data type "OBJ_3D_PLANE" is developed in order to keep 3D information in this section.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

- Step 1: an object is created with 3 (three) attributes: X_AXIS, Y_AXIS and Z_AXIS (properties of these columns are numeric). The name of this object is OBJ_3D_PLANE.

```
CREATE TYPE OBJ_3D_PLANE AS OBJECT (
X_AXIS NUMBER,
Y_AXIS NUMBER,
Z_AXIS NUMBER);
```

- Step 2: a varray data type object: OBJ_3D_DATA_TYPE is created. Where, VARRAY (1048576) is the maximum size. Here, nested (PL/SQL) table can also be used instead of varray.

```
CREATE TYPE OBJ_3D_DATA_TYPE AS VARRAY (1048576) of NUMBER;
```

- Step 3: a 3D object OBJ_3D_DATA is created where it holds 3 objects (NO_OF_DIM NUMBER, OBJ_3D_DATA_INFO OBJ_3D_PLANE and OBJ_3D_DATA OBJ_3D_DATA_TYPE). Now this data type (OBJ_3D_DATA) is able to contain 3D information.

```
CREATE TYPE OBJ_3D_DATA AS OBJECT (
 NO_OF_DIM NUMBER,
 OBJ_3D_DATA_INFO OBJ_3D_PLANE,
 OBJ_3D_DATA OBJ_3D_DATA_TYPE);
```

- Step 4: adding 3D column OBJ_3D_COL into T_MATRIX_INFO which is an object of OBJ_3D_DATA

```
ALTER TABLE T_MATRIX_INFO ADD OBJ_3D_COL OBJ_3D_DATA;

tableT_MATRIX_INFO altered.
```

As 3D column is added, now updating information into the newly added OBJ_3D_COL column in T_MATRIX_INFO with 3D information.

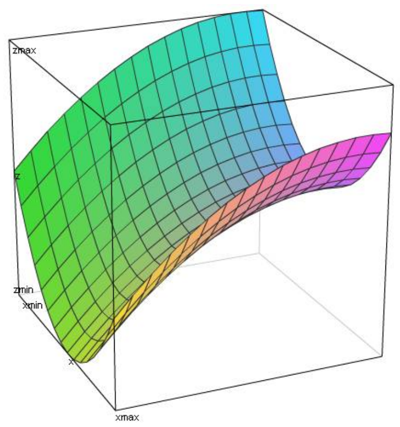


Figure 3.2: Coordinates for 3D CAD representation

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

```
UPDATE T_MATRIX_INFO
SET OBJ_3D_COL =
OBJ_3D_DATA(
    3,  --Number of Dimension
OBJ_3D_PLANE(X_AXIES, Y_AXIES, Z_AXIES),
OBJ_3D_DATA_TYPE(
( 0.0,0.0,0.0), ( 2.0,5.0, -15.0), (-1.5, -2.2, -2.5),(-3.8, -2.0, -12.3),( 2.4, -0.4, -3.5),(-
1.7,3.0, -7.5),( 1.3, -2.0, -2.5),( 1.5,2.0, -2.5), ( 1.5,0.2, -1.5), (-1.3,1.0, -1.5),
…………………
    )
  )
  WHERE ID=1;
```

After adding 3D data, checking from T_MATRIX_INFO for ID = 1

```
SELECT ID, ROW, COLUMN, VALUE, OBJ_3D_COL from T_MATRIX_INFO WHERE ID=1;

ID    ROW       COLUMNVALUE        OBJ_3D_COL
-----------------------------------------------
1     00        1
OBJ_3D_DATA(3,OBJ_3D_PLANE(0.0,0.0,0.0),OBJ_3D_DATA_TYPE(0.0,0.0,0.0,
2.0,5.0, -15.0, -1.5, -2.2, -2.5,-3.8, -2.0, -12.3,2.4, -0.4, -3.5,-1.7,3.0, -7.5,1.3, -2.0, -
2.5,1.5,2.0, -2.5, 1.5,0.2, -1.5, -1.3,1.0, -1.5
………………… ))
```

IV. PERFORMANCE ANALYSIS

In this part, we investigated the performance for traditional way and also newly developed 3D data type. Here, we used 'EXPLAIN PLAN' for both of cases and found huge performance improvement.

A. *TRADITIONAL DATA TYPE*

In order to the performance testing, we created 2 (two) additional tables (T_MATRIX_INFO_TRADITIONAL and OBJ_3D_DATA) and did the "EXPLAIN_PLAN" and found the following statistics:

```
EXPLAIN PLAN FOR SELECT T_MATRIX_INFO_TRADITIONAL.*
FROM T_MATRIX_INFO_TRADITIONAL INNER JOIN OBJ_3D_DATA ON OBJ_3D_DATA.ID=T_MATRIX_INFO_TRADITIONAL.ID
WHERE T_MATRIX_INFO_TRADITIONAL.ID=1;

SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());
-----------------------------------------------------
Plan hash value: 8863888095


-----------------------------------------------------
| Id  | Operation          | Name                     | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------
```

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

```
|   0 | SELECT STATEMENT    |                            |    36 |   3350 |     9  (15)| 00:00:01 |
|*  1 |  HASH JOIN          |                            |    36 |   3350 |     9  (15)| 00:00:01 |
|*  2 |   TABLE ACCESS FULL| T_MATRIX_INFO_TRADITIONAL  |    1 |     77 |     4   (0)| 00:00:01 |
|*  3 |   TABLE ACCESS FULL| OBJ_3D_DATA                |    36 |    395 |     4   (0)| 00:00:01 |
-----------------------------------------------------
Predicate Information (identified by operation id):
---------------------------------------------------


   1 - access("OBJ_3D_DATA"."ID"="T_MATRIX_INFO_TRADITIONAL"."ID")
   2 - filter("T_MATRIX_INFO_TRADITIONAL"."ID"=1)
   3 - filter("OBJ_3D_DATA"."ID"=1)

Note
-----
   - dynamic sampling used for this statement (level=2)
```

B. ***DEVELOPED 3D DATA TYPE***

In section III, we developed a 3D data type. Now doing the “EXPLAIN_PLAN” and found the bellow statistics:

```
EXPLAIN PLAN FOR SELECT * FROM T_MATRIX_INFO WHERE ID=1;
SELECT PLAN_TABLE_OUTPUT FROM TABLE(DBMS_XPLAN.DISPLAY());
-----------------------------------------------------

Plan hash value: 8045439966

-----------------------------------------------------
| Id  | Operation          | Name          | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------
|   0 | SELECT STATEMENT   |               |     1 |    77 |     4   (0)| 00:00:01 |
|*  1 |  TABLE ACCESS FULL| T_MATRIX_INFO |     1 |    77 |     3   (0)| 00:00:01 |
-----------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   1 - filter("ID"=1)
```

V. SIMULATION RESULTS

As per illustration of above section IV, it is clear the improved performance of our developed 3D data type for CAD. We just added our newly developed a 3D column instead of creating a different table what the traditional way followedand got the dramatically performance improvement in terms of Rows, Bytes, Cost (%CPU), Times (Seconds).

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

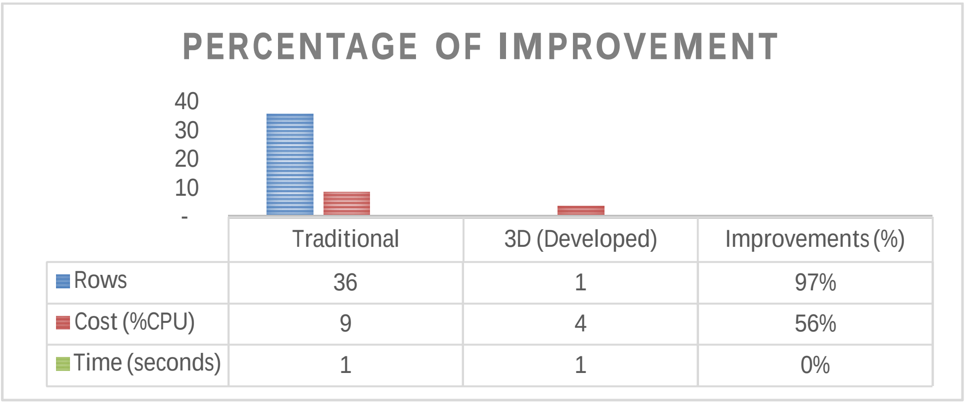
International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 4, Issue 9, September 2016

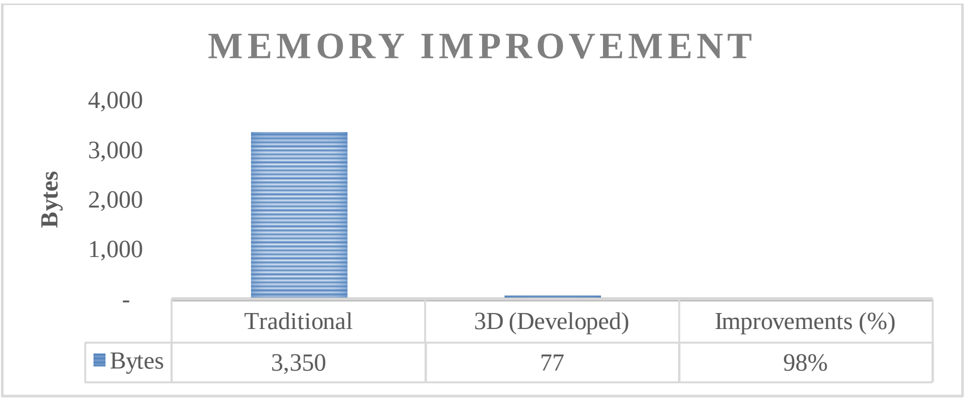
Table 5.1: Performance analysis of Traditional and newly developed data type

	Data type		
Category	**Traditional**	**3D (Developed)**	**Improvements (%)**
Rows	36	1	97%
Bytes	3,350	77	98%
Cost (%CPU)	9	4	56%
Time (seconds)	1	1	0%



	Traditional	3D (Developed)	Improvements (%)
🟦 Rows	36	1	97%
🟥 Cost (%CPU)	9	4	56%
🟩 Time (seconds)	1	1	0%

Figure 5.1: Performance analysis (Rows, Cost) for traditional and developed 3D data type



	Traditional	3D (Developed)	Improvements (%)
🟦 Bytes	3,350	77	98%

Figure 5.2: Memory analysis (Bytes) for traditional and developed 3D data type

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a clear research evidence of development 3D data type for CAD in DBMS. Where, this approach is implemented and contributed to the rising 3D data type for all CAD systems. We can get all information in a single table rather doing multiple pieces and joining from multiple tables; just adding a custom made column able to query. We presented how to develop a custom made data type and also showed how 3D data can be inserted without using existing spatial data type. Hence, it speeds up the process, reduce memory and also easy to manage data.

To conclude, our concept is developed, well tested and also provided promising outcomes in Oracle 11g database environment. Future research is to develop 4D data type for Computer Aided Design (CAD) systems.

VII. ACKNOWLEDGEMENT

This is a unique and interesting research work where it mainly focused to developed 3D data type for CAD in order to preserve 3D architecture of an object. I would like to thanks to research supervisor for giving the opportunity.

REFERENCES

1. Foley, J., van Dam, A., Feiner, S., and Hughes, J. 1995. Computer graphics: principles and practice. Addison Wesley, 2nd Ed. Geodata Infrastructure North-Rhine Westphalia (GDI NRW).
2. Arens, C. A. 2003. Modeling 3D spatial objects in a geo-DBMS using a 3D primitives. Msc thesis, TU Delft, The Netherlands. 76 p.
3. Arens, C., Stoter, J.E., and van Oosterom, P.J.M. 2005. Modeling 3D spatial objects in a geo-DBMS using a 3D primitive. In Computers & Geosciences, volume 31, 2. pp. 165-177.
4. CalinArens et al., Modeling 3D spatial objects in a geo-DBMS using a 3D primitive, Computers & Geosciences, Vol.31, pp. 165-177, 2005.
5. T.K Chen et al., 3D Spatial Operations for geo-DBMS: Geometry vs. Topology, International Archives of the Photogrammetry and Spatial Information Sciences, Vol.37, pp.549-554, 2008.
6. JantienStoter et al., Visualization and Editing of 3D Objects Organized in a DBMS, Geo-Database Management Center, 2003.
7. JantienStoter et al., 3D Modeling with National Coverage: Bridging the Gap, 3D Products at NMCAs, Open Issues.
8. Prof. Abbas Rajabifard et al., Advance Principles of 3D Cadastral Data Modeling, 2nd International Workshop on 3D Cadastres, 2011.
9. CalinArens et al., Modeling 3D objects in a geo-DBMS using a 3D primitive, 6th AGILE, France, 2003.
10. Narayan, K. Lalit (2008). Computer Aided Design and Manufacturing. New Delhi: Prentice Hall of India. p. 4. ISBN 812033342X.
11. Madsen, David A. (2012). Engineering Drawing & Design. Clifton Park, NY: Delmar. p. 10. ISBN 1111309574.
12. http://en.wikipedia.org/wiki/Spatial
13. http://en.wikipedia.org/wiki/Three-dimensional_space
14. http://en.wikipedia.org/wiki/Data_type
15. https://en.wikipedia.org/wiki/Cartesian_coordinate_system

BIOGRAPHY

Kazi Shamsul Arefin is engaged in research activities throughout his undergraduate years and has published more than 24 international research papers. Moreover, he is a member of International Association of Computer Science and Information Technology (IACSIT), Membership No: 80337708.