

# A Literature Review on Evolving Database

Shagufta Praveen

Research Scholar

Computer Science Department  
Glocal University, Saharanpur  
India

Umesh Chandra, PhD

Assistant Professor

Computer Science Department  
Glocal University, Saharanpur  
India

Arif Ali Wani

Assistant Professor

Computer Science Department  
Glocal University, Saharanpur  
India

## ABSTRACT

Since 90's, database has shown tremendous growth. This growth can be determined in different aspects. Different demands of every era give database a new bunch of challenges. To achieve that challenges, researchers come up with different ideas and combinations. These various combinations enhance features of database and this way database starts evolving from one period to another. Database that we had in 1960 is entirely, absolutely different from what we have now. This survey addresses the evolution of database from beginning to today. It also introduces that how different technologies, concepts and theories helped database to evolve, as just one good player in a team is not enough to win the match. The fact is still industries are in hope for something better in context to database technologies.

## Keywords

Evolution, RDBMS, ORDBMS, OODBMS, NOSQL, NewSQL

## 1. INTRODUCTION

A meaningful data becomes an information is the foremost line that people often start their studies with. A data should be stored properly so that it could become information for the reader and for whoever accessing it. Finally a system came into existence which gathers all important fact and figure about a particular product. And then things are enhanced by people as per their need. Manual System (File system) moved to Data processing system (Sequential File system, Random access file system). Later data processing moved to information processing, and finally came up with demand of MIS (Management Information system.)

### 1.1 Relational Database

Early in 1960's, IDS (Integrated data store) was designed by Charles Bachman, which was a network database, it influenced other DBMS for further growth. It was designed to improve performance through available hardware of that time.

In 1966, IMS (Information management System) released by IBM, and was based on hierarchical database. It was designed to keep the stock details of very large bills of materials for Saturn V moon rocket and Apollo space vehicle.

Survey presented by Shamkant B. Navathe (Navathe,1992)and Michael Stonebraker et. al (Stonebraker et al. 2000) both tried to describe the difference between network database and hierarchical database. In Hierarchical

data model records are organized in the form of tree structures and records were related to each other as parent-child relationship. As per hierarchical model, tree structure can have only one parent. Hence, one to many relationships were possible but not many to many (Gill, 2010).Whereas, IMS (based on hierarchical data model) was not strictly followed hierarchical. In this model, a child could have two parents:

Physical parent and logical parent. Physical parent is the parent present in the record whereas logical parent to connect a record to another different hierarchy. Network data model shows relation between objects through graph structures. It was useful to decrease redundancy too.

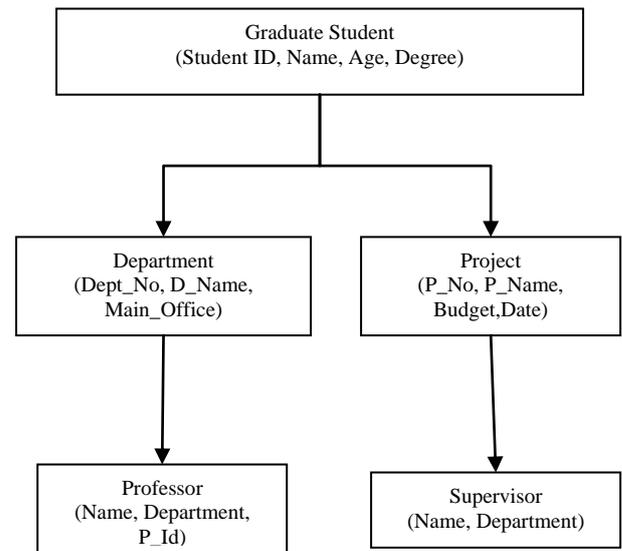


Fig 1: Hierarchical Database

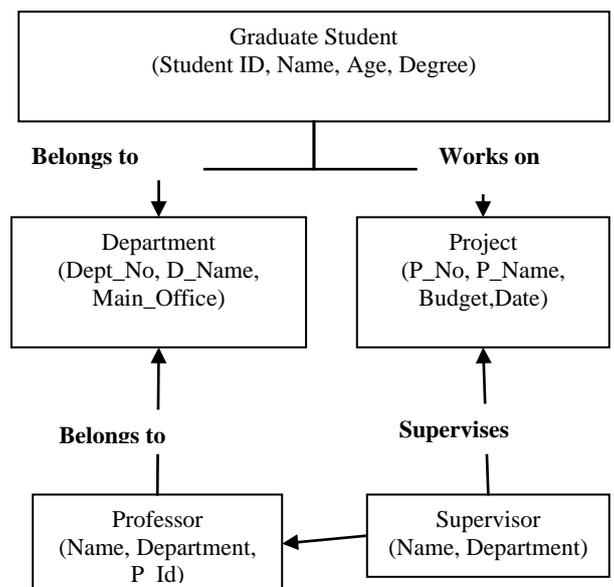


Fig 2: Network Database

In 1970, Relational data model was proposed by E. F. Codd (IBM Researcher). It was theoretical paper which was presented for the model. E. F. Codd presented series of paper after this. As per him in relational database, through table form data is represented. Accessing of the data doesn't require any algorithm whereas a non procedural high level language is used.

Content: Different sections of his paper were (Codd, 1970)

- a) Data Dependencies in present system
  - Ordering Dependence
  - Indexing Dependence
  - Access path Dependence
- b) A Relational View Of Data
- c) Normal Forms
- d) Some Language aspects
- e) Expressible, Named and stored Relations
- f) Operations On Relations
  - Permutation
  - Projection
  - Join
  - Restrictions
- g) Redundancy
  - Storage Redundancy

- Weak Redundancy

h) Consistency

In 1980, Standard approach developed for DBMS and that was Relational Model. SQL was developed as query language and adopted by ANSI and ISO. In the same year many relational DBMS came into existence like Oracle, DB2 Informix.

**1.1.1. Limitation:** Implementation for this proposed work was not discussed. For section name language aspects, only some of the properties of language were demonstrated (Codd, 1970). Many researchers were not ready with the concept of the relational model in starting.

After using RDBMS researchers found relational model doesn't support multimedia data, unstructured data and inheritance relationship (Silberschatz et al., 2006)(Chorafas et al., 1993). Relational model doesn't support scalability for applications. When several tables are joined together leads to inefficient query processing (Dietrich et al., 2005)(Rahayu et al., 2001). The cost for joining tables is very high as join operation in RDBMS is very expensive (Bertino et al., 1992).

- High level query language is used such as SQL
- While designing a table, table should have a primary key and foreign key should also be added to show many to many relationships between table
- Referential constraints should be defined
- Correct Data type and value restriction should be used for every field of table.

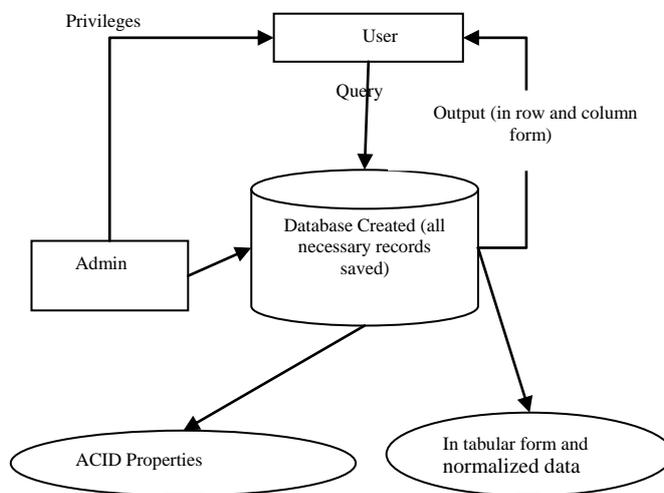


Fig 3: Relational Database

## 1.2 Object Relational

In 1990, Object- relational database came into existence. The idea behind the object-relational database is adding object concept with relational database. It's like adding OOP principle (Inheritance, encapsulation and polyphormism) with relational database. This idea was given by Stonebaker et al. who proposed to extend the capabilities of RDBMS by support of rich object structures. Components required for an object relational model are: Data structures (object table for storage), Integrity constraints(object identifier, relationship), Operations (query language)(Codd, 1970).

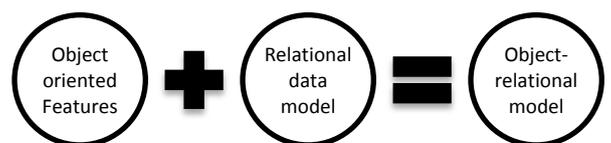


Fig 4: Object-Relational Database

### 1.2.1 Reason

The main goal of Object-Relational is to bridge the gap between relational database and object-oriented modeling technique used in programming language such as Java, C++. In order to achieve storage for complex data types and to show relationships while developing applications. By adding both of these technologies a user can achieve transaction management, scalability, flexibility, performance management, user-defined data types and much more.

### 1.2.2 Limitation

Increased complexity and associated cost. There is large semantic gap between two different technologies. Object applications are not data centric ones like relational based ones.

## 1.3 Object- Oriented

After 1990, OODBMS came into existence. Concept behind OODBMS is the combination of object oriented programming principle and database management concept. Its basic feature is to access object in the database. It supports features like encapsulation, complex object, types and classes, extensibility, persistence etc. Instead of records like E-R model, data is represented in form of classes and objects in OODBMS. Objects are run time entities which have importance in real world problem. All values which are stored in an object are in the form of instance variable. A collection of same type of objects having same type of values and method is called classes. Class is a user-defined data type. Two objects can interact with each other by invoking the method of other objects. Method and data are not visible outside the object and therefore Object oriented Data model provide data abstraction (Kshitij et al., 2001). It is considered to be fifth generation database.

### 1.3.1 Reason

It was designed to store and manage object created by programming languages like Java, C++ and provide object-oriented features to users. It was designed to satisfy the demand for better model for real world entities and to provide richer data model than traditional data model. OODBMS used multi-valued properties to represent complex data structures whereas, in relational, it is done by additional relations. OODBMS allow nested object structures; it can have non-primitive objects as values (Bertino et al. 1992). OODBMS can reduce the issue of issue through the feature of navigating through object structures and the resulting path expressions in object attributes (Bhagui, 2003).

### 1.3.2 Limitations

There was lack of flexibility like schema change was one of the issues as in OODBMS, any change in the schema makes user to change in other classes too. It was language dependent, as it was tied to a specific language by specific API. There was lack of Ad-Hoc queries too. Queries that can be performed on the data in an OODBMS are highly dependent on the design of the system (Bertino et al. 1992). Lack of security, increase in complexity, lack of standards, and lack of experience were some other issues. In spite of different proposals, OODB doesn't support Views. It was complicated to develop object-oriented view by such model having feature as object identity (Bangui. 2003)

Dr. Michael Stonebraker classified DBMS with a help of a four quadrant (Collony., T.M. et al. 2002)

Query	Relational Database	Object- relational Database
	File System	Object-Oriented Database
No Query	Simple data	Complex data

Fig 5. DBMS classification by Dr. M. Stonebraker

## 1.4 Multi-Dimensional Database

Multidimensional structures are used in this data model to organize data and represent relation between stored data. These structures are represented through broken cubes and these cubes are used to store data and user access data from these cubes. These databases use online analytical processing application. Star Schema and snowflake schema are used to represent relational implementation of multidimensional data model. Idea behind this data model was as relational model represent itself in rows and column in table form, like same multi-dimensional data base can be created as cube with many dimensions. Two aspects are considered while creating cubes i.e. Dimension and facts. Dimension helps to divide data into comparable group and facts are numerical values about data like quantity, size or magnitude (Langseth,2004) (Srimani et al.).

### 1.4.1. Reason

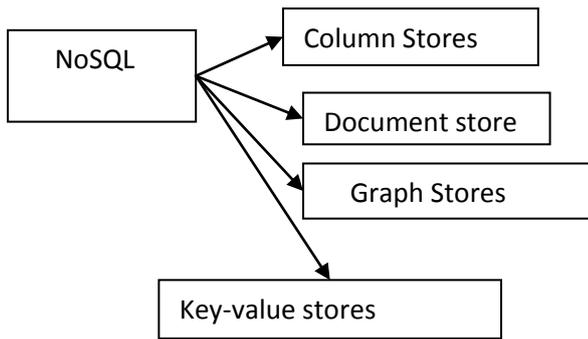
It was designed for data analysis and for data ware housing. It was designed to achieve better data extraction and for a scalable data model.

### 1.4.2. Weakness

Data updation and searching can be challenge. It failed to deal with database having sparse data.

## 1.5.NoSQL Database

After last 40 years, variation in relational model has been seen. Many researches has been done with the thought to extend capability of relational database by adding and making variations in it. But this data model was not in tabular form. Name of this model is NoSQL termed as Not only SQL. Scientists were working for this model from 19's but it came into power in 21<sup>st</sup> century. This model gave us new more idea and flexibility for data storage. It mainly include four data model : Key-value stores, Document , Graph and column oriented. With this model scenario came up with 150 NoSQL products and most of them are adopted by leading organization today.



**Fig 6: NoSQL Database**

**Key-value Stores:** Key-value store is based on key-value pair, which is similar to a dictionary (Hecht et al,2011). It is one of famous data model which similar to associative array (key-value pair based) .It was found to be very efficient because of its time complexity of its access data which is  $O(1)$ ( Grolinger et. al. 2013).Value are stored in this model correspond to a unique key and these key used to access and retrieve value from or into the data store. It is a schema free model, provide rapid processing. Through keys only queries are meant to be performed in this model as values are stored with a respective key. Some of the key-value store based products are: Riak, Level DB, Berkeley DB, etc.

**Column Oriented Stores:** Column oriented is based on table and row but many different features separate it from our traditional database. Like in column oriented database a number of columns can be added corresponding to particular row. That row has unique identifier which acts as a primary key in the table. A number of rows with unique identifier can be added to a table. Every row can have millions of columns. These columns are called column family. It is also schema free but yes its basic infrastructure resembles to relational database. Some of the column-oriented based NoSQL products are: HBase, Cassandra. These products belong to Column store family. A large amount of data can be processed through this store.

**Document oriented Stores:** As per name, many people think this store is to store documents or a document management system. But this word document- oriented refers to loose collection of Key/value pairs. But the difference is instead of dividing document to store in key-value store. Data is stored as whole document. Document oriented stores document using JSON (He, C.2015). A document can have different set of structures, complex structures (nested objects) and it doesn't require any fixed schema (Grolinger et. al. 2013), example: MongoDB

**Graph Database:** Graph theory is the concept from where graph database came into existence. As graph theory is mathematical concept which represent object relations through nodes, edges and vertices. Like same graph database store relationship between different nodes and it is specialized in handling highly interconnected nodes and efficient in traversing relationship among different objects (Grolinger et. al. 2013). With storing nodes, edges, weight of edges, it also provide graphic algorithm and support graphic query (He, C.2015).Example: Neo4j.

**1.5.1. Reason**

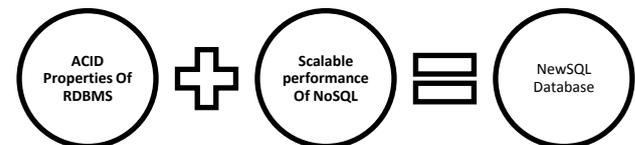
It was designed to cope up with scalability issue (Big Data Challenge) and to increase availability .No dependence on any particular schema, lack of complexity (No Join Operations). It was designed to achieve consistency in distributed system

**1.5.2. Limitations**

The CAP- Theorem, which says during partition tolerance any of the property between consistency or availability will be compromised. It has Lack of experience, support no Firm ACID transaction like relational database.

**1.6.NewSQL Database**

NewSQL is also a new term in database that works on relational model and provide scalability performance same as NOSQL.



**Fig 7: NewSQL Database**

**1.6.1. Reason**

The main aim of NewSQL is to provide scalability, availability without compromising with transaction capabilities and strong consistency.

**1.6.2. Limitation**

Many Existing RDBMS products support many of its features.

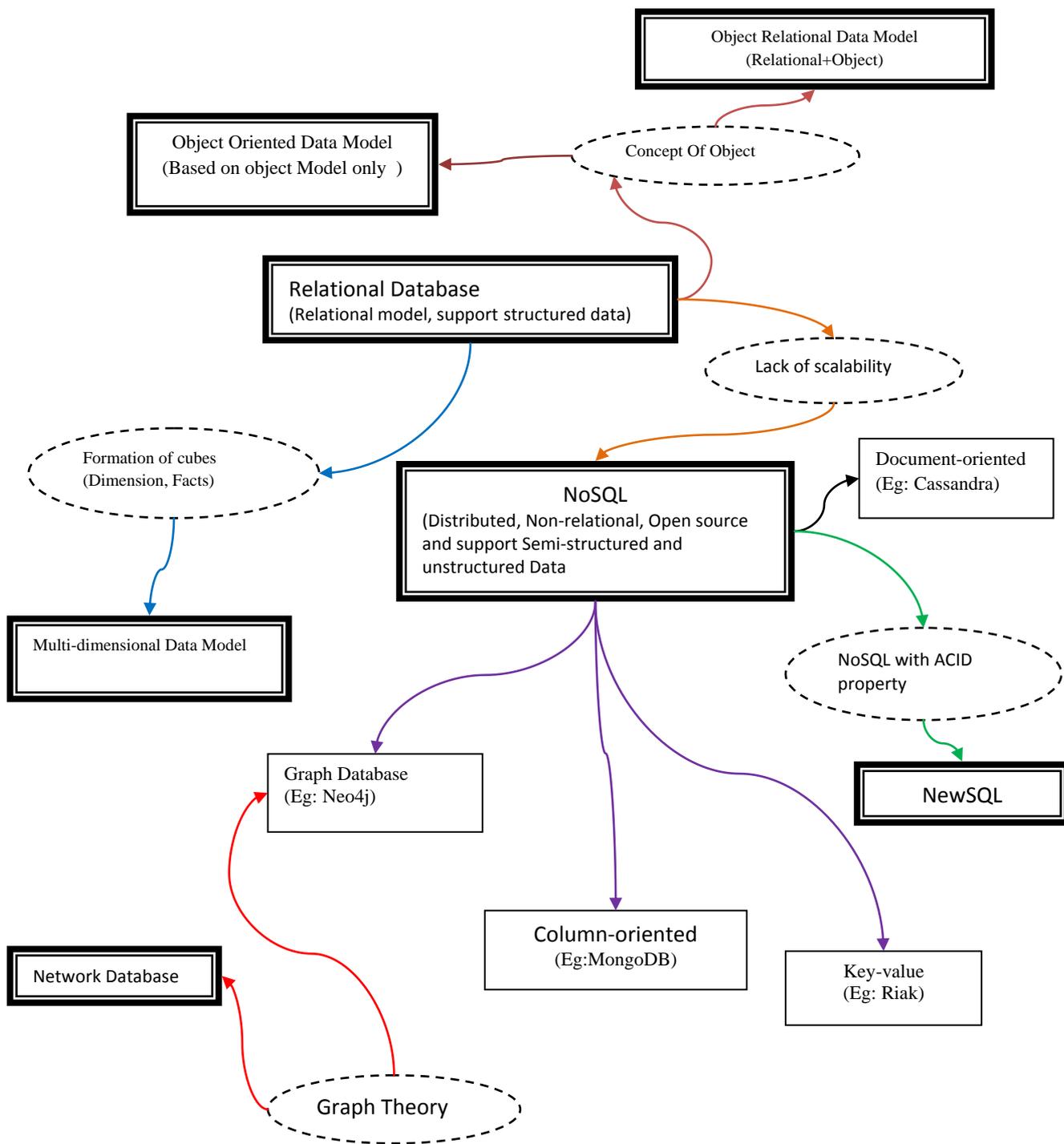


Fig 8: Graph showing relation between various databases

Table.1. Databases

Databases	Year	Objective	Products	Limitation
NewSQL	2011	-to achieve scalability with ACID properties	-Neo4j -VoltDB	-Not much experienced -
NoSQL	2009	- To achieve scalable, flexible database by compromising with	-HBase	-CAP theorem can affect availability or

		ACID properties  -Support unstructured data and called non-relational database	-Cassandra -MongoDB -CouchDB	consistency while partitioning  -new in market
Multi-dimensional database	1993	-Relating different tables in form of a cube  -For Dataware mining and data warehousing  -For analytical processing	-Aplix  -Infor  -Cognos	-Complex queries  -difficult in Data update, searching
Object Oriented Data Model	1980	-mapping between different objects to perform a task	- Gemstone  -GBase  -VBase	-Lack of standard  -Lack of view control  -concurrency by lock
Object-Relational Model	1985	-Support both relational data model concept and object oriented principle	- IBM DB2  -Postgre SQL  -Illustra	-Complexity  -Semantic gaps between both of the technologies
Relational Data Model	1970	-To store huge volume of data  - Store data into multiple table in place of a single table  -to define relations between tables  -Queries to access data from table	- Oracle  -MySQL  -Informix	- don't support storage of semi-structured data  -Scalability issues
Network Data Model	1969	-Flexible model for collection of data  -Data Connected through graph theory where object is node and connection is arc  -N:N relation	-IDS  -IDMS  -Univac DMS-1100  -RDM Sever	-Making changes in database is difficult
Hierarchical Data Model	1960	-collection of data in form of records  -records are linked by pointers  -1:N relations (Followed tree structure)	-IMS	-Record replication  -Inconsistency  -Space wastage

#### 4. CONCLUSION

This survey represents the variation in data model since 90's to current century. It also represent that researchers always thought to add something to relational database and gave new model for betterment. Due to this relational model was working with us from last many years but NoSQL gave us new direction and provide new way to organize, retrieve data. This all shows that relational model is old enough to trust and to work with, as many number of experiments has already been done with it. Like same NoSQL is under development

and adding new features to extend it capability will make users more understand that what next generation requirement is, after this. Polygot persistence is another new technology in the way of database generation after NewSQL and NoSQL.

## 5. REFERENCES

- [1] Gill, P.S.2010 Database Managementt System. I.K. International Pvt. Ltd.
- [2] CODD, E.F.1970. A Relational Model of Data for Large Shared Data Banks.
- [3] Navathe B.S.1992. Evolution of data modeling for data base
- [4] Stonebaker,M.,Held,G.2000.Network, hierarchies and relational in database management system
- [5] Silberschatz, A., Korth, H.F., Sudarhan, S.2006 Database System Concepts. McGraw-Hill, Inc, New York, NY, USA
- [6] Chorafas, D. N., and Steinmann. H., 1993.Object-Oriented Databases. Prentice Hall, Englewood Cliffs.
- [7] Dietrich,S. W., and Urban. S.D. 2005.An Advanced Course in Database Systems Deyond Relational Databases. Pearson Prentice Hall, Upper Saddle River.
- [8] Rahayu, J.W., Chang, E., Dillon, T.S., Taniar. D., Performance evaluation of the object-relational transformation methodology. *Data Knowledge Engineering*, 38(3):265–300, September 2001.
- [9] Jain, k. and Bhatnagar, N. 2001.Database Management system
- [10] Bertino, E., Negri, M., Pelagatti, G., and Sbattella, L., “Object-Oriented Query Languages: The Notion and the Issues,” *IEEE Transactions on Knowledge and Data Engineering*, Vol. 4, No. 3, 1992.
- [11] Bhagui.S., 2003.Achievements and Weaknesses of Object-Oriented Databases , Vol. 2, No. 4, July-August
- [12] Langseth, J., 2004, “Real-Time Data Warehousing: Challenges and Solutions”, DSSResources.COM,
- [13] Srimani, P.K., Rajasekharaiah K.M .THE ADVANTAGES OF MULTI DIMENSIONAL DATA MODEL – A CASE STUDY
- [14] Hecht R, Jablonski S: NoSQL evaluation: A use case oriented survey.*Proc 2011 Int Conf Cloud Serv Computing* 2011, 336–341
- [15] Grolinger et. al .2013.Data management in cloud environments: NoSQL and NewSQL data stores
- [16] He, C.2015.Survey On NoSQL technology.
- [17] Connolly., T.M. et al.2002.Database System: A practical Approach to design and implementation, 3<sup>rd</sup> Edition
- [18] [https://en.wikipedia.org/wiki/Integrated\\_Data\\_Store](https://en.wikipedia.org/wiki/Integrated_Data_Store)
- [19] [https://en.wikipedia.org/wiki/IBM\\_Information\\_Management\\_System](https://en.wikipedia.org/wiki/IBM_Information_Management_System)
- [20] [http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos\\_act/2000/DAP\\_DisAvDB/documentacion/OO/Evol\\_Da taModels.html](http://www.fing.edu.uy/inco/grupos/csi/esp/Cursos/cursos_act/2000/DAP_DisAvDB/documentacion/OO/Evol_Da taModels.html)
- [21] [https://en.wikipedia.org/wiki/Object-relational\\_database](https://en.wikipedia.org/wiki/Object-relational_database)
- [22] <http://ecomputernotes.com/database-system/adv-database/object-relational-database-systems>
- [23] <http://www.25hoursaday.com/whyarentyouusinganoodb ms.html>
- [24] <http://www.dbta.com/Columns/DBA-Corner/What-Is-a-NewSQL-Database-System-104489.aspx>