Global Journals LaTeX JournalKaleidoscope™
Artificial Intelligence formulated this projection for compatibility purposes from the original article published at Global Journals. However, this technology is currently in beta. *Therefore, kindly ignore odd layouts, missed formulae, text, tables, or figures.*

Object Oriented Database Management Systems-Concepts, Advantages, Limitations and Comparative Study with Relational Database Management Systems By

Hardeep Singh Damesha[1]

[1] Lovely Professional University

Received: 16 December 2014 Accepted: 5 January 2015 Published: 15 January 2015

Abstract

Object Oriented Databases stores data in the form of objects. An Object is something uniquely identifiable which models a real world entity and has got state and behaviour. In Object Oriented based Databases capabilities of Object based paradigm for Programming and databases are combined due remove the limitations of Relational databases and on the demand of some advanced applications. In this paper, need of Object database, approaches for Object database implementation, requirements for database to an Object database, Perspectives of Object database, architecture approaches for Object databases, the achievements and weakness of Object Databases and comparison with relational database are discussed.

Index terms— relational databases, object based databases, object and object data model.

I. Introduction istory of data processing goes through many different changes with different technologies along with the time. In decade there is huge increase in the volume of data that need to be processed due to which sometimes old technology do not work and need to come with new technology to process the data. History of database technology has used Unit Records & Punch Card, Punch Card Proliferation, Paper Data Reels, & Data Drums, File Systems, Database Systems, NoSQL and NewSQL databases. From last five decades, the mostly used technology is database management systems.

After some limitations of file systems, researchers come up with new technology known as Database Management Systems which is the collection of software or programs to maintain the data records. Initially, two models are proposed are hierarchical and network models, but these models don't get much popularity due to their complex nature. Then a researcher E.F. Codd comes up with a new data model known as relational model in which data items are stored in a table. Many DBMS's are developed on the basis of this model. This is the most popular model till now because it has conceptually foundation from relational mathematics.

In mid-1980's ,no doubt RDBMS are very much popular but due to some limitation of relation model and RDBMS do not support for some advanced applications [1] OODB comes in the picture. At that time Object Oriented Programming paradigm is very much popular. Due to this researcher think to combine the capabilities of database and object based paradigm for programming. In Object databases data is stored in the forms of objects. These database management systems are not very much popular because due to the lack of standards.

Research is going on the database technology from 1960's up to this day. Many improvements are done in database technology by researcher in last decade and more technologies are coming to improve the database technology. The new database technologies include new transaction management and concurrency control methods and Redundant Array of Independent Disks (RAID) for efficient storage and Big Data and Cloud Computing.

1 II. Why Object Oriented Databases?

There are three reasons for need of OODBMS: For this you must know:

? Depth of recursive query must be known.

? You can use the transitive closure operations to handle recursive queries in RDBMS. 9. Impedance mismatch: SQL Data Manipulation Language (DML) is lack computational completeness [10]. To overcome this situation, you must embed the SQL with any high programming language like C++, Java, and C #. Due to there will be impedance mismatch between two language SQL and higher programming language. 10. Poor support for long duration transactions: In RDBMS, generally transactions are short lived and concurrency control techniques or mechanisms are not good for .long duration transactions. 11. Poor Schema Evolution support: Schema Evolution means making changes to schema of database at runtime without interrupt the execution of the application. 12. Poor Navigational Access: There is very poor support for the navigational access in RDBMS.

There are some advanced applications need the database with deeper structural and functional foundation of capabilities that are not provided by conventional database [1]. These applications are:

2 B. Need for Advanced Applications a) Computer Aided Design (CAD):

? In these types of applications, relevant data about buildings, airplanes and integrated circuit chips is stored and managed. In this type of applications, database design may be very large.

? Design in these types of applications is not static. This design is evolves through the times. Updates need to be propagated.

3 C. Popularity of Object Oriented Paradigm

Another domain that enforces the development of OODBMS is popularity of object oriented programming paradigm [4], [5], [6], [7], [8]

4 IV. Oodbms Manifesto a) Mandatory Features

The OODBMS paradigm manifesto set the minimum fundamental directional basis for an OODBMS model [3], [4], **??**5, [8].These characteristics can be classified as mandatory and optional features:

1. Support for complex objects: A OODBMS must support for complex objects. Complex objects can be obtained by applying constructor on basic objects. 2. Object Identity: It is the unique identifier associated with every object in the system. It has following characteristics:

? It is generated by system.

? It is unique to that object in the entire system.

? It is used only by the system, not by the user.

? It is independent the state of the object. 3. Encapsulation: An OODBMS should enforce encapsulation through access objects only.

5 Types or Classes:

A OODBMS must support for one of them types or classes. 5. Inheritance and Hierarchies: A OODBMS must support for concept of super classes and subclasses. The types of heritance can be: ? Substitution ? Inclusion ? Constraint ? Specialization 6.

6 VI. OODBMS Architectures Approaches

The basic theme of OODBMS is to add persistence to OOPL as they provide object orientation. The major difference is that here database needs to store data as well as methods [1]. c) Database Server: In this approach, Client simply passes the request to the server, receives results and passes them to application. Most of database processing done at server .This approach is used mainly by RDBMS's.

7 Client/Server

VII. Achievements and Weaknesses of OODBMS a) Achievements 1. Support for User Defined data types: OODBs provides the facilities of defining new user defined data types and to maintain them [9].

8 OODB's allow creating new type of relationships:

OODBs allow creating a new type of relationship between objects is called inverse relationship (a binary relationship) [11]. [11]. 9. No full-fledged facilities to implement complex objects: No doubt, object oriented databases provide some facilities to implement the concept of complex objects. But there is no full -fledged implementation of complex objects [11]. 10. Interoperability between OODB and Object Oriented Systems: In Object Oriented Programming objects are transient in nature. To provide persistent to data, OODB and OO systems need to be interoperable. Many problems may arise during interoperable between OODB and OO systems [11]. 11. Limited performance gains over RDBs Decrease in performance: Performance gains changes application to application. Applications that make the use of object identity concept having performance gains over RDBMS's. But application that requires bulk database loading and does not make use of OID then

performance of OODBMS's is not good [9]. 12. Some basic features are not present: Some basic features like triggers, meta data management [11] and constraints such as UNIQUE and NULL [9] not present in object databases.

VIII. Comparison of OODBMS and RDBMS

9 IX. Summary and Conclusion

No doubt, relational databases are very popular and there are found everywhere. The object oriented database comes into action in mid-1985's to remove the limitations and to support some advanced database applications like CAD, CASE etc. Another point that provides the momentum to develop object based database is popularity of object based programming. So; researchers in the field of database think to combine the object oriented programming concepts with database to make powerful database management systems. Different approaches are adopted by industry to make the database with object oriented features. These approaches include relational extensions and pure object oriented are most popular to develop object oriented database systems. OODBMS's are made by many vendors by using different approaches. OODBMS's removes the limitations of RDBMS's, also provide support for advanced database application with some additional features. But due to the lack of standards, they do not get much popularity in the industry. Then after some time, some limitations are found in object oriented database management systems. [1] [2]

Figure 1:

[1] © 2015 Global Journals Inc. (US) 1
[2] © 2015 Global Journals Inc. (US)

to this, sometimes it becomes very difficult to find
out that which is going to model data or
relationship?
4. Poor support for integrity and enterprise constraints:
Constraints are very much needed for your
database have to be desired data. RDM supports
only limited number of constraints. The enterprise
constraints are those which are defined by industry
standards.
5. Homogeneous data structure: RDM requires
homogeneous data structures like:
? RDM assumes both horizontal and vertical
homogeneity.
? Relational mathematics algebra has only fixed
number of operations due to which Relational
Model operations cannot be extended.
6.
A. Limitation of RDBMS
B. Need for Advanced Applications
C. Popularity of Object Oriented Paradigm
A. Limitation of RDBMS
These limitations are in relational model. Due to
this these limitations are reflected to all RDBMS [2].
These limitations are:
1. Poor representation of real world entities: The
Relational model cannot represent real world in
proper way because it has only one semantic that is
table which can represent the real world entity in
proper way.
2. Normalization is necessary, but sometimes not

[Note: useful: Normalization in RDBMS to maintain the consistency of the database, but some broken relations is not related to real world.3. Overloading of semantic structure: Relational DataModel has only one semantic structure for representing data and relationship that is table. Due]

Figure 2:

? Mapping to OO schema and queries to relational ones.
? The underlying relational storage manager is ALLBASE/SQL.
C. Pure OODBMS
These type OODB's systems are not much popular because lack of standards [9]. There is no single definition for a single concept. For Example: An Object has many definitions, but in RDB there is a fixed standard

Information Hiding: It is process of separates external properties of an object from its internal properties, which are hidden from external environment. These two concepts also related with abstraction. Importance: These two concepts support the facility that internal properties of an object to be changed without affecting applications that use it, provided external properties remain same. It also provides data independence. ? To provide fully implemented functionality of complex objects. ? Support for User/Abstract Defined Types, operators and functions for accessing. ? To provide functionality of Active Databases and Inferencing.

? Database management, concurrency transaction control etc. functionality like ? Facility of Object Identity (OID). ? Facility of encapsulation. ? Facility of complex objects. ? Inheritance not must but may useful.

? QUEL is the manipulation language in INGRES.
? POSTQUEL in POSTGRES :Most of QUEL commands are included in POSTQUEL:
1. Time varying data (snapshots and historical data)
2. Iterative queries 3. Alerters, Triggers and Rules for Inferencing.

OODBMS: It is system which contains application programs which are used to manage all object oriented database activities like manipulation of objects.
Some Commercial OODBMS [9]: ? Gemstone OODBMS is develop Systems Incorporation.
? Objectivity/DB OODBMS is developed by ? Stick to relational r

B. Object/Relational DBMS
These systems have relational and object based both features by the definition [1], [2]. They provide similar objectives as provided by the Relational Extension approach of RDBMS. In this approach, build an object layer on the top of relational system like Open ODB and ODAPTER. They are built on different applications. It has following features: HP during mid's 90 and aims to support for broad base Open ODB/ODAPTER: Open ODB is an ORDBMS from architectures like Query Server or Client/Server.

Generalization: It is method to create a superclass is called generalization. Specialization: It is process of forming a sub class is ? Based on Iris

because a real life situation can be model in best way by using obje

V. OODBMS Perspectives
a) Single Level Storage Model
Dynamic Binding: An OODBMS must support
concept of dynamic binding in programming
language such as:
? Overloading
? Overriding
? Late binding
7. Computationally Complete DML: To provide a
support for data processing database have use
computationally completely language like SQL-3.
8. Extensible set of data types: A OODBMS must
support for used defined data types.
9. Data Persistence: This is basic requirement for any
DBMS.A OODBMS must provide persistent by
storing object in proper way.
10. Managing very large databases: A OODBMS must
support for large databases.
11. Concurrent Users: This is basic requirement for any
DBMS. It must support for concurrency control.
12. Transaction Management: This is also basic
requirement of any DBMS.
13. Query Language: This is also a basic requirement of
any DBMS. This query language must be
computationally complete.
b) Optional Features
1. Multiple Inheritance: Multiple inheritance is not
directly support by multiple objects oriented
programming languages. An OODBMS can also
support for multiple inheritance.
2. Type checking and inferencing: Type Checking and
Inferencing features can be added to Object
Databases.
3. Long duration and Nested Transactions: Relational
database transactions are short-lived. An OODBMS
can support for .long duration transactions and also
for nested transactions.
4. Distributed databases: An object database may
have support for distributed database which is a
collection of multiple databases logically related
and distributed over the network.
5. Versions: An OODBMS can support for version
control and configuration management.

Figure 4:

3

d) Weaknesses
1.

[Note: the system[13].Optimization of queries is very important for performance gains. But due to following reasons it is difficult to optimize queries in object databases:?User defined data types ? Changing variety of types ? Complex objects, methods and encapsulation ? Object Query language has nested structure ? Object Identity 3. No fixed query algebra for OODBMS's: Due to lack of the standards in OODBMS, there is no standard query algebra for OODB. Lack of standard query algebra becomes one of the reasons for problem of query optimization. There are different query languages for different object databases. 4. No full-fledged query system: Query system also not fully implemented. Some query facilities lacks in Object databases like nested sub-queries, set queries and aggregation function [9], [11]. 5. No facility for Views: In relational databases, views are temporary tables. Object databases having no facility for views. An object oriented view capability is difficult to implement due to the features of Object Model such as object identity. Object oriented programming concepts like inheritance and encapsulation makes the difficult to implement views in object databases [11]. 6. Security problems in Object databases: Security is related to authentication, authorization and accounting. Discretionary Access Control (DAC), Mandatory Access Control (MAC) security policies are implemented in object databases to secure data. In some systems provide security on the basis of object oriented concepts like encapsulation.Object database having to facilities for authorization[9].7.No support for schema evolution with OODBs:]

Figure 5: 3 .

.1 Acknowledgement

The Author would like to special thanks Sardar Chet Singh Damesha, Sardarni Sachiar Kaur Damesha and whole Damesha family for inspiring me everytime and always with me.

[Kim (ed.) (1988)] *A foundation for object-oriented databases*, W Kim . MCC Tech. Rep., N.ACA-ST-248-88 (ed.) Aug. 1988.

[Scholl and Schek ()] 'A relational object model'. M Scholl , H Schek . *Proc. 3rd Int. Conf. On Database Theory*, Lecture Notes in Computer Science S Abiteboul, P C Kanellakis (ed.) (3rd Int. Conf. On Database Theory) 1990. Springer Verlag. 470 p. .

[Bancilhon ()] 'Object Oriented database systems'. F Bancilhon . *Proc. 7th ACM SIGART/SIGMOD Conf*, (7th ACM SIGART/SIGMOD Conf) 1988.

[Object Oriented Database Systems: Approaches and Architectures by C.S.R. Prabhu 2. Database System: A Practical Approach *Object Oriented Database Systems: Approaches and Architectures by C.S.R. Prabhu 2. Database System: A Practical Approach to Design, Implementation and Management by T. Connoly and C. Begg*, (Research Papers)

[Kim and Kim ()] 'Object-Oriented Database Systems: Promises, Reality, and Future'. W Kim , W Kim . *Modern Database Systems: The Object Model, Interoperability and Beyond*, 1995. Addison Wesley. p. .

[Stefik and Bobrow (1986)] 'Object-oriented programming: Themes and variations'. M Stefik , D Bobrow . *The AI Mag* Jan 1986.

[Bertino et al. ()] 'Object-Oriented Query Languages: The Notion and the Issues'. E Bertino , M Negri , G Pelagatti , L Sbattella . *IEEE Transactions on Knowledge and Data Engineering* 1992. 4 (3) .

[Erlingsson] *Object-Oriented Query Optimization*, U Erlingsson . (unpublished manuscript)

[Zdonik and Maier ()] *Readings in Object-Oriented Database Systems*, S B Zdonik , D Maier . 1989. San Mateo, CA: Morgan Kauffman.

[Atkinson et al. (1989)] *The Object Oriented Database Manifesto*, M Atkinson , F Bancilhon , D Dewitt , K Dittrich , D Maier , S Zdonik . December 1989.

[Atkinson ()] 'The object-oriented database system manifesto'. M Atkinson . *Proc. Int. Conf. On Deductive and Object-Oriented Databases*, (Int. Conf. On Deductive and Object-Oriented Databases) 1989.

[Dittrich and Dittrich ()] 'Where Object-Oriented DBMSs Should Do Better: A Critique Based on Early Experiences'. K A Dittrich , K R Dittrich . *Modern Database Systems: The Object Model, Interoperability and Beyond*, (Kim, W.) 1995. Addison Wesley. p. .