International Journal of Engineering Research & Management Technology

May- 2015 Volume 2, Issue-3

Email: editor@ijermt.org www.ijermt.org

"Sharpening Skills..... Serving Nation"

Relational Database Management System Over Object Oriented Database Management System

Amod kumar upadhyay
M.Tech Scholar
Department of Computer Science & Engineering
Al-Falah School of Engineering and Technology
Faridabad Haryana

Shahid Sagar
Assistant Professor
Department of Computer Science & Engineering
Al-Falah School of Engineering and Technology
Faridabad Haryana

ABSTRACT:
Object-oriented database systems began developing in the mid-80's out of a necessity to meet the requirements of applications beyond the data processing applications which were [are] served by relational database systems. This paper serves as an overview on the achievements of object-oriented database technology so far, and also discusses the weaknesses that have to be yet resolved by the object-oriented database community before object-oriented database technology can become as widespread as relational databases.

KEYWORDS—database, object oriented database, relational database, advantages and disadvantages

I. DATABASE:

A database is a an organized collection of related data held in a computer or a data bank, which is designed to be accessible in various ways. The data within a database is structured so as to model a real world structures and hierarchies so as to enable conceptually convenient data storage, processing and retrieval mechanisms. Clients (Services or applications) interact with databases through queries (remote or otherwise) to Create, Retrieve, Update and Delete (CRUD) data within a database. This process is facilitated through a Database Management System (DBMS)

II. OBJECT ORIENTED DATABASES:

Object oriented databases are also called Object Database Management Systems (ODBMS). Object databases store objects rather than data such as integers, strings or real numbers. Objects are used in object oriented languages such as Smalltalk, C++, Java, and others. Objects basically consist of the following:

- Attributes - Attributes are data which defines the characteristics of an object. This data may be simple such as integers, strings, and real numbers or it may be a reference to a complex object.
- Methods - Methods define the behavior of an object and are what was formally called procedures or functions.

III. RELATIONAL DATABASES:

Relational databases store data in tables that are two dimensional. The tables have rows and columns. Relational database tables are "normalized" so data is not repeated more often than necessary. All table columns depend on a primary key (a unique value in the column) to identify the column. Once the specific column is identified, data from one or more rows associated with that column may be obtained or changed.

IV. WHEN TO USE OBJECT DATABASES:

Object databases should be used when there is complex data and/or complex data relationships. This includes a many to many object relationship. Object databases should not be used when there would be few join tables and there are large volumes of simple transactional data.

Object databases work well with:

- CAS Applications (CASE-computer aided software engineering, CAD-computer aided design, CAM-computer aided manufacture)
- Multimedia Applications
- Object projects that change over time.
- Commerce

V. OBJECT DATABASE ADVANTAGES OVER RDBMS:

- Objects don't require assembly and disassembly saving coding time and execution time to assemble or disassemble objects.
- Reduced paging
- Easier navigation
- Better concurrency control - A hierarchy of objects may be locked.
- Data model is based on the real world.
- Works well for distributed architectures.
- Less code required when applications are object oriented.

VI. OBJECT DATABASE DISADVANTAGES OVER RDBMS:

- Lower efficiency when data is simple and relationships are simple.
- Relational tables are simpler.
- Late binding may slow access speed.
- More user tools exist for RDBMS.
- Standards for RDBMS are more stable.
- Support for RDBMS is more certain and change is less likely to be required.

VII. OBJECT-ORIENTED DATABASE (OODB):

Object-Oriented Language Features: abstract data types in heritance object identity persistence support of transactions simple querying of bulk data concurrent access resilience security

OODB = Object Orientation + Database Capabilities

FEATURESTOBECONSIDERED:

persistence support of transactions simple querying of bulk data concurrency control resilience and recovery security version in gintegri typer form an issues

DATA MODEL STOBE CONSIDERED:

Complex object model Semantic data model such as Extended ER (EER) model, OPM model
not all oodb supports same object-orientation, same data model and same set of features

VIII. WHY OBJECT-ORIENTED DATABASE:

- Industry Trends: Integration and Sharing
- Seamless integration of operating systems, databases, languages, spreadsheets, word processors, AI expert system shells.
- Sharing of data, information, software components, products, computing environments.
- Referential sharing: Multiple applications, products, or objects share common sub-objects. (Hypermedia links are then used to navigate from one object to another) Object-oriented data bases allows referential sharing through the support of object identity and inheritance.

IX. ADVANTAGES OF OODB:

An integrated repository of information that is shared by multiple users, multiple products, multiple applications on multiple platforms.Italsosolves the following problems:

1. The semantic gap: The real world and the Conceptual model is very similar.
2. Impedance mismatch: Programming languages and database systems must be interfaced to solve application problems. But the language style, data structures, of a programming language (such as C) and the DBMS (such as Oracle) are different. The OODB supports general purpose programming in the OODB framework.
3. New application requirements: Especially in OA, CAD, CAM, CASE, object-orientation is the most natural and most convenient.

X. COMMERCIAL OODB:

SIM: Semantic Information Manager, UNISYS 1987.Supports semantic data model.Core system of the Info Exec Environment of UNISYS.Uses the Semantic Data Model of Hammer and McLeod 1981. User can define entity types that can inherit from one another. Attributes of entities are like functions from one entity to another.
Relational DB Extensions: Many relational systems support OODB extensions.
1. User- defined functions (dBase).
2. User-defined ADTs (POSTGRES)
3. Very-long multimedia fields (BLOB or Binary Large Object). (DB2 from IBM, SQL from SYBASE, Informix, Inter base)

XI. ALTERNATIVE OODB STRATEGIES:

1. Develop novel database data model or data language (SIM)
2. Extend an existing database language with object-oriented capabilities. (IRIS, O2 and VBASE/ONTOS extended SQL)
3. Extend existing object-oriented programming language with database capabilities (Gem Stone OPAL extended Small Talk)
4. Extendable object-oriented DBMS library (ONTOS)

XII. OODB QUERY LANGUAGE

ONTOS (from On to logic), O2 (from O2 Technology) and IRIS (from HP) all offer object-oriented extension of SQL. IRIS has Object SQL. Each entity type specifies one or more properties. Properties are functions that apply to the instances of the type. Entity types have extensions and can inherit from one another.
Example: The type PERSON can be created by: Create Type Person (Name char(20), Age integer Address Where address is another type. In IRIS, Name, Age and Address are called properties (functions) and apply to instances of type Person.

XIII. OODB QUERY LANGUAGE :

Object SQL Query: retrieve the name and state of all people who are older than 21:
Select Name(p), State(Address(p)for each Person p
where Age(p) > 21(Here we assume address has another property called "state") The user can compose the function application: P5(P4(...P1(P))...) in the select statement.

XIV. WHY EXTENDING SQL:

SQL is the most popular relational query language
SQL is also the only relational language that has a standard.
SQL is being promoted by many companies as the interface language for database engines and database servers. New applications developed in SQL extensions can easily call these servers for remote access. Developed by ODMG, Object Query Language allows SQL like queries to be performed on a OODB. Like SQL, it is a declarative language. Based loosely on SQL, OQL includes additional language constructs

which allow for object oriented design such as operation invocation and inheritance Query Structures look very similar in SQL and OQL but the results returned are different.

XV. ODBMS STANDARDS:

- Object Data Management Group
- Object Database Standard ODM6.2.0
- Object Query Language
- OQL support of SQL92

XVI. CONCLUSION:

Due to the weaknesses of OODBs discussed above, OODBs have not been able to keep up with the expectations of providing all the important features that targeted OODB application areas would like to use. The term OODB has become a misnomer for most current OODBs. Most current OODBs are closer to being merely persistent storage systems for some object-oriented programming language than database systems [26]. So, though the OODM is richer than the relational data model in many respects, the OODM has not matured enough, and to date, the weaknesses of OODB systems outweigh the achievement of OODB systems.

REFERENCES :

1. Abiteboul, S. and Bonner,A "Objects and views," in Proc. ACM SIGMOD Int. Conf. On Management of Data, pp. 238-247, 1991.
2. Atkinson, M., et. al., "The object-oriented database system manifesto," in Proc. Int. Conf. On Deductive and Object-Oriented Databases, 1989.
3. Atkinson, M., Bancilhon, F., DeWitt, D., Dittrich, K., Maier, D, and Zdonik, S. "The Object-Oriented Database System Manifesto," in Bancilhon et. al. (eds.), Building an Object-Oriented Database System: The Story of O2. Morgan Kaufman, 1992.
4. Bancilhon, F., "Object Oriented database systems," in Proc. 7th ACM SIGART/SIGMOD Conf., 1988.
5. Banerjee, J., et. al., "Data model issues for object oriented applications," ACM Trans. On Office Information Systems, vol. 5, no. 1, Jan 1987.
6. Banerjee, J., Kim, W., and Kim, K.C., "Queries in object oriented databases," in Proc. IEEE Data Engineering Conf., Feb. 1988.
7. Beech, D., "A Foundation for evolution and relational to object databases," in Proc. Extended Data Base Technology, Mar. 1988
8. 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.
9. Brown, A.W., Object-Oriented Databases, Applications in Software Engineering.New York: McGraw-Hill, 1991.
10. Cattell, R.G.G., Object Data Management, Object-Oriented and Extended Relational Database Systems.Reading, MA: Addison-Wesley, 1991.
11. Cherniack, M., "Form(ers) over Function(s): The KOLA Query Algebra," Technical Report, Brown University, December, 1995.
12. Cluet, S., et. al., "Reloop, An algebra based query language for an object-oriented database system," in Proc. 1st Int. Conf. On Deductive and Object Oriented Databases, Dec. 1989.
13. Cruz, I.F., DOODLE: A visual language for object-oriented databases. In Proc. ACM SIGMOD Int. Conf. On Management of Data, pp. 71-80, 1992.