Scientific Journal Impact Factor (SJIF): 1.711

e-ISSN: 2349-9745
p-ISSN: 2393-8161

International Journal of Modern Trends in Engineering and Research
www.ijmter.com

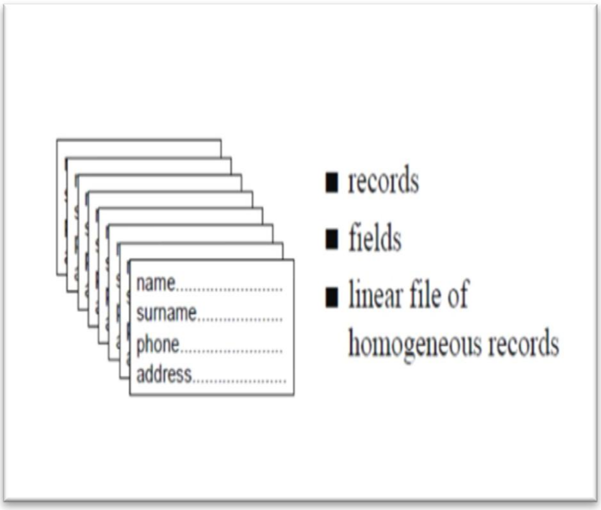
Survey of Object Oriented Database

Pratiksha P. Gaurkhede[1], Prof.P.J. Pursani[2]
[1]*CSE Dept, HVPM COET, Amravati*
[2]*Assistant Professor, CSE Dept,HVPM COET, Amravati*

Abstract- The technology of object oriented databases was introduced to system developers in the late 1980's. Object DBMSs add database functionality to object programming languages. A major benefit of this approach is the unification of the application and database development into a seamless data model and language environment. As a result, applications require less code, use more natural data modeling, and code bases are easier to maintain.

I. Introduction

1.1. Database



We are all familiar with concepts like records and fields. So a stack of cards like the one pictured above is perhaps our mental image of a database. Thinking of a linear file of homogeneous records as the archetype for a database is as reductive as thinking of a skateboard as the archetype for a roadworthy vehicle. A flat file is only a very, very restricted form of database.

A database is 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)

1.2. Relational databases

Relational databases can be accessed by means of object oriented programming languages. However that does not turn a relational database into an object oriented one. In relational database model data are logically organized in two dimensional tables. Each individual fact or type of information is stored in its own table. The relational database model was developed using a branch of mathematics called set theory. In set theory a two dimensional collection of information is called a relation. A relational database management system allows users to query the tables to obtain information from one or more tables in a very flexible way. The relational database is attractive from a user's standpoint because end users often think of the data they need as a table. The capability of a relational database management is to handle complex queries is important. Althoughthe relational database is considered to be a dramatic improvement over the network ad relational models, it have two disadvantages. First a relational database requires much more computer memory and processing time than the earlier models. Increases in computer processing speeds as well as a steady decrease I hardware costs have reduced the impact of this first disadvantage. The second disadvantage is that the relational database allows only text and numerical information to be stored in the database. It did not allow the inclusion of complex object types such as graphics, video, audio, or geographic information. The desire to include these complex objects in databases led to the development of object oriented databases.

1.3 Object-Oriented Databases:

An Object oriented databases is a database that subscribes to model with information represent with object. Object oriented database is the niche offering in the relational database management system field and not successfula well known as mainstream database engines. Object oriented database systems enables direct access to objects defined in the programming language in question and the storage of such objects in the database without conversion. It is precisely this that is not possible with relational database systems, in which everything must be structured in tables. Both simple and complex objects can be stored in an object oriented database model. In object oriented database other types of data can be stored. Object oriented database includes abstract data types that allow the users to define the characteristics of the data to be stored when developing an application. This overcomes the limitations of relational databases. Relational databases limit the types of data that can be stored in table columns. Instead of tables an object oriented database stores the data in objects. An object can store attributes and instructions for actions that can be performed on the object or its attributes. These instructions are called encapsulated methods. Objects can be placed in a hierarchy attributes from objects higher in the hierarchy. Many researchers have argued that the object oriented databases are superior to relational database most organizations still use relational.

II. History:

Computerized Databases evolved with DBMS in the 1960s with the availability of disks and drums to provide an easy alternative to maintaining large amount of diverse information. In the 1970s the main objective of database technology was to make the data independent of the logic of application programs to enable concurrent access to different application programs. The first generations of databases were navigational, where applications accessed data through record pointers moving from one record to another. This was a precursor to the IBMs hierarchical model (IMS System) and network model. This was followed by the relational model which placed the emphasis on content rather than links for data retrieval. This kind of database the most widely till date. Relational models were limiting in the kind of data that could be held, the rigidity of the structure, and the lack of support for new data types such as graphics, xml, 2D and 3D data. In the 1980swith the advent of Object Oriented methodologies and languages, integration of database capabilities with object oriented programming language provided a unified programming environment. This led to the development of OODB and OODBMS where objects are stored in databases rather than data such as integers, strings or real numbers. Relational models were limiting in the kind of data that could be held, the rigidity of the structure, and the lack of support for new data types such as graphics, xml, 2D and 3D data. In the 1980s with the advent of Object Oriented methodologies and languages, integration of database capabilities with object oriented programming language provided a unified programming environment. This led to the development of OODB and OODBMS where objects are stored in databases rather than data such as integers, strings or real numbers.

III. Features of Object Oriented database

3.1 Object identity

Every instance in the database has a unique identifier (OID), which is a property of an object that distinguishes it from all other objects and remains for the lifetime of the object. In object-oriented systems, an object has an existence (identity) independent of its value. Object oriented database systems integrate the object identity in the database with the identity of objects in memory. If you store an object then it knows if it corresponds to an object in the database, and when you retrieve an object it knows if the object has already been loaded into program memory. There is no need for the programmer to maintain the relationship between database objects and objects in memory.

3.2 Encapsulation

Encapsulation means that code and data are packaged together to form objects, and that the implementation of these objects can be hidden from the rest of the program. Object-oriented models enforce encapsulation and information hiding. This means, the state of objects can be manipulated and read only by invoking operations that are specified within the type definition and made visible through the public clause. In an object-oriented database system encapsulation is achieved if only the operations are visible to the programmer and both the data and the implementation are hidden. There are two views of encapsulation: the programming language view (which is the original view since the concept originated there) and the database adaptation of that view. The idea of encapsulation in programming languages comes from abstract data types. In this view, an object has an interface part and an implementation part. The interface part is the specification of the set of operations that can be performed on the object. It is the only visible part of the object. The implementation part has a data part and a procedural part. The data part is the representation or state of the object and the procedure part describes, in some programming language, the implementation of each operation. The database translation of the principle is that an object encapsulates both program and data. In the database world, it is not clear whether the structural part of the type is or is not part of the interface (this depends on the system), while in the programming language world, the data structure is clearly part of the implementation and not of the interface.

3.3. Inheritance

Inheritance is a powerful mechanism which lets a new class be built using code and data declared in other classes. This allows common features of a set of classes to be expressed in a base class, and the code of related class. Classes which are derived from other classes can be stored with one operation. The database system must know the class hierarchy and manage object storage accordingly. The relational database systems cannot store functions you must ensure that the function hierarchy is restored when you read objects from the database. This usually requires type information to be stored in the database to enable your application to build objects appropriately before loading their data.

3.4. Polymorphism

Relational databases offer no support for polymorphism. We must be careful to store enough information about our data types to enable you to reconstruct objects properly before loading their data. Suppose you have a variety of Items, but the rules for computing their total price differs. For instance, medical items might be exempt from sales tax in an American database system, or a German database system might compute value added tax differently for different items. You still want all Items to have a function called Total Price (), but instead of one Total Price () function we now want different functions depending on the class.

3.5. Combination of object oriented programming and database technology:

The object oriented database technology combines object oriented programming with the database technology to provide an integrated application development system. There are many advantages of including the definition of operations with the definition of data. First the operations which are defined are applied and are not depending on the specific database application running at the moment. Second the data types can be extended to support complex data.

3.6. Improves productivity: Inheritance allows programmers to develop solutions to complex problems by defining the new objects in terms of defined objects in the past. Polymorphism and dynamic binding allows programmers to define operations for one object and then to share the specification of the operation with other object. These objects can further extend this operation to provide behaviors that are unique to those objects.

3.7 Data access:

Object oriented databases represent relationships explicitly supporting both navigational and associative access to information.

3.8 Extensibility:

OODBMSs allow new data types to be built from existing data types. The ability to factor out common properties of several classes and form them into a super class that can be shared with subclass can greatly reduce redundancy within system is regarded that one of the main advantage of object orientation.Further,the reusability of classes promotes faster development and easier maintenance of database and its applications.

3.9 Multiple large varieties of data types:

Unlike traditional databases(such as Hierarchical, Network or Relational)the object oriented database are capable of storing different types of data, for example pictures voice video, include text numbers and so on.

IV. Usage of object oriented databases:

Object oriented databases are today rarely used, but in some cases they are used. It is where the data is very complex, in the sense that the data is not primitive data types, or if there is a need for storing behavior, that the need for object oriented databases arises. For example if a company has developed several software agents and want to reuse them in different ways. 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.

V. Advantages

5.1 Composite Objects and Relationships:

Objects in an OODBMS can store an arbitrary number of atomic types as well as other objects. It is thus possible to have a large class which holds many medium sized classes which themselves hold many smaller classes, ad infinitum. In a relational database this has to be done either by having one huge table with lots of null fields or via a number of smaller, normalized tables which are linked via foreign keys. Having lots of smaller tables is still a problem since a join has to be performed every time one wants to query data based on the "Has-a" relationship between the entities. Also an object is a better model of the real world entity than the relational tuples with regards to complex objects. The fact that an OODBMS is better suited to handling complex, interrelated data than an RDBMS means that an OODBMS can outperform an RDBMS by ten to a thousand times depending on the complexity of the data being handled.

5.2. Class Hierarchy:

Data in the real world is usually have hierarchical characteristics. The ever popular Employee example used in most RDBMS texts is easier to describe in an OODBMS than in an RDBMS. An Employee can be a Manager or not, this is usually done in an RDBMS by having a type identifier field or creating another table which uses foreign keys to indicate the relationship between Managers and Employees. In an OODBMS, the Employee class is simply a parent class of the Manager class.

5.3. No Impedance Mismatch: In a typical application that uses an object oriented programming language and an RDBMS, a significant amount of time is usually spent mapping tables to objects and back. There are also various problems that can occur when the atomic types in the database do not map cleanly to the atomic types in the programming language and vice versa. This "impedance mismatch" is completely avoided when using an OODBMS.

5.4. No Primary Keys: The user of an RDBMS has to worry about uniquely identifying tuples by their values and making sure that no two tuples have the same primary key values to avoid error conditions. In an OODBMS, the unique identification of objects is done behind the scenes via OIDs and is completely invisible to the user. Thus there is no limitation on the values that can be stored in an object.

5.5. One Data Model: A data model typically should model entities and their relationships, constraints and operations that change the states of the data in the system. With an RDBMS it is not possible to model the dynamic operations or rules that change the state of the data in the system because this is beyond the scope of the database. Thus applications that use RDBMS systems usually have an Entity Relationship diagram to model the static parts of the system and a separate model for the operations and behaviors of entities in the application. With an OODBMS there is no disconnect between the database model and the application model because the entities are just other objects in the system. An entire application can thus be comprehensively modeled in one UML diagram.

5.6. Support for long transactions: Current relational DBMSs enforce serializability on concurrent transactions to maintain database consistency. OODBMSs use different types of long duration transaction that are common in many database applications.

5.7. Better support for applications like software engineering or computer aided design (CAD).Arguably better performance; though benchmarks have mainly been applied in areas like engineering support to which OODBMS are better suited.

VI. Recent working on OOD:

These are currently using an OODBMS to handle mission critical data :

1.The Chicago Stock Exchange manages stock trades via a Versant ODBMS.

2.Radio Computing Services is the world's largest radio software company. Its product, Selector, automates the needs of the entire radio station -- from the music library, to the newsroom, to the sales department. RCS uses the POET ODBMS because it enabled RCS to integrate and organize various elements, regardless of data types, in a single program environment.

3.The Objectivity/DB ODBMS is used as a data repository for system component naming, satellite mission planning data, and orbital management data deployed by Motorola in The Iridium System.

4. The Object Store ODBMS is used in South West Airline's Home Gate to provide self service to travelers through the Internet.

5.Anjou University Medical Center in South Korea uses Inter Systems' Cache ODBMS to support all hospital functions including mission-critical departments such as pathology, laboratory, blood bank, pharmacy, and X-ray.

6. The Large Hadron Collider at CERN in Switzerland uses an Objectivity DB. The database is currently being tested in the hundreds of terabytes at data rates up to 35 MB/second.

7.As of November, 2000, the Stanford Linear Accelerator Center (SLAC) stored 169 terabytes of production data using Objectivity/DB. The production data is distributed across several hundred processing nodes and over 30 on-line servers.

VII. Literature Review

1. A Gentle Introduction to Relational and Object Oriented Databases[1]

This report is an exact reproduction1 of his 1995 material. It consists of three parts: a talk on relational databases, a talk on object oriented databases and a commented bibliography on object oriented databases. The talks are intended as one hour introductions for an audience of computer professionals, assumed to be technically competent but not familiar with the topics discussed.

2. Object Oriented Database[2]

This presentation on Object Oriented Databases. It gives a basic introduction to the concepts governing OODBs and looks at its details including its architecture, the query languages used etc. A contrast between OODBs and RDBs is also presented. The reader will gain insight into databases, data models, OODB architecture, Object Query Language, OODBMS.

3. Advantages of object oriented over relational database on real life application- [3]

In this paper, author introduces the advantages of the object oriented database.

4. Object oriented databases a natural part of object oriented software development. [4]

The presentation on Object Oriented Databases gives a basic introduction to the concepts governing OODBs and looks at its details including its architecture, the query languages used etc. A contrast between OODBs and RDBs is also presented.

VIII. Conclusion

Now a day's there is a trend of Relational databases, but as we are moving towards E-commerce form traditional commerce there is a need of considering multimedia data which is frequently found on internet which is not supported by relational database. Thus there is need for object oriented database which is providing an alternative to relational database for the representation, storage and access of non-traditional data forms that were increasingly found in advance application like Geographic Information System, CAD/CAM etc. Object Oriented database can handle large collection of complex data including user defined data types and supports for inheritance, polymorphism etc.

References

[1][HakanButuner]volum 5,2012*Industrial Management and Engineering Co., Istanbul, Turkey*

[2] [Arjun Gopalkrishna & Bhavya Udayashankar 2007]

[3].[Won Kim]1990 ACM *Microelectronics and Computer Technology Corporation*3500 West Balcones Center Drive

[4].[IEEE] IEEE POSIX 1003.4 Working Group, *POSIX Srd1003.lb-1993 (P1003.4)Real-Time Extension.* 1994, IEEE.

[5].[KRUP94 Kmpp P., et al.] Real-Time Processing and CORBA.presented at the OOPSLA 94 Conference Workshop on CORBA, Portland, OR, 1994.

[6].[Kappel, G., RauschSchott,S., Retschitzegger, W.] BottomUpDesign of Active ObjectOrientedDatabases. Communications of the ACM, 2001

[7].[Fayad, Mohamed E., Schmidt, DouglasC]ObjectOrientedApplication Frameworks. Communications of the ACM, 1997.Loomis, Mary E. S. ODBMS versus Relational. JOOP Focus on ODBMS, 1992

[8].[cadf]The Committee for advance DBMS FUNCTION, Third Generation Database System Manifeto, In *Computer Standards and Interfaces* 13 (1991), pages 41-54. North Holland. Also appears in *SIGMOD Record* 19:3 Sept, 1990.

[9].[date] C. J. Date, “An introduction to database systems”, Vol. 1, 6th ed,*Addison-Wesley*, 1995

[10].[W. KIM AND LOCHOVSKY (EDS]),*Object-Oriented Concepts, Databases, and Applications,* Addison-Wesley (Reading MA), 1989

[11].[ullman89] JEFFREY D. ULLMAN,*Principles of Database and Knowledge-based Systems, Vol.2,*Computer Science Press, 1989(2nd volume of 2-volume textbook.)

[12].[Kofler Michael, Ph.D.] Thesis, *R-trees for Visualizing and OrganizingLarge 3D GIS Databases,* TechnischenUniversität Graz,July 1998,www.icg.tugraz.ac.at/kofler/thesis,

[13].[Kuzniarz, Ludwik and Staron,Miroslaw,]*Customisation of UnifiedModeling Language for Logical Database Modeling*, Research report no2002:12, Department of Software Engineering and Computer Science,Blekinge Institute of Technology.