

History of Object Oriented Databases and its Significance

Jayakranthi Katam

School of Engineering, University of Bridgeport

126 Park Ave, Bridgeport, CT 06604

Email: Jayakranthi@gmail.com

Abstract—This paper discusses about the principles and understanding of the Object Oriented Programming (OOP) and the features that supports idea to make it distinctive among other programming concepts and then it explains about the object oriented database system which is one of the dynamic patterns of the today's database system. The structure of object oriented databases may not stay consistent and may change about whether to a vast degree and also we can know the fundamental of the database system and up to where it extended now. The various types of models discussed in this research are Network model, relational model and the object model. The best example for the object model is the database with C++ or java programming languages as they don't need to decipher into a database sublanguage like SQL etc. The culmination to this is that any data structure that you can notice in Java or C++ can be stored directly without interpretation in an ODBMS

Keywords—Database, Models, Methods, Objects, Encapsulation.

I. INTRODUCTION

Object oriented programming is a type of programming based on the concept of Objects. Objects in object-oriented programming can be defined as data structures that contain data namely attributes in the form of fields. Further, object-oriented programming uses Methods that are typically procedural form of codes (Kindler & Kirvy, 2011). Most of the renowned object-oriented programming languages utilize class-based late. Approval particular in object oriented databases is constantly progressively researched as of late by numerous analysts. Fundamental goal of OODBMS, is to give reliable, data free, secure, controlled and extensible data management services to help the object oriented model. They were made to handle enormous and complex data that relational databases proved unable. The most critical k is the joining of item situated programming with database technology, which gives an incorporated application advancement framework. Object oriented programming brings about 4 objects. An object is an instance of the class and the concept of class-based objects provide developers with the convenience of creating multiple instances; also, each instance can have its own data type. To perceive

better the idea behind object-oriented programming, let us approach the concept by relating it to building a community of houses.

The classes in OOP can be compared to a blueprint. Alike the blueprint, a class, once created can be reproduced multiple times. Each reproduction of a class is called an object. Furthermore, the actual process of building the house can be related to methods. Programmers who use OOP must give priority to the data that they want to operate with rather than the logic involved with the manipulation. Therefore, it is important to determine all the objects and the relation between different objects and this process is known as data modeling. Once the programmers determine the object, they must define the type data that the object encloses and any logic sequence that can manipulate the data if any. Individual modules of logic sequences are known as methods (Rouse, 2008).

Object oriented database system is one of the dynamic patterns of today's database system, henceforth data assurance in object oriented databases has turned into a dynamic examination range as of principle qualities: legacies, data encapsulation, object character, and polymorphism. An OODBMS is the aftereffect of joining object oriented programming standards with database administration standards. Object oriented programming ideas, for example, encapsulation, polymorphism and legacy are implemented and database administration ideas, for example, the ACID properties (Atomicity, Consistency, Isolation and Durability) which prompt framework honesty, help for a specially appointed for an ad hoc query language and secondary storage management systems which consider overseeing a lot of information. The Object Oriented Database Manifesto particularly records the accompanying gimmicks as obligatory for a framework to backing before it can be called an OODBMS. An OODBMS ought to have the capacity to store objects that are about undefined from the sort of articles underpinned by the target programming language with as meager limit as could be expected under the circumstances. Object-oriented programming supports various features that make it unique from rather conventional programming languages and they are described in the following section.

Abstraction: Data abstraction is the process of defining the properties of an object so that the object

in view is unique from all other objects (Raymondlew, 2005).

Encapsulation: Encapsulation of data adds a header to the data. It's like putting the data within an envelope with the recipient and the sender's addresses written on it. When an object is created, the complexity of the internal code and its functioning can be hidden in an object. Also through encapsulation if a program is changed the encapsulated part of the program is not affected by changes to any other part of the program, which reduce cascading on other parts of the object. For example: In a program called bank account which consists of elements like name of account holder, account number, date of birth etc. and the underlying object consist of details like deposit date, withdrawal date etc. In this scenario it is a good idea to encapsulate the first three components; the account holder, account number and date of birth of the customer who is opening an account as they are created only once for a customer and also it is also very important information. However, the other three components like deposit date and withdrawal date can be multiple times and it is not an important information therefore it is not required to encapsulate such kind of data (Madan, 2004).

Inheritance: Inheritance in OOP can be associated to a child inheriting a property from his parents or ancestors in real world. A child class in OOP inherits all of the parent class properties to which it is associated. More properties can be added to the child class; however, the child class would automatically inherit any characteristic added to the parent class. This feature enables users to utilize functionalities of a class multiple times without duplicating it where otherwise each class had to be implemented independently. The parent or in technical terms the superclass has only the data and the methods that are common to all the classes that were to inherit from this class (Raymondlew, 2005).

Polymorphism: Polymorphism is the feature that utilizes same name for all of the methods but each method is distinct and has different functionalities. There are two-types of polymorphism namely Overriding and Overloading is done by the compiler during compile-time and decides which method is to be executed depending upon the arguments and its data types passed to the method. Overriding on the other hand is done at run-time depending on the dynamic type of an object (Raymondlew, 2005).

II. HISTORY OF OBJECT ORIENTED DATABASES

The structure of object oriented databases may not stay consistent and may change about whether to a vast degree. We need for different reason, for example, to amend mistakes in the database outline, to include new peculiarities amid incremental configuration and to reflect changes in the structure of this present reality antiques demonstrated in the database. Automated databases advanced with DBMS in the 1960s with the accessibility of circles

and drums to give a simple option to keeping up huge measure of differing data.

In the 1970s the fundamental target of database engineering was to make the information free of the rationale of use projects to empower simultaneous access to diverse application programs. The pilot generation of databases were navigational, where applications got to access information through (IMS system) and network model.

This was trailed by the relational model which set the attention on substance as opposed to connections for information recovery. This sort of database is mostly used these days. Relational models were constraining in the sort of information that could be held, the inflexibility of the structure, and the absence of backing for new information sorts, for example, representation, XML, 2d, and 3d information. In the 1980s with the approach of object oriented methodologies and languages, reconciliation of database abilities with article situated programming dialect gave a unified programming environment. This prompted the advancement of OODB and OODBMS where objects are put away in databases rather than data, for example, numbers, strings or real numbers.

The sort of a database is chosen by the data model utilized within the configuration of the database. Data models are information structures which portray how information are spoken to and got to. Data models must be straightforward and natural to empower applications. A portion of the work proposed a security model for object oriented databases, some displayed a formal determination of an approval model for object oriented database system. A database is a sorted out accumulation of related information held in a computer or an information bank, which is intended to be available in different ways. The information inside a database is organized to ensure data is stored conveniently to ensure retrieval and preparing recovery instruments. Customers cooperate with databases through inquiries to make, recover, redesign and erase information inside a database. This procedure is encouraged through a Database Management System (DBMS).

Furthermore, a DBMS likewise gives tools to support, for example, running security checks, guaranteeing information integrity, recovery and backups. Object oriented database systems are an endeavor to tackle the issues specified and still keep up the focal points of database systems. Object oriented databases treat every entity as a different object. A get together made out of a few parts, hence, can allude specifically to its segments rather than expressly partner some exceptional identifier with every part in some connection. Database and its administration characterize distinctive objects, they are as one and are critical for business in all parts of the present day world be it in innovation situated organizations or clinics and medicinal services. The significant types of data models in the history of databases are:

Various leveled model which contain information composed into a tree-like structure. This helps parent-child relationship between data same as tree data structure where object types are represented by nodes and their similarities are represented by arcs. This model is prohibitive in that it just permits one to numerous relationship (a guardian can have numerous children yet a child can just have one parent)

Network model is like the various leveled model in representation of information however takes into consideration more prominent adaptability in information get to as it backings numerous to numerous connections.

Relational model arranges information into two dimensional shows known as relations (tables) and every connection comprise of columns and sections. An alternate, significant qualities of relational model is that of keys-assigned segments in a connection used to request information or build relations.

Object model means to diminish the overhead of hanging over data representation in the database to an application particular representation. Unlike to a traditional database, an object model takes into consideration information constancy and capacity by assorting objects in the databases. The connections between different objects are innate in the structure of the objects. This is fundamentally utilized for complex information structures, for example, 2d and 3d illustrations which should overall be leveled before capacity in a relational database.

III. CONCLUSION

From my viewpoint, there are two essential benefits in utilizing object oriented databases, Both profits reflect an essential thought - when you utilize an ODBMS, the way you utilize your data is the way you store it. The main profit can be found being developed. When you utilize an ODBMS, you will compose less code than if you were writing with a RDBMS. The explanation behind the more diminutive measure of code is basic - when you are utilizing Java or C++ - you won't need to decipher into a database sublanguage such as structured query language. The result is less code. By and large, this code be to the extent that 40 percent less. The culmination to this is that any data structure that you can notice in Java or C++ can be stored directly without interpretation in an ODBMS. This is an essential thought. You can store exceptionally unpredictable data structures specifically in an ODBMS.

The second profit, which is connected, happens in production. In the event that you are working with complex information, an ODBMS can provide for you execution that is ten to a thousand times quicker than a RDBMS. That is on account of when the information is perused off the disk, it is in the configuration that Java or C++ uses. No interpretation is required. The extent in execution increase relies on upon the unpredictability of your data.

ODBMS is used when you have a business requirement for elite on complex data. In any case, what does that mean precisely? Quickly, when you meet a business need, you either make or save money. Elite, in the setting of an ODBMS, implies that you truly could see an execution change of between ten to a thousand times in examination with a RDBMS. Lastly, you can make sure that you have complex data if your data resembles an associated chart, for example, a bill of materials or a diagram that has numerous convergence elements. For instance, how about we say you have a robotized stock exchanging application. The application gets a live sustain of stock information. It utilizes that data with a chronicled database and applies a calculation that figures out whether any activity ought to happen focused around the live sustain. The quicker the database inquiry is, the shorter time between accepting information from the live feed. Expecting the algorithm is right, this implies the stock exchanging application will either be profiting or losing less cash. As it were, this is a case where there is a business requirement for superior on complex data.

REFERENCES

1. Rashid, A., & Sawyer, P. (2005). A database evolution .Journal Of Software Maintenance & Evolution: Research & Practice, 17(2), 93-141. doi:10.1002/smj.310
2. Retrieved from ComputerNetworkingSimplified.com.
3. Kindler, E., & Kirvy, I. (2011). Object-oriented simulation of systems 313-314.
4. Raymondlew, A. (2005, July 19).4 major principles of Object-Oriented Programming. Retrieved March 22, 2015, from Codebetter.com: <http://codebetter.com/raymondlewallen/2005/07/19/4-major-principles-of-object-oriented-programming/>
5. Rouse, M. (2008, August). Object-oriented programming (OOP) definition.
6. Scott, M. L., & Kaufmann, M. (2006). Programming language pragmatics (2 ed.).
7. <http://www.studymode.com/essays/Oodbms-767362.html>
8. http://www.service-architecture.com/articles/object-oriented-databases/odbms_faq.html#Can