Evolution of Object-Oriented Database Systems

Hibatullah Alzahrani[1]

[1] Clark Atlanta University

Abstract

Data bases are quintessential part of most modern web and mobile applications. In most part, relational databases dominate the database market but the evolution of object-oriented databases has provided users and developers with an alternative option. Object-oriented databases provide a number of advantages over relational databases like ease of extensibility, custom data models, provision for modelling complex data structures and faster access time. But they do lack in certain areas and have no strict standards and implementation mostly depends upon the vendor. Nevertheless, object-oriented databases are slowly finding their way into database market, especially in the area of large-scale databases. But the long history of relational databases keeps them alive as tough competitor and the future seems to be going towards object-relational databases.

Index terms— object-oriented, database, relational, data- base management system, evolution, advantages, disadvantages.

1 Introduction

atabases are the nuts and bolts of the modern information systems. Every major application on Internet and smartphones uses them in one way or another. They are ubiquitously used in data centers and for maintaining records in hospitals, universities and all kinds of government and private institutions. Strictly speaking, there is a distinction between a database and a database management system (DBMS) -database is an organized collection of data whereas DBMS is a software which interacts with the database and the user and acts as an interface between them. But usually database is used to refer to both the database itself and the DBMS. Most commonly used DBMS is Relational DBMS (RDBMS) which is based on relational data model in which data is stored as tables or "relations" consisting of rows and columns. With the advent of object-oriented programming paradigm and the rise of object-oriented programming languages, the concept of object-oriented databases was conceived in which data is represented as objects rather than as tables. Figure 1 provides a mapping between the relational and objectoriented database model. In this article, we will briefly discuss what object-oriented databases are, trace the evolution of object-oriented databases, their use in modern systems and their advantages and disadvantages over traditional Relational Databases.

Author: e-mail: mesfer66@gmail.com Object-oriented databases are designed and built according to the object-oriented paradigm in which everything is modeled as objects including the data. This type of data model helps in tackling complex data structures, for instance multimedia content, in a more natural way and provides a seamless transition from design to conception. According to object-oriented database system manifesto [1], an Object-Oriented Database Management System (OODBMS) must satisfy two criterion: i) It should be a Database Management System (DBMS) ii) It should be an object-oriented system

The first criterion means that the OODBMS should provide the five features which are must for any database system -persistence, concurrency, data recovery, secondary storage management and ad hoc query facility. The second criterion means that the database system should support all the requisite features of an object-oriented system like encapsulation, complex objects, inheritance, polymorphism, extensibility etc. Hence the data in OODBMS is represented as collection of interacting objects instead of collection of inter-related tables. Usage of

object-oriented concepts like polymorphism and inheritance make the interaction between the objects a trivial task. Figure 2 provides an example of how the same data, customer account information for a banking system, is represented in two different formats. Whereas data is stored as tables in the relational database and we need to relate of "join" tables to perform a query, it is stored as a collection of objects in object-oriented database, and query can be easily performed by following the pointer from parent object to its children.

2   Evolution of Object-Oriented Databases

The term "object-oriented database system" was first introduced in 1985 in [2] and [3]. Orion Research Project at MCC headed by Won Kim was the first major project initiated for the development of OODBMS and several papers were published during its course, best of which were compiled by Won Kim in the form of book [4] in 1990. Meanwhile Servo-Logic [5] began work on one of the first commercial projects which was later renamed as Gem Stone [6] in 1995 and was based on Small Talk object-oriented language. Another Lisp-based system, Graphael, was introduced at around the same time in France and it was followed by O2 [7] in 1992 which was later acquired by IBM. Tom Atwood at Onto logic produced V base which later became ONTOS and was the first to support C++. Drew Wade at same time produced Objectivity/DB [8], **??**9]. In 1989, first OODBS manifesto [1] was published by Malcolm Atkinson which criticized relational model for being inadequate in meeting the demands of new applications and laid down the criterion for the objectoriented database system. A year later, second manifesto [10] was published, which went the opposite way by supporting relational model and wanting to support SQL for OODBS. The third and final manifesto was published in 1995 [11] and it presented OO model as an extension of relational model and proposed to extend the relational model to incorporate the objectoriented characteristics by allowing for custom userdefined data types.

Meanwhile, work on standardization of OODBMS began in 1991 when Rick Cattell of Sun Microsystems formed a consortium of 5 major OODBMS vendors, named ODMG (Object Database Management Group). As a result of these efforts for standardization, standards were published for ODL (Object Definition Language), OQL (Object Query Language) and OML (Object Manipulation Language). First standard was released in 1993 [12], which was mostly designed with C++ in mind as primary object-oriented language but in the final release 3.0 in 2001 [13], Java bindings were added to the standard to support Java and later bindings for Small Talk were also added. Afterwards, the ODMG was disbanded. Meanwhile, some objectoriented features were also included in SQL: 1999 [14] and were then revised in its 2003 version and then in the latest 2011 version [15]. SQL: 2011 supports Object Language Bindings (SQL/OLB) for performing queries to Object-oriented databases and various other object oriented features.

In 2004, db4o [16] was released as the first free open-source OODBMS and it was the first DBMS to implement native queries in the programming language itself like Java and C#. Also Perst and DTS/S1 were made available under dual open-source and commercial licenses. In 2005, Microsoft implemented Native Queries in its .NET framework by introducing LINQ and DLINQ. As a result, many object-oriented languages now support native queries. Though this is not the same as a full-fledge open-source database system, but it does provide the user or developer with an alternative option other than commercial databases, important since commercial OO databases are usually costlier than relational ones.

3   Global Journal of Computer Science and Technology

Volume XVI Issue III Version I

4   Advantages of oo Databases

Since Relational Databases have been the norm of the day for a long period of time, any new emerging technology is compared against this benchmark to measure its usefulness. Some of the advantages of OODBMS over traditional RDBMS are: a) Complex Data Models

In relational databases, all data is modeled as relations or tables and it is extremely complicated to model complex data structures. Object-oriented databases model all data as objects which can model even complex data structures very easily. In fact, one of the very first application for object-oriented databases were graph-based data structures which were modelled easily using object-oriented concepts but were a nightmare to model in relational databases.

5   b) Real-World Modelling

One of the strengths of object-oriented paradigm is its ability to model the real-world objects in a natural way. This becomes useful in maintaining data when we can store it as an object rather than a table and then perform manipulations on it. For instance, a very important application of OO databases is maintaining multimedia content. While using relational databases, it is cumbersome to store this content in form of tables but OO database can model the whole multimedia document as an object and store it easily. c) Extensibility (Provision of New Custom Data Types)

OO databases provides the ability to form new data types from existing ones using the object-oriented concepts of inheritance and polymorphism. No such feature is available in relational databases.

6 d) No Impedance Mismatch

In OODBMS, there is no mismatch between data represented by database system and the data representation required by application. We can model our data as required by the application and subsequently use it directly in our application. This is in contrast to RDBMS in which all data has to be stored in form of tables and then at runtime, data is manipulated into the form required by the application. e) Ease of Design and Implementation Relational Data Models are not very descriptive in nature and in general, the database is first designed using Entity Relationship (ER) model and then implemented using relational model. Object-oriented databases do away with this hassle by designing the data directly as objects and then implementing it as such.

f) Faster Data Access and Improved Performance OODBMS are usually much faster than relational ones since they have a many-to-many relationship and objects can be accessed using pointers only. Furthermore, there is no need for 'join' as objects are linked through pointers and any specific object can be found following the chain of pointers. g) Easy handling of very large data Object-oriented databases can very easily handle very large amounts of complex data and as a result, very large databases are often built using OODBMS. In fact, world's largest database is an OODBMS -Stanford Linear Accelerator Center (SLAC) BaBar database **??**17] uses Objectivity/DB [9] and currently sizes almost 900 TB.

IV.

7 Disadvantages of oo Databases

Despite all its advantages, OODBMS are still not as popular as relational databases due to following reasons: a) No Universal Data Model Relational databases have a fixed data model in which data is always represented in the form of tables. No such standard exists for object-oriented databases in which data is modeled as custom objects and depends on the type of data and need of application.

8 b) No Standard Query Language

There is no standard query language for OODBMS like SQL for Relational Databases. Even though ODMG standardized OQL (Object Query Language) but it is still widely unimplemented. More recently, trend has been towards implementing Native Queries in programming languages like LINQ in C# or the database vendor provides separate bindings for most popular object-oriented languages like Java, C++ etc.

9 c) No Mathematical Foundation

Relational Databases are based on the solid foundations of Relational Algebra and Relational Calculus. No such mathematical foundation exists for object-oriented databases. d) Lack of Ad hoc Queries or Closure Closure is a property of relational databases which enables nested queries where new tables are created by joining existing ones and then querying the new table. Since there are no joins in OODBMS, there are no nested queries and the nature of query that can be performed is highly dependent on the design of database. Hence strictly speaking, some OODBMS can violate an essential criterion for database management systems which requires that all database management system should support ad hoc query.

V.

10 Conclusion

Object-oriented databases have been around for quite some time now but they haven't found widespread acceptance like relational databases. They [1] [2] [3]

1

Figure 1: Figure 1 :

2

Figure 2: DFigure 2 :

Object-Oriented Data Model

karl: Customer
sName=Karl
fName=Meier

: BankAccount
accNo=395382

: BankAccount
accNo=824432

berlin: Address
city=Berlin

leipzig: Address
city=Leipzig

Relational Data Model

Customers

ID	SName	FName
1	Karl	Meier

Addresses

ID	City	CID
1	Berlin	1
2	Leipzig	1

BankAccounts

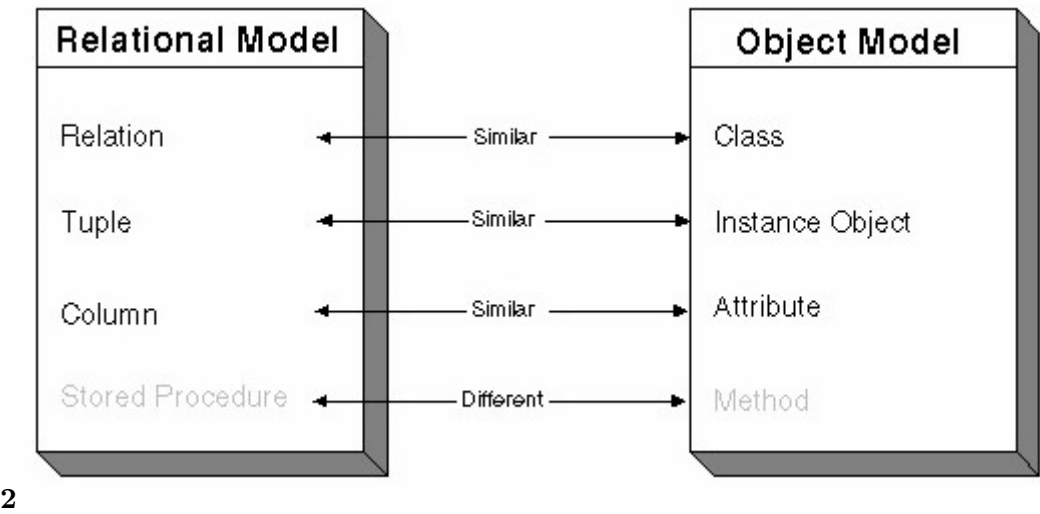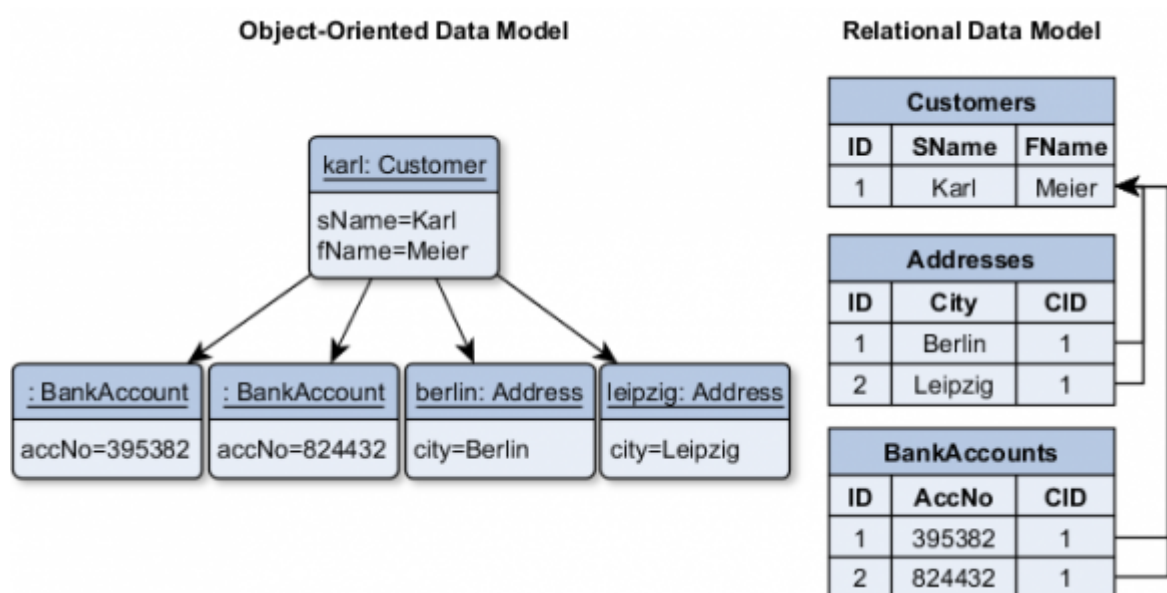| ID	AccNo	CID
1	395382	1
2	824432	1

Figure 3: Global

data structures. But still, they are not as ubiquitous as relational databases. The present trend is towards incorporating the good features of both types of databases forming a hybrid one, called Object-Relational Database, which is intrinsically a relational database with object-oriented features. Even the modern standards of SQL provide some object-oriented features. Thus integration of object-oriented features in relational databases highlight their importance and the competition they pose. Already, very big databases are being designed using object-oriented paradigm and as databases become larger and larger, it is inevitable that the trend would go towards object-oriented databases in future.

[Melton] *Advanced SQL, 1999: Understanding Object-Relational and Other Advanced Features*, Jim Melton .

[Atwood (1985)] 'An Object-Oriented DBMS for Design Support Applications'. T Atwood . *Proceedings of the IEEE COMPINT* September 1985. 85 p. .

[Delobel and Kanellakis ()] *Building an Object-Oriented Database System: The Story of O 2*, Claude ; Delobel , Paris Kanellakis . 1992. Morgan Kaufmann Publishers.

[Guess (2013)] Angela Guess . *Objectivity Launches Objectivity/DB 11.0". DATAVERSITY. Retrieved*, February 6. 2013. December 2, 2014.

[Kim ()] *Introduction to Object-Oriented Databases*, Won Kim . 1990. The MIT Press.

[Maier et al. (1985)] 'Object-Oriented Database Development at Servio Logic'. D Maier , A Otis , A Purdy . *Database Engineering* December 1985. 18 (4) .

[Derrett et al. (1985)] 'Some Aspects of Operations in an Object-Oriented Database'. N Derrett , W Kent , P Lyngbaek . *Database Engineering* December 1985. IEEE Computer Society. 8 (4) .

[The Committee for Advanced DBMS Function, Third Generation Database System Manifesto Computer Standards and Interfaces 'The Committee for Advanced DBMS Function, Third Generation Database System Manifesto'. *Computer Standards and Interfaces*, 1991. North Holland. 13 p. .

[Edlich et al. ()] *The definitive guide to db4o*, Stefan Edlich , Jim Paterson , Henrik Hörning , Reidar Hörning . 2004. Apress.

[Cattell and Barry (ed.) ()] *The Object Data*, R G G Cattell , Douglas K Barry . Standard: ODMG 3.0. Mark Berler, Jeff Eastman, David Jordan, Craig L. Russell, Olaf Schadow, Torsten Stanienda, and Fernando Velez (ed.) 2000. Morgan Kaufmann Publishers, Inc.

[RG G ()] *The Object Database Standard: ODMG-93, Release 1.1 The Morgan Kaufmann Series in Data Management Systems*, RG G . 1994. (Cattell (ED))

[Atkinson et al. (1989)] 'The Object-Oriented Database System Manifesto'. M Atkinson , F Bancilhon , D Dewitt , K Dittrich , D Maier , S Zdonik . *Proc. of the First International Conference on Deductive and Object-Oriented Databases*, (of the First International Conference on Deductive and Object-Oriented DatabasesKyoto, Japan) December 1989. p. .

[Darwen and Date (1995)] 'The Third Manifesto'. H Darwen , C J Date . *SIGMOD RECORD* March 1995. 24 (1) p. .

[Zemke ()] 'What's new in SQL'. Fred Zemke . *ACM SIGMOD Record* 2011. 2012. 41 (1) p. .