

Why have object oriented databases not been successful (yet)? [closed]

Asked 16 years, 2 months ago Modified 9 years, 5 months ago

Viewed 4k times



21



Closed. This question is [opinion-based](#). It is not currently accepting answers.

💡 **Want to improve this question?** Update the question so it can be answered with facts and citations by [editing this post](#).

Closed 8 years ago.

[Improve this question](#)

That's the question. Give only one reason you think why have OODB failed or why many systems nowadays still use relational databases.

database

orm

rdbms

object-oriented-database

Share Follow

edited Aug 31, 2011 at 11:33



Michael Easter

24.5k ● 9 ● 80 ● 110

asked Oct 4, 2008 at 17:06



lurks

2,606 ● 5 ● 31 ● 39

+1 marxidad Pointers are BAD in data models... OO model is very much like the old network model, which is not only proven to be inferior theoretically to the relational model, but also a pain to use in practice. It's weird how failed data models (hierarchical -> XML, network -> OO) keep getting resurrected. To me, not learning from the past is plain silly, it looks like such a waste of precious human brain cells.

– [Andrew G](#) Oct 5, 2008 at 1:19

2 I think this one is en.wikipedia.org/wiki/Begging_the_question

– [Goran](#) Oct 5, 2008 at 8:28

why pointers are BAD in data models? you should explain better what you mean because just about any O/R mapping tool is using pointers in its generated data structures.

– [lubos hasko](#) Mar 10, 2009 at 0:05

Pointers are a PHYSICAL memory concept, to say that they form a LOGICAL data model is somewhat of a contradiction.

– [Andrew G](#) Mar 17, 2009 at 1:25

You should read something about proofs before coming to such absurd conclusions. – [Stephan Eggermont](#) May 11, 2011 at 17:06

14 Answers

Sorted by:

Highest score (default)



25

The main reason is SQL. It is very useful to be able to use the data from a database in other contexts outside of the application, and often with object databases the data is stored in a format that can't easily be queried. With a relational database the data can become part of a data



warehouse, for instance, or just queried by sys admins etc.



Share Follow



edited Jul 5, 2015 at 9:42



Ajay

18.4k ● 14 ● 63 ● 109

answered Oct 4, 2008 at 17:13



Phil Bennett

4,839 ● 4 ● 31 ● 28

Exactly; SQL can be used from almost any programming environment (or even directly from the query analyzer), while OODBMSes always have some nonstandard, programmatic-only way to query objects in them. – [Jacob Krall](#) Oct 4, 2008 at 20:44

- 1 You can query a DB4O with SQL. There's a small perf hit involved, but it is supported.. – [Boris Callens](#) Jan 14, 2009 at 8:58
-

+1 just for the handle. Pets.com? But I think a major reason is that every OODB thinks they've solved all the world's problems, and charge accordingly. Companies don't want to bet the business on a small, unproven company, which is why Oracle exists. – [chris](#) Mar 28, 2009 at 2:27



23



Can we answer more than once? Another reason is that relational DB's have a strong foundation in mathematics: from the definition of a relation, right through to the normal forms, the theory is rock solid. It is true that the relational model does not map well to OO, but IMHO the



benefits and stability of that model outweigh the mapping problem.



Share Follow

answered Oct 4, 2008 at 17:21



[Michael Easter](#)

24.5k ● 9 ● 80 ● 110

Yes you can. And good second answer as well.

– [Scott Alan Miller](#) Oct 4, 2008 at 17:23

3 Set/Relational theory is king, indeed – [Camilo Díaz Repka](#)
Oct 4, 2008 at 18:46

any OO database could (and many do) provide relational access to its data. – [lubos hasko](#) Mar 10, 2009 at 0:01

Hmm. They might have a foundation there, but they picked the wrong abstractions. The base abstraction should have been a partial ordering (is that english?), not a set.

– [Stephan Eggermont](#) Sep 25, 2009 at 17:56



13



I think it's because "object databases" are solving a problem that (almost)nobody really has. For simple persistence of object graphs, the serialization built into most OO environments is "good enough". If you want to do sophisticated operations on a subset of your data, then a relational database and SQL are a perfect fit.



Other than some fringe applications (enormous object graphs that can't be kept in memory, but for which the relationships don't simplify down well for RDBMS use), there really isn't any need for these tools.



Mark Bessey

19.8k ● 4 ● 49 ● 69

5 "I think it's because "object databases" are solving a problem that (almost)nobody really has." I don't think this is the case: look at the sheer number of ORM projects in use to combat impedance mismatch. – [Phillip B Oldham](#) Nov 16, 2008 at 0:07

1 That's kind of my point. ORM systems handle the match between the OO layer and the database representation well enough that a "pure" object database isn't perceived to add much value. Also, you can relatively easily hire someone to manage an SQL database, but experts in any OODB are much scarcer. – [Mark Bessey](#) Nov 20, 2008 at 20:38

I think that Object Databases are a solution looking for a good problem, and I kind of agree with Mark Bessey.
– [thomasrutter](#) Apr 30, 2009 at 14:27



11



Just because OODB are not the mainstream we should still consider the successes that they have had. Cache and Zope are both widely used (relatively) but would be considered successful by some standards.

Perhaps the biggest reason that OODB have not taken hold dramatically is because of the success of the hybrid object-relational systems that take most of the potential marketshare from OODB: PostgreSQL and Informix.

I know that this does not directly answer the question but it is, I think, part of the equation. Overall, though, I think that momentum and the heavily ingrained thought

processes supporting relation databases make it difficult for people to switch. Currently the DB profession is trained almost exclusively in relational theory making your DB professionals very interested in avoiding OODB and academia teaches DB theory for practitioners almost exclusively on relational.

Until large, corporate DBAs and mainstream professors and curriculum and turning out staff beyond developers prepared to managed OODB I feel that it is unlikely to see mass appeal no matter how good it is from the development side.

Share Follow

answered Oct 4, 2008 at 17:22



Scott Alan Miller

969 ● 6 ● 12



8



Well, it's strange isn't it? There is such a push towards domain driven design as the zenith of object orientated analysis and design, and there are enterprise patterns out there to leverage ORM systems to persist our objects. It just makes total sense to me that if your application DESIGN is object orientated and domain focussed at heart, that an OODB will greatly benefit your application.

Aside from the issues around maturity and uptake, from a philosophical perspective an OODB would appear beneficial or an OO application. not having to maintain that mapping layer for starters;)

But look, if you aren't doing domain drive design and use objects as data objects and like your stored procs, then you're not really going to get it;)

Share Follow

edited Jul 5, 2015 at 9:45



Ajay

18.4k ● 14 ● 63 ● 109

answered Oct 6, 2008 at 4:12



Keith Patton

948 ● 5 ● 12



5



RDBMs are (built on a strong theoretical foundation, have been in the market for a much longer time, can model data more faithfully than OODBs in many cases, can be used by more DBAs than OODBs). That's one reason in the form of a relational tuple.



Share Follow

answered Oct 4, 2008 at 17:45



Mark Cidade

99.8k ● 33 ● 229 ● 237



3



If I can amplify Phil's point: the standardization of SQL. OODB's have tried query languages such as OQL but they never seemed to follow a true standard. Also the quality of the query languages were suspect, arguably due to lack of standardization. Standards foster competition, which spawns quality.



Share Follow

answered Oct 4, 2008 at 17:18





Michael Easter

24.5k ● 9 ● 80 ● 110

2 In my current pet project using db4o I have a line (C# 3.5):
IList<Users> list = Persistence.Database.Query<Users>(u =>
u.Name == "Admin"); Using strong typed lambda expression
to get a (lazy) list of objects from the database. I'd say that
beats SQL as a standard. – [Goran](#) Oct 6, 2008 at 22:34

how many 3rd party tools support that line? Can a sys-admin
use that line at 11 pm at night? To me standardization allows
the creation of tools both for developers and sys admin staff.
– [Michael Easter](#) Oct 7, 2008 at 22:05

db4o supports several forms of query languages. As a
programmer I strongly favor the Linq variant off course.
– [Boris Callens](#) Jan 14, 2009 at 9:02

Which standard? SQL 92 is not even implemented, as soon
as you have a date in your database you have a problem
– [Stephan Eggermont](#) Sep 25, 2009 at 17:59



2

That, and o/r-mappers. Through them, the difference to
true OO-DBs becomes way smaller, while the
aforementioned benefits stay valid.



Share Follow

answered Oct 4, 2008 at 17:18



Urs Reupke

6,901 ● 3 ● 38 ● 49



One reason is that databases are about data, and objects
are about structures and algorithms. Once you take the

2

data and embed it in classes, you characterize the relationships and operations in a static structure.



Databases, on the other hand, are about unstructuring the data into a bunch of instances of atomic tables that can be reassembled into different structures (usually with classes) without disturbing the integrity of the atoms.



Databases are somewhat analogous to hexahexaflexagons.

Share Follow

answered Mar 28, 2009 at 2:50



dkretz

37.6k ● 13 ● 83 ● 140

Except that that doesn't work. Relational databases are the source of all data problems: up to 30% of all records in relational databases contain mistakes – [Stephan Eggermont](#) May 10, 2011 at 23:18



2

Expensive technology decisions are not made by people with technical knowledge. Companies using relational databases employ lots of people who feel threatened by OODBs and therefore will avoid learning about them.



Share Follow

answered May 10, 2011 at 23:22



Stephan Eggermont

15.9k ● 1 ● 40 ● 65



I think that there are two philosophical reasons.

1



First, people traditionally tend to separate persistence from real functionality. Once you strip away an object's "life" away from it and keep it primarily for persistence, it becomes a record, and then there is a tendency to treat it as a "lifeless" data object.

Following on that, when people think of a large collection of very similar things, they start thinking of them as tables rather than objects.

I think with O/R the distinction is starting to disappear. For example, I use hibernate to dump really complex class hierarchies into a MySQL database. However, I don't write performance-critical stuff for my project so I'm sure it's not done efficiently.

Share Follow

answered Oct 4, 2008 at 19:04



Uri

89.6k ● 51 ● 226 ● 322



1



The reason for the slow adoption of OODB's are based largely on a few key factors that make the relational SQL databases more popular and/or more appropriate. While pure object-oriented databases are now in a state where they overcome much of the drawbacks of the relational model, there are some key pieces missing.

For one they tend to lack support for central database management, though this is rapidly being rectified in various products.

A second reason is that very few systems implement a standard query language and instead relies on the programming language or specialized query languages to retrieve and manipulate the data in store. This is a show stopper for many if they have to learn a new query language on top of the totally different mindset of a programmer used to NoSQL based solutions. On top of that, most SQL based / Relational databases now have some support for Object Oriented Design, plus we have wrappers like ORM that many use to "bypass" the problems of relational databases not being readily available in the programming language of choice.

But these problems exist mostly in corporate environments. As embedded databases in small devices, as web site storage and in fields like aerospace they have become very popular and in many cases totally replaced the need for regular relational databases.

Who knows what the future holds?

Share Follow

answered Oct 30, 2013 at 22:16



[Sederqvist](#)

2,951 ● 1 ● 20 ● 28



0



Serialization provided by the most of Language lets you flatten the Object attributes and thus storing easily them into the RDBMS and similarly retrieving objects is not a big issue. The wide and solid foundation still lacks which hinders the use of OODBMS to be implemented.



I currently thinking of doing this as my Master Thesis project to provide a general framework for OODBMS that supports almost all the components which is commonly used in now a day RDBMS thus providing a non-linear structured DBMS. While studying I came across a project called db4o which is an approach (implemented) of using OODBMS for Java and .net only, so this could be another reason of lack of generality for all types of platforms and languages.

Share Follow

answered Feb 18, 2014 at 19:21



INDERJEET SINGH

13 ● 3



-1



I think that's because big guys like Oracle had been investing in relational databases while object oriented movement was getting momentum...may be they will become mainstream if Oracle / Microsoft invest in it in a big way...which seems unlikely because they don't have a strong reason to do so...it will simplify lives of many programmers...but "making programmers lives simpler" is not a very good business goal for them!

Share Follow

answered Jun 15, 2011 at 11:39



Sumedh

435 ● 6 ● 14

Well, Oracle have their own NoSQL database as we speak...

– [Sederqvist](#) Oct 30, 2013 at 22:20

