

MS Access as Enterprise Software?

Asked 15 years, 11 months ago Modified 12 years, 4 months ago

Viewed 10k times



21



Something that I often run into with my users is their desire to acquire solutions quickly means that they sometimes have said "Heck, I'll just roll up my sleeves and do it in Access - it's installed on my desktop".

Sometimes, we're lucky and the person that creates the Access database back-ends it to a SQL Server, so at least the mdb file issues that often come up aren't an issue.

However, it is my opinion that rolling out an Access front-end to a SQL Server database as an enterprise solution with thousands of users, and hundreds of thousands of rows is still problematic.

What are your opinions on this? What are some of the potential pitfalls?

OR

Is this a perfectly acceptable, stable, maintainable, and robust solution?

ms-access

Share

Improve this question

Follow

edited Jul 27, 2012 at 8:00



ThinkingStiff

65.3k ● 30 ● 147 ● 241

asked Jan 22, 2009 at 15:00



Sam Schutte

6,736 ● 7 ● 45 ● 55

-
- 2 How have people's thoughts on this changed in the last 10 years? – [Justin Cervero](#) Jan 14, 2019 at 15:40
-
- 1 I would bet and hope there's less Access development going on out there as opposed to ten years ago...but the same isn't true with Excel sheets being used for business process, so maybe not. With the rise of no- and low-code solutions, Access seems like an even sketchier bet. More likely, there are still some very large systems out there that were active ten years ago, and still active now, because no one has ever gotten rid of them, and they've just grown and grown (I know of several...). – [Sam Schutte](#) Jan 15, 2019 at 17:42
-

#Justin Cervero, 15 years later, I would say to stay away from Access at all cost. My app (using an Access backend) worked perfectly fine up until we tried migrating to windows 10. Then we started encountering several memory leaks coming from access and also "Out of memory" exceptions coming from access on a very small databases! Microsoft seems clueless as to how to fix those errors. – [Maxter](#) Oct 9 at 14:33 ✎

11 Answers

Sorted by:

Highest score (default)



I've worked with this scenario a great deal. In fact as a consultant/developer Access front end SQL Server back

23



end has been a significant part of my bread and butter work over the past 10 years. Which doesn't mean I *like* Access ;-)



Up until the common adoption of AJAX it was a perfectly reasonable solution. And there's still vast numbers of small to medium sized applications put together in Access out there that run bespoke business systems perfectly happily and I doubt it's going to go away for the next 10 or more years - indeed Access/SQL is probably going to be the Cobol of the 21st century. If you're working on a 'green field' site then there is now virtually no excuse for deploying Access when building from scratch - but if you do inherit an existing application then the costs of a rewrite may not be worthwhile and difficult to pass with the users.

Access does have some advantages that are still significant - and can present problems if proposing to convert to a web app

1. It's quick. For simple CRUD work it's as fast to write and deploy as any other realistic solution.
2. Built-in reporting is easy to get running and remarkably powerful given the system. It's usually pretty easy to create and deploy new reports for users on demand.
3. It integrates well with Office. This one tends to be the show-stopper when looking to move Access apps to web-apps. It's extremely common for a 'department-size' Access application to **tightly** integrate with

Outlook, Word or Excel - and often all three. This is **the** major problem when dealing with real-world situations. It's very easy for coders to underestimate the importance of this for everyday usage of such systems and the imposition of even a small degree of additional hassle for the users will generally be met with much resistance - often enough to completely scupper the project.

4. If your working with a reasonable sized department - a dozen people or so - it's quite common for there to be someone in the office who fancies themselves as a bit of a computer wizard. These people can be a major pain if handled incorrectly, but equally can be a major asset. If I have such a person I will try to get management to send them on an Access course or two so they can write simple queries and reports, and set up a separate Access application for them *which they own* which has appropriate (restricted) access to the SQL database. You can then trust this person to handle producing simple reports and the like for their colleagues. This can be a real win-win - you gain someone who is on your side and will use you as a mentor - a ready-made advocate for you in the department - and they keep the grunt report work out of your hair. They gain a lot kudos and job satisfaction - and even a potential career path. It's far harder, well near impossible, to do this kind of thing with any other system but Access.

Main practical disadvantages

1. Deployment can be a nightmare. Generally if you have a very tightly defined environment - a small company, single department, citrix based or distributed with an IT department that closely controls it's PCs then you're fine. Deployment as a commercial app across multiple companies - well only if you can charge significant maintenance (been there).
2. Code does not scale. Access VBA code, even when written by a professional has a strong tendency to rot into rancid spaghetti. It's quite common to end up with an Access application that was easy enough to maintain, but gradually becomes unmaintainable as dependencies multiply.

So I'd say Access still has a place, and it's use is defensible in many real world situations, but increasingly it's better to choose a more modern solution if circumstances permit.

Share Improve this answer

edited Jan 22, 2009 at 17:40

Follow

answered Jan 22, 2009 at 17:32



Cruachan

16k ● 6 ● 62 ● 112

-
- 2 I've said for years that Access' reporting tools are top notch -- quick, easy, and powerful. – [John Mo](#) Jan 22, 2009 at 18:18
 - 1 I agree Access' reporting tools are top notch... which is why there are so many excellent copies of those tools in far better

products than Access. – [Sailing Judo](#) Jan 22, 2009 at 19:17

- 1 I've found a better solution to be opening up Report Builder in SQL Server Reporting Services to that "tech savvy" person. They can build their own web reports then, with a little training. – [Sam Schutte](#) Jan 22, 2009 at 19:49
-

I've not been in a position to try the SQL Reporting Services approach yet, but certainly it's an option I'm keen to try. I really am *not* a fan of Access, I simply find I am forced to use it by circumstances more often than I'd like. – [Cruachan](#) Jan 22, 2009 at 20:18

Deployment issues can be solved, making it as easy (even easier) than any other app. Please check my post.
– [Philippe Grondier](#) Jan 23, 2009 at 3:29



13



We have built such a solution (Access front-end, SQL back end), with now something like 80 users, millions of rows replicated between different countries, more than 100 000 updates a month. It works fine. I think the main mistake about Access is to consider it as a tool made for amateurs to develop applications. It can work this way, but keep in mind that amateur development will give you amateur applications, while professional development will give you professional results.

A quick list of its advantages, problems and limits:

- It's free for the final user, thanks to msAccess runtime
- It works with the free SQLServer Express, and not so expensive SQL Server Enterprise.

- It's quick, specially when dealing with forms
- It communicates very easily with other Office apps, which are still enterprise standards
- You can manage its interface to be so close to Office standards that using it can be very intuitive, making people happy (I talked a little bit about that on [my blog](#), need to be updated!)
- On a large scale, you have to think about the best way to distribute it to your users. This issue can turn into a nightmare, as noted by #Cruachan, but it can be solved by building and distributing msi packs for example. Such msi packs can also contain all your external references such as 'added' dll, ocx, tlb files (report dll, activeX scanner controls, etc). We had a few words on this [here](#).
- When distributing an updated version of the mdb file, you can have a common network folder holding the new mdb/zipped file that clients will check/update at startup. Your clients should have the possibility to reinstall a previous version of the mdb file. Upgrading becomes then easier than installing a new .exe file.
- You have to set a version controlling system. Please check [here](#) for details.
- You must be very strict on your code organisation. One of our basic rules is for example not to have any specific code at the form level. Please check [here](#) on this subject.

- I didn't find any problem with VBA code scaling, as noted by #Cruachan. If professional coding rules are implemented, there will not be any unusual code scaling issue. As an example, our application is now working really fine with more than 180 different forms, and still growing without any problem.

As a conclusion, our main problem with Access is an image problem, where Microsoft still let people think that Access is here to give them the possibility to develop real software in 10 lessons ... and professionals, who know that is not possible, view it as a amateur tool for amateur development, looking down on ms-access users as boring low IQ red-necks.

Share Improve this answer

Follow

edited May 23, 2017 at 12:00



Community Bot

1 • 1

answered Jan 23, 2009 at 3:21



Philippe Grondier

11.1k • 3 • 36 • 73

-
- 2 The problem is that all the positive points you mentioned are also positive points in most other platforms (ie., Access gives you nothing). Just because you *can* do something in Access doesn't mean you *should*. – [Sailing Judo](#) Feb 11, 2009 at 16:39
 - 3 ... meaning Access can be considered as equivalent to other platforms (let's say 'other rich-client platforms'), an assertion that most people might not accept! – [Philippe Grondier](#) Feb 11, 2009 at 20:32
-



10



I know quite a few professional Access developers who have developed and maintained Enterprise-level apps using Access as the front end (either MDB or ADP) and supporting user populations in the 100s (and even in a few cases, thousands).

Like any Enterprise-level application development, it requires a higher level of programming skill than building a little Access database for your 5-person department.

Oddly enough, the design principles that make for an efficient Enterprise-level app also make for a more efficient workgroup-level Access app.

I think the reason most of the people posting in this thread can't conceive of it as a good solution is simply because they've never seen it done properly, or were themselves not sympathetic to the development model that Access uses.

Yes, it's hard to do properly.

But at that level, so is every other development platform - all of them require planning, experience and a high-level skillset.

And you can rag on Access apps developed by people without all of that (Enterprise or not), but frankly, I've encountered a boatload of non-Access database apps of all kinds that are incredibly badly implemented.

Sturgeon's law applies everywhere, and there's no reason to assume that Access development would be any different.

Share Improve this answer

edited Jun 23, 2009 at 22:04

Follow



Marc Gravell

1.1m ● 273 ● 2.6k ● 3k

answered Jan 23, 2009 at 3:16



David-W-Fenton

23.1k ● 4 ● 47 ● 58

Good point. I think that Access makes it harder than a lot of other platforms that are more popular for that type of use, but I understand what you're saying. Just anyone can't fire up Visual Studio and pump it out to a web server, simply because of permissions - but in access they can.

– [Sam Schutte](#) Jan 23, 2009 at 21:23

-
- 1 This is a *good* thing, not a bad thing. The reason there's so much horrid HTML out there is because HTML is so easy. We wouldn't throw out HTML because some people are bad at it.

– [David-W-Fenton](#) Jan 25, 2009 at 1:00

Everything David Fenton says makes sense, and he would get an extra vote from me, if for nothing else than mentioning Sturgeon's Law. – [Knox](#) May 21, 2009 at 18:00



6



I started out doing desktop applications in Access with JET back-ends. I moved up to using SQL Server/MSDE with Access as the front-end and then VB6 and a smattering of classic ASP.



There are many "enterprisey" reasons to go with a "real" development tool like Visual Studio. For the scale you are inquiring about, thousands of users, I think those reasons may apply.

That said, I think there are scenarios where it still works to use Access. In my own experience, I fell back to Access with a SQL database when given a mandate to come up with an enterprise solution, albeit for a much smaller enterprise, in a very short period of time. The main reason driving my decision was time. I can put together a database UI in Access much, much faster than I can in any other tool. Some of that is familiarity with the tool, but a lot of it is that Access just gives you more database purpose-specific bits to work with out of the box. The Access UI can also be tweaked to look and operate very much like a standard WinForms app.

The hitch that many run into in an enterprise scenario is rolling Access and the application MDB/MDE out to the masses. This is easily resolved by setting it up on a Windows Terminal Server, which can also be rigged to operate almost like another app window on the client machine with the right RDP file parameters. But even that approach has its limits. I don't think it would scale into the thousands very well, but for several dozen users, I found that it worked just fine and bought enough time to meet the time constraints I had to work with so a web interface could be implemented when time allowed.

For a professional who knows what they're doing in a SQL database, an Access front end is not necessarily an unpardonable sin, especially when the mandate is cheap and fast and there isn't a religious purism involved.

Share Improve this answer

answered Jan 22, 2009 at 16:04

Follow



John Mo

1,326 ● 10 ● 14

Interesting that you bring up Terminal Server - one of the Access Apps I'm currently exposed to uses Citrix to make the MDB look like a local app. – [Sam Schutte](#) Jan 22, 2009 at 19:45



4



If you have the choice, no.

That being said, there are situations where it may be alright. One situation is if you never ever plan on updating the access application. If it is installed for thousands of users, you may run into problems getting all of the client apps updated.



You are much better off making a web front end...



Although Access makes the multiple master-detail forms easier than anything I have seen. Even Oracle Application Express, intended to compete with Access, cannot do everything that Access does.

My advice is that if you are a programmer, you can make an asp.net app that will do the same job in a much more scaleable, maintainable, nature.

Share Improve this answer

answered Jan 22, 2009 at 15:12

Follow



jle

9,459 ● 5 ● 50 ● 68



4



For a lot of CRUD (Create Read Update Delete) work, MS Access is OK. I'm more confident in it if the data is in another engine (MSSQL/Oracle/MySQL). However, most of the time I have problems with an MS Access database it's because:

1. It was home grown by a desktop user (not a programmer/IT professional) and hasn't planned ahead for future development (so additions are often more painful than if a pro had been involved)
2. It's full of unnormalized tables, inconsistencies, and key-less tables.

My solution. Limit MS Access to the pro's and deploy the runtime version to the users desktops.

Share Improve this answer

answered Jan 22, 2009 at 15:18

Follow



BIBD

15.4k ● 26 ● 87 ● 141



4



For the multiple user/high data volume situation, I use Access front-ends with a MySQL back end. I must say that in the client-server situation especially on a LAN, MS Access is as good as they come. Personally, I find development in MS Access much faster than say Visual Studio especially when it comes to database driven apps. And Access reports are as good, if not better than the



industry standard, Crystal Reports. The only shortcoming I see with Access is when it comes to non-LAN situations where you have to distribute the application to users spanning a wide geographical area. But again, web apps themselves have a major shortcoming - handling of one-to-many relationships, something access superbly handles with its sub-form, sub-report feature. And more importantly, Access has a very powerful event model that most applications cannot match. Personally, I can literally do anything on Access! So, my conclusion is that MS Access has many advantages that make it a competent development tool especially in LAN environments.

Share Improve this answer

answered Nov 25, 2010 at 8:31

Follow



Peter Ssetumba
Mugerwa

41 ● 1



3



Sadly, I have quite a bit of experience with this. We built an entire product around Access Forms tying into SQL database. Honestly, the performance wasn't an issue - it really is the normal db connection type scenarios that you'd have to be concerned about with any client/server app. In our case, the original developer knew tons of "tricks" in Access, and did things like databinding drop downs to stored procedures. Oh, and the awful triggers. Awful. As in, 45 triggers firing per update awful.

The tables we worked with did indeed have millions of rows of data, however typically the roll-out was to tens or hundreds of users. I'd imagine that any effort going out to

thousands of users would benefit more from a custom development so that you can do things like build the software correctly, support it from a performance and development perspective, and build automated deployment options (MSIs or ClickOnce, for example).

So, I would not say it is a perfectly acceptable, stable, maintainable or robust solution. It worked for us because we were there to support it (and eventually rewrite it in .NET), but I wouldn't recommend it for anyone. I have, however, worked in government where trying to get anything done from "IT" (which I was part of) was so filled with red-tape and paperwork that departments would oftentimes just do the Access solutions.

Ultimately if that's the case you are in - where the departments simply can't get access to IT resources - then showing them at least some best practices for how to eventually scale the app would be helpful. As long as right after you show them, you put your resume out to find a better job.

[Share](#) [Improve this answer](#)

[Follow](#)

answered Jan 22, 2009 at 15:13



[Cory Foy](#)

7,212 ● 4 ● 32 ● 34



2

12-15 years ago this might have been an acceptable practice (not really advisable, but acceptable) but nowadays its unforgivable. There are so many more scalable and distributable solutions that Access should be the last thing to cross somebody's mind.



When you say Access as a solution what comes to my mind is a simple, 2-3 table application that some marketing employee put together, not a real developer. If the marketing guy had a really good idea then perhaps the development team should look at it (I'm assuming there is one sense you indicated there may be thousands of users), refactor it to a better platform (intranet or winforms distributed via ClickOnce, etc), and then deploy it.

Back in the early 90s I was an Access developer--even had a MS certification. I built dozens of "Enterprise" apps (meaning 10-15 people used them). Those days are gone, IMO. There are easier solutions to build, deploy, and maintain nowadays.

Share Improve this answer

Follow

answered Jan 22, 2009 at 15:13



[Sailing Judo](#)

11.2k ● 20 ● 71 ● 100

"There are easier solutions to build, deploy, and maintain nowadays" What are these solutions? – [Mikhail Poda](#) Jan 9, 2010 at 0:25

Off the top of my head: SharePoint. This has come a long ways in the last few years. But any web-based solution will be far easier to deploy and maintain for a corporate app.
– [Sailing Judo](#) Jan 9, 2010 at 22:55



I've had the misfortune to work on Access front ends like you describe, here are some non-Enterprise arguments.

1



Programming is easy! Creating forms in access is geared toward non-developers. Case in point, if you have multiple columns in a drop down, do you have list fields and data fields. No way! you just set the width of things you don't want to see to 0". So your looking at forms either thrown together by non-developers, or that will irk most people that have to work on them.

Versioning? Who needs versioning?, Just send out an attachment If changes need to be made to the front-end re-deployment is time consuming and fault prone.

This form, I'm thinking magenta The front end doesn't lock down well so end-users can get creative.

Share Improve this answer

answered Jan 22, 2009 at 15:14

Follow



cmsjr

59k ● 11 ● 74 ● 63

-
- 1 Programming is easy! Better Access than Excel. Version updates? Use the free auto FE Updater at autofeupdater.com FEs can and should be locked down by distributing MDE/ACCDEs. – [Tony Toews](#) Oct 9, 2009 at 21:35
-



0



With Microsoft "giving away" free versions (MSDE, or SQL Express for 2005 onwards) of the SQL Server engine with each release, there is really no need to use Access any more. Although these free versions don't have a visual front end which can make development harder, good knowledge of SQL is all you need.



Share Improve this answer

answered Jan 22, 2009 at 15:17



Follow



J__

3,827 ● 1 ● 26 ● 33

Not really referring to using Access as a database - just as a front end. So really more than just good knowledge of SQL is needed (Access expertise, C# expertise, or whatever).

– [Sam Schutte](#) Jan 22, 2009 at 19:41

This comment completely reflects the ignorance of so many people who simply don't understand Access. Even with a very clearly stated question regarding only the issue of front-end development, yahoos still come in and say "use SQL Server!" – [David-W-Fenton](#) Jan 23, 2009 at 3:10

Yahoos? I developed on Access for nearly 10 years before I made the switch to SQL Server. – [J__](#) Jan 23, 2009 at 8:42

You couldn't possibly have switched from developing in Access to developing in SQL Server, as SQL Server has no UI development tools. So, yes, YAHOO. – [David-W-Fenton](#) Jan 25, 2009 at 1:05

Excuse me? Enterprise Manager/SQL Server Management Studio. And then we wrote our own as well. You know nothing about me and then purport to tell me how I've run my career? For a former music teacher (you see, I did my research), I think you need to learn a little more before you spout off. – [J__](#) Jan 26, 2009 at 16:05
