

Re: DAO vs ADO

Source: <http://www.tech-archive.net/Archive/Access/microsoft.public.access.conversion/2004-08/0214.html>

From: Craig Alexander Morrison (*reply_at_newsgroups.com*)

Date: 08/22/04

Date: Sun, 22 Aug 2004 19:01:52 +0100

Mary

- > *If you're interested, these days you'll get the livliest discussions*
- > *on relational design in the microsoft.public.sqlserver.programming*
- > *newsgroup, where Joe Celko livens up many a thread.*

In the past I have had my differences with Joe as he comes from the SQL side rather than the Relational side. Some may not be aware that SQL is not (wholly) relational. One shudders to think what CLR could do it could even allow Joe and I to agree on something. (vbg) On the 10 year old thread I mentioned earlier Joe and I came to blows on the problems associated with not defining a Primary Key. Our disagreement was based on him coming from the SQL side and me from the Relational side, we were both right in our own terms. I think Joe now agrees more with the relational side. Having had a look at his recent contributions on the above newsgroup I find little to disagree with. Mind you, give me time. (vbg)

Regarding little ol' Access, whilst the architecture may be 10 years old it is (roughly) based on Relational Theory which is over 30 years old and is still relevant.

- > *Your Access FE should work fine with SQLS 2005 as long as you're*
- > *running in 2000 compatibility mode.*

What's the problem with 2002/2003 in this respect?

- > *It's unclear at this point if Access will ever have full support for new*
- > *features like CLR assemblies and UDTs. There's only so much you*
- > *can tack on to what is essentially an ancient architecture.*

However a tool like Access is required to allow users and database developers to be able to harness the power of the new database engines. We certainly do not want to have to rely on programmers with little understanding of set theory and their love of record numbers and "so-called" surrogates.

Note: A true surrogate (from say a hashing algorithm) is hidden to the user and that includes the developer, it is system generated and internal. It

allows the database developer to pick the real natural Primary Key even if that is fairly complex and the system hashes this to some internal value that is then used to refer to this value in all relationships.

> *How much has really changed in the last 10 years?*

Very little on the theory side. If anything the theory is becoming better implemented, I may have to leave SQL Server 2005 out of that generalisation though. I am no