

Re: About MS–Access RDBMS

Source:

<http://www.tech–archive.net/Archive/Access/microsoft.public.access.tablesdbdesign/2005–04/msg00410.html>

- *From:* Tim Ferguson <FergusonTG@xxxxxxxxxxxxx>
 - *Date:* Mon, 11 Apr 2005 09:30:39 –0700
-

"=?Utf-8?B?U3Jpbml2YXNhbG==?=" <Srinivasan@xxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
wrote in news:86190119-FA54-4665-AD9C-BFC659ABE540@xxxxxxxxxxxxx:

> Can we say MS–Access as True RDBMS?,

I have a feeling that this is a troll, but in any case I'm sucker enough to bite... Codd describe the twelve rules of relational fidelity, and Pascal added rule 0.

0) Foundation rule: a relational dbms must manage databases entirely through relational capabilities.

The mdb file is a monolith, and it's not possible to pretend that tables in different file can be related, and there are no tools to manipulate the data in mdb files directly; so this is a pass.

1) Information rule: all information must be held in R–tables.

Access allows tables to be created without primary keys and therefore fails rule 1.

2) Guaranteed logical access rule: every value must be accessible via table name, PK value and column name.

Since Access fails rule 1, this also fails by definition.

7) Set level operation rule: select, insert, update and delete operations must be available via set operations rather than requiring row–by–row manipulation

Within the limitations of the language, all these operations are called by SQL and so this is a pass.

3) Missing information rule: missing information should be handled by the database not by applications' own exception processing features.

Access provides NULL handling on all data types except booleans. Note

Re: About MS–Access RDBMS

that Codd himself had big problems with proper handling of null data, so Access is not significantly worse than any other product. I think MVDBMSs are meant to be better, but I don't understand enough to make a valid judgement.

8) Physical independence rule: applications and interactive operations should remain independent of internal storage mechanisms.

Microsoft only provides access to (logical) data via the Jet engine, and the internal (physical) structure of mdb files is a deep mystery which may or may not change from one version to another. This rule is therefore a pass.

9) Logical independence rule: applications and interactive operations should remain independent of the structure of the base tables.

Access does not support views as an alternative to direct base table access — queries are partly updateable and cannot always be substituted for base tables. This is at best a partial pass, and strictly it's a fail.

6) View update rule: the rdbms should identify at definition time whether or to what extent a view can be updateable.

I have a feeling that Access should do this, but updateability seems so unreliable and paradoxical that I can't use the capability. In my hands, Access fails this rule, but it may be that in others' more experienced, it's at least a partial pass.

10) Integrity independence rule: operations and applications should be independent of internal integrity declarations.

Access can define unique key and foreign key integrity rules, validation rules that cover one table and rules across multiple tables. Unfortunately, the user interface makes these hard and often pretends that they are optional, so although this rule is a pass, Microsoft loses credit for trying to obfuscate it.

11) Distribution independence rule: operations and applications should be independent of distribution and re–distribution of the database

I don't have a lot of experience with Jet replication, but I gather that the applications run just as happily against a replica as with a real database, so this is a pass. Just whether replication counts as true distribution is another argument and it's well over my head!

4) Dynamic online relational catalog rule: The database description should be held just like ordinary data i.e in R–tables

Access nearly gets a pass on this. Actually, the entire schema is held in R–tables, but then again Microsoft has officially not documented these and

Re: About MS–Access RDBMS

the only reliable access is via proprietary dao or adox libraries. In practical terms, therefore, it's a fail.

5) Comprehensive data language rule: there should be one language to support all the relational features of the database – data definition, data manipulation, integrity rules, authorisation etc.

Access has a pretty complete implementation of SQL DDL and DML; but it does not use the same language for security features or the system catalog or queries, so this is a fail.

12) Non subversion rule: operations in a native language should not be able to subvert data integrity rules declared in a higher–level language.

Access does not allow the DAO library (for example) to break integrity rules, so this is a pass.

Summary: Access passes seven out of thirteen rules for relational fidelity. It's up to you whether this makes it a True RDBMS, or even whether you think that a product can be half–relational. Are you looking for a platform to learn about relational theory and design? Are you looking for a standardised environment to develop a mission–critical database? Are you comparing with other desktop products, or with industry–strength software systems costing thousands of dollars?

All the best

Tim F

• **Follow–Ups:**

- ◆ **Re: About MS–Access RDBMS**
◇ From: david epsom dot com dot au
- ◆ **Re: About MS–Access RDBMS**
◇ From: Jamie Collins
- ◆ **Re: About MS–Access RDBMS**
◇ From: Srinivas

• **References:**

- ◆ **About MS–Access RDBMS**
◇ From: Srinivasan
- Prev by Date: **Re: Normal Form–Second(Mr. Vinson?)**
- Next by Date: **garage repair sheet**
- Previous by thread: **Re: About MS–Access RDBMS**
- Next by thread: **Re: About MS–Access RDBMS**
- Index(es):

Re: About MS–Access RDBMS

- ◆ *Date*
- ◆ *Thread*