DATABASEDEBUNK

Schema, NoSQL & Database Management

Fabian Pascal
www.dbdebunk.com

"SCHEMA-LESS DBMS"?

... Most any NoSQL store is schema-less. And while perhaps schema-less-ness is an integral part of NoSQL … it's an orthogonal concern … document-oriented databases, e.g. MongoDB, could arguably have a schema … [But] MongoDB actually touts its lack of a schema as a benefit, claiming it is "agile" and offers "simplicity and power".

--S. Haberman, *What's Wrong with the Schema?*

DEBUNKINGS

DATABASEDEBUNK

NoSQL

- Not *just* NoSQL;
- Not even just *anti*-SQL;

- No SQL, but no NoSQL either (C. J. Date)

"DYNAMIC SCHEMA"

MongoDB (from "humongous") is a *scalable*, *high-performance*, open source NoSQL database. Written in C++, MongoDB features *document-oriented storage*: JSON-style documents with *dynamic schemas* offer simplicity and power.

--MongoDB.com

DATABASEDEBUNK

WRONG W. THIS PIC?

In relational data models [sic], conceptually there is a "correct" design for a given entity-relationship model independent of the use case. This is typically a third normal form normalization. One typically only diverges from this for performance reasons.

In MongoDB, the schema design is not only a function of the data to be modeled, but also of the use case. The schema design is optimized for our most common use case. This has pros and cons – that use case is then typically highly performant; however there is a *bias* in the schema which may make certain ad hoc queries a little less elegant than in the relational schema.

--MongoDB.com

"FIXED" R SCHEMA?

Codd taught the computing world that databases should have *fixed* logical schemas (which protect the user from having to know about physical database organization) … proposed 12 rules for a relational DBMS, the three most fundamental of which are:

Ø **Foundation Rule**: A relational database management system must manage its stored data using only its relational capabilities.

Ø **Information Rule**: All information in the database should be represented in one and only one way — as values in a table.

Ø **Guaranteed Access Rule**: Each and every datum (atomic value) is guaranteed to be logically accessible by resorting to a combination of table name, primary key value and column name.

… The clear implication was that programmers could or should be able to write anything they wanted to against that schema, without database performance being unduly compromised.

–Curt Monash

DATABASEDEBUNK

DATA MODEL

Database management requires *some* **data model.**

- Structure
- Manipulation
- Integrity

--E. F. Codd

FORMAL & WELL DEFINED

- Structure → Manipulation & integrity
- **R-tables on domains**
- **Predicate logic & set theory**

- Real world interpretation

DEBUNKINGS

Domains are things we can talk about, [R-tables] represent sets of statements that we can utter about those things.

–-C. J. Date

DATABASEDEBUNK

"LOOSELY DEFINED"

Document-oriented stores … *encapsulate data into loosely defined documents*, rather than tables with columns and rows … represent a document as XML [or] as JSON, for instance.

--Haberman

- What is the *structure* of "documents that loosely encapsulate data"?
- What *operations* and *constraints* is it amenable to?
- On what *theoretical foundation* do they rest?

Precisely, please!

XML "QUERY ALGEBRA"

Now, my eyes light up at the word "algebra" ...

Set of operations that are closed over some type: every operation in X algebra operates on zero or more values of type X and returns a value of type X e.g. set algebra, Boolean algebra, relational algebra, number algebra (arithmetic).

Over *what* is the XML Query Algebra closed? Nobody has ever given me an answer that makes sense (apart from the occasional, honest "I don't know")

--H. Darwen

DATABASEDEBUNK

XQUERY (cont'd)

I can guess that every [W3C] XQuery expression perhaps operates on one (zero?) or more sequences of zero or more *things*, each of which is

- an atomic value
- an element
- an attribute
- a document
- a text comment
- a processing-instruction node

yielding one such sequence. I am already struck by the complexity, without delving into what any of these things might be.

–H. Darwen

CODD'S GENIUS

FP: It is rather revealing that the very raison d'etre of XML idea—the document—had to be discarded in favor of the "sequence" abstraction, which says everything you need to know about the whole endeavor.

DEBUNKINGS

That *ad-hoc complexity* is precisely what Codd intended to avoid and was smart enough to succeed.

HD: Absolutely!

DATABASEDEBUNK

TEXT: STRUCTURE?

I can envision how you could (simplistically) "model" a document (paragraphs *made up of sentences made up of words*), but other than that what would you do with it? It's not like you could really apply any rules (aside from grammatical, perhaps) which made much sense … metadata, like *author, subject, etc. might be useful*, but that's neither here nor there.

--Matt Rogish

DOCUMENT DATA MODEL?

- Structure (leases, contracts, reports …)
- Operations?
- Integrity?

DEBUNKINGS

- Document-specific expertise
- DBMS Integration/Maintenance/Support

Remember blades?

DATABASEDEBUNK

THOSE WHO FORGET THE PAST...

The NoSQL movement has spawned a slew of alternative data stores, all of which attempt to fill voids left by traditional relational database implementations. But while it's easy to fit the various relational databases (MySQL, Oracle, DB2, and so on) under a single categorical umbrella, the NoSQL world is much more diverse, and the NoSQL label is too general. NoSQL data stores such as MongoDB and Cassandra are so vastly different from each other, that apples-to-apples comparisons are practically impossible. Thus, within the world of NoSQL, there are subcategories such as key-value stores, graph databases, and document-oriented stores.

–Online report

REALITY CHECK

Ø Relational schema

- Database vs. app. functions
- App-neutral db/app specific views
- Data independence

Ø Not NoSQL, NoR!

DEBUNKINGS

Ø Data model?

- The NoDesign illusion
- Manipulation and integrity?
- App-specific DBMS/databases?

DATABASEDEBUNK

Ø One of the major complaints, and sources of home-grown scripts and ad hoc procedures is the inability to automatically synchronize database schema with code release. I've seen countless PL/SQL script packages that migrate tables to new versions, and they all differ (and typically get some things wrong).

Ø … tooling to assist the developer in keeping different software branches and releases in sync with the database schema and managing the schema through source code control e.g. CVS, Perforce, etc.

--E. Kaun

DON'T TRADE DOWN!

Any data management technology claimed to be an improvement over the relational model

- *Must* be based on a data model that:
  - Has a *formal* foundation as sound as
    - Predicate logic
    - Set theory

  and
  - Has a real world interpretation

  and
  - Is as complete
    - Structure
    - Integrity
    - Manipulation

  and
  - Is more general and/or simpler

DEBU

DATABASEDEBUNK

DBDebunk.com

DATABASE DEBUNKINGS

To correct misconceptions about and educate on the practical implications of data fundamentals--concepts, principles and methods--that receive little, incorrect, or no coverage. And to make foundation knowledge accessible to the thinking database professional and user, regardless of the DBMS used.

- To correct myths and misconceptions about and explain the practical implications of **data fundamentals**--concepts, principles and methods--that receive little, incorrect, or no coverage in the industry, in language accessible to data professionals and users;
- For anybody who interacts with data and databases, prefers to think for herself/himself, *understanding* to a "cookbook approach" and soundness to fads and fashion;
- Focus on *education*—as distinct from *tool-specific training*—useful for any and all DBMS products used.

NO COMMENT

DEBUNKINGS

I've read a few things about NoSQL (technology or movement? That is the question!). As my colleague Stuart McLachlan rephrased the question: If it's a movement, the real question is whether it's a religious or bowel movement.

--*SQL/PostSQL/NoSQL*, artfulopinions.blogspot.com

DATABASEDEBUNK

SEMINARS & PAPERS

DATABASE DEBUNKINGS

To correct misconceptions about and educate on the practical implications of data fundamentals--concepts, principles and methods--that receive little, incorrect, or no coverage. And to make foundation knowledge accessible to the thinking database professional and user, regardless of the DBMS used.

- ***BUSINESS MODELING FOR DATABASE DESIGN***
- ***THE COSTLY ILLUSION: NORMALIZATION, INTEGRITY AND PERFORMANCE***
- ***TRULY RELATIONAL-WHAT IT <u>REALLY</u> MEANS***
- ***THE FINAL NULL IN THE COFFIN***

And much more.