Big databases jump on the JSON bandwagon

Database titans like Oracle and IBM can no longer ignore the NoSQL insurgents. Are their efforts to accommodate JSON serious, or superficial?

InfoWorld | Aug 25, 2014

Web developers love JSON (JavaScript Object Notation). Like XML, it's a human-readable format for transmitting data, except it's a whole lot easier to work with than XML. It just stands to reason that Web developers would find JavaScript a much more pleasant way of accessing data than writing to a SQL database.

Several NoSQL databases -- including the wildly successful MongoDB and Couchbase -- store data in JSON documents natively. No surprise then, as MongoDB CTO Eliot Horowitz acknowledged in an interview I did a few weeks ago, that developers have been "the No. 1 group" driving NoSQL document database adoption.

[First look: MongoDB 2.6, the all-grown-up version | Also on InfoWorld: The time for NoSQL standards is now | Get a digest of the key stories each day in the InfoWorld Daily newsletter.]

SQL database vendors have not been blind to JSON's appeal. Last month, Oracle weighed in with JSON support in Oracle 12c version 12.1.0.2, which Oracle's Andrew Mendelsohn detailed at the NoSQL Now conference last week. Last September, PostgreSQL -- widely considered the leading open source alternative to Oracle -- debuted

support for JSON in its 9.3 version, enhanced just last week by the release of a developer kit to make it easier to exploit PostgreSQL's JSON functionality.

Even stodgier companies are catching on. Teradata announced last October that customers could add JSON data to their data warehouses. IBM was actually ahead of them all when in June of last year it said that DB2 would support MongoDB's JSON-oriented query method. Microsoft is taking a different approach: SQL Server doesn't yet accommodate JSON queries, so instead the company announced Azure DocumentDB, a native document DBaaS (database as a service) for the Azure cloud.

It seems clear that the database incumbents are feeling the heat from NoSQL insurgents, particularly MongoDB. But are these JSON concessions check box features or serious attempts to deliver usable MongoDB-like functionality?

Until we do some testing, I can't offer a real answer. But one thing I can say for certain is that retrofitting JSON support into a traditional RDBMS delivers only part of the NoSQL value proposition. If JSON support is implemented well, it's conceivable that on the coding side developers won't mind JSON documents being stuffed into tables -- as with NoSQL, they may be insulated from the rigmarole RDBMS databases demand when data structures change.

But will it perform? Half the value proposition of NoSQL is the ability to scale out rather than up -- that is, to snap on commodity servers to increase performance rather than beefing up a monster database server. Also, the way a document database distributes data across multiple servers increases reliability: If a node goes down, the database keeps chugging along without data loss.

I spoke with InfoWorld's own Andrew Oliver last week, who has been involved in a number of MongoDB deployments. Although he also reserves judgment until RDBMS's JSON support can be evaluated, he feels that Oracle and other SQL databases "lack the semantics and underpinnings to scale" when loaded with JSON documents. In part, he says, that's due to the aforementioned scale-out problem. But a simpler way to look at it is that you should use the right tool for the job.

For tabular data, you're probably best advised to choose a relational solution (or in some cases a key-value or column family database). For analysis of semi-structured data such as log files, choose Hadoop. If your data is document-oriented, choose a document database. Really, document databases are best suited to Web applications where you need lightning-fast reads and expect highly variable numbers of users. Plus, part of the deal, frankly, is to avoid Oracle-sized licensing and lock-in.

That said, Andrew thinks someone (not Oracle) will eventually succeed on integrating NoSQL and RDBMS and probably Hadoop capabilities into a single database solution. I was intrigued by a persuasive essay written in May by VoltDB CTO Ryan Betts and posted in InfoWorld's New Tech Forum, which said: "In reality, it will be substantially easier for relational vendors to extend their systems to include JSON than it will be for the new document store vendors to retrofit their non-relational systems with the features lost in the move away from relational structures."

That certainly seems true, as the recent announcements of JSON support indicate. But I'm also willing to bet that simply retrofitting an existing RDBMS with JSON support will not result in a solution that Web developers will love.

This article, "Big databases jump on the JSON bandwagon," originally appeared at InfoWorld.com. Read more of Eric Knorr's Modernizing IT blog. And for the latest business technology news, follow InfoWorld on Twitter.

Eric Knorr — *Editor in Chief*

Eric Knorr is the editor in chief for InfoWorld and has been with the publication since 2003. Eric has received the Neal and Computer Press Awards for journalistic excellence.