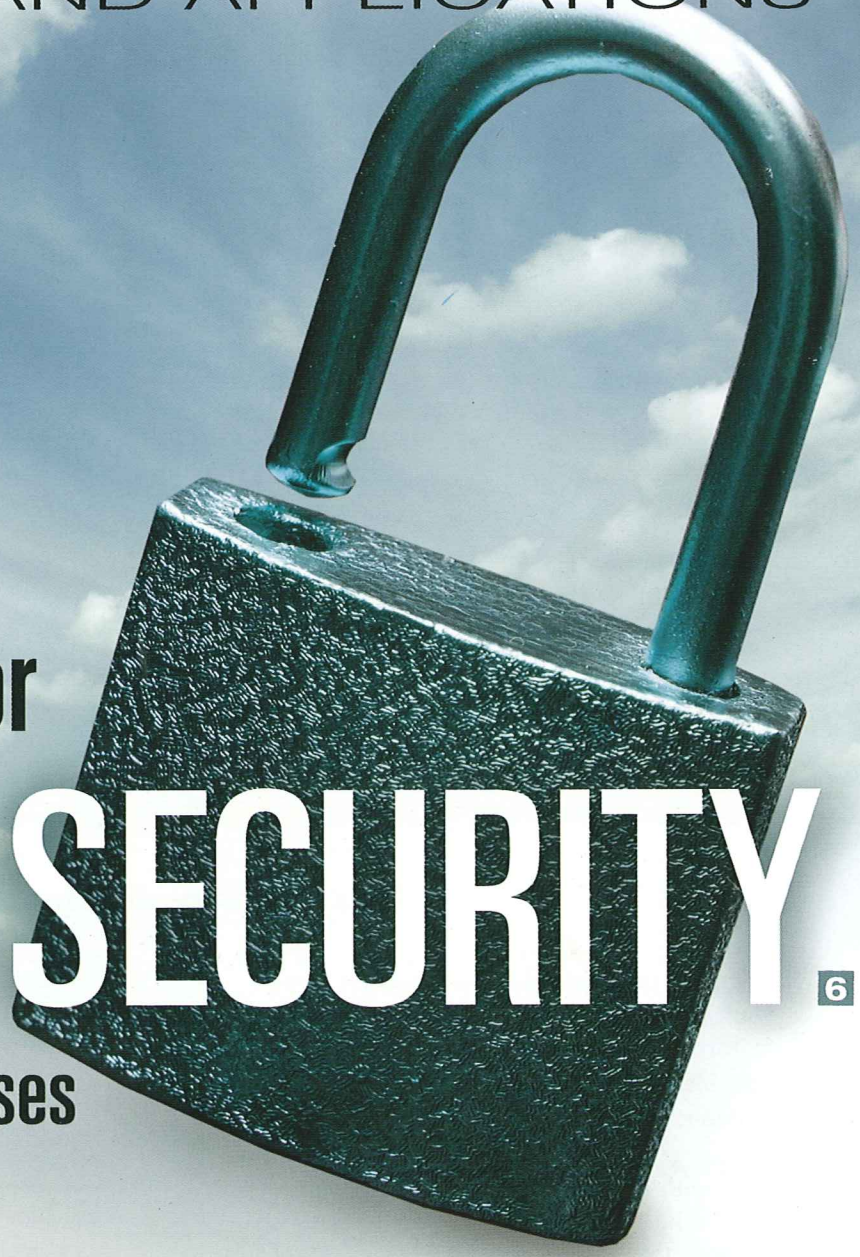
Volume 25 Number 1
MARCH 2011

The Journal of
Information Integration and Management

database
TRENDS AND APPLICATIONS



The Cloudy Prognosis for DATA SECURITY

in Virtual Enterprises

WWW.DBTA.COM

TRENDS

Triplestores: A NoSQL Option

By Dr. Jans Aasman

RELATIONAL DATABASES (RDBMSs) have been the dominant data management tool for 30 years. They proved to be a good solution for the capture and management of structured data and fairly reasonable for decision support analysis. Their shortcomings, however, have become increasingly obvious in recent years as unstructured information has begun flooding into the data center.

Business leaders are realizing the tremendous value of this unstructured data, which includes corporate email, documents, video, audio, still images, and social networking data, for such vital uses as:

1. Predicting market trends
2. Identifying informal relationship structures and key influencers inside large enterprises and in external markets
3. Targeting marketing investments to gain the most advantage in the market
4. Predicting the needs of individual customers in order to increase service levels while decreasing costs

NoSQL as an Alternative?

To capture these diverse data types and support this type of analysis, businesses have turned to two new classes of database technology: big data systems (or key/value systems) such as Hadoop and Hbase, and semantic web systems, aka "triplestores." These have been lumped into the general term of "not only SQL" (NoSQL) and are typically not seen as replacements but rather supplements to RDMBSs, with the capability of organizing very large volumes of both structured and unstructured data and combining them in various kinds of analysis. Each of these has its own strengths and weaknesses and its own natural application areas.

Relational databases are strongest in regular enterprise applications that only deal with structured data. The enterprise values in particular the transactionality and ACID (atomicity, consistency, isolation, durability) properties of the relational database model.

Big data technologies are designed to work with billions of nested objects (a webpage, a Facebook account, etc.) that by virtue of its size needs to run on large clusters of machines. These big data databases don't have the rigor of a relational database when it comes to transactions and ACIDness and they have given up on doing any complex joins, but they do an amazing job at making billions of objects available for millions of requests per second.

Semantic web triplestore databases are best at complex metadata applications where the number of classes change on a day-by-day basis, where classes can change on-the-fly, and where it is really important to have self descriptions of data. Modern triplestores have developed to the point where they offer the rigor of relational databases, the scalability of big data systems, and still support big complicated joins.

In particular, NoSQL as the "big data" type of database has been a movement to offer nonrelational distributed data storage that does not try to provide full ACID compliance. These offerings provide weak consistency guarantees such as eventual consistency and transactions restricted to single data items. While this offers significant flexibility and scaling, it may not be the best choice for primary storage of business-critical data.

The Hbases or big data databases are designed to accept very high volumes of data objects that are largely self-contained and involve very few joins. Like the RDBMSs they are very good at concurrent dynamic access. Big data systems also provide high availability. One thing they cannot do well is complex graph searches, and they are not good at combining structured and unstructured data, two areas where triplestores excel. Triplestores offer a viable option for NoSQL flexibility along with the ACID compliance you need from RDBMSs. The scaling capabilities of triplestores are continually maturing, and we are starting to see large-scale projects rely on triplestores in an enterprise setting.

Need Ultimate Flexibility? Triplestores Come Out on Top

The highly structured nature of RDBMSs makes them inflexible in the kinds of data they can accept. If you would want to add relationships between data, you would have to overhaul your schema system and add new link tables. In comparison, triplestores offer several ways of adding new relationships.

For instance, in the triplestore data model shown in Figure 1—Conceptual Triplestore Model, the simplest way to add a new relationship is to add a triple like "person1 uncle-of person2," with no need to make a new schema and add a new link table. Just add this new triple and now you can ask new queries involving uncles.

The disadvantage of this approach is that you would have to add a lot of triples to record all these family relationships. Thus, it is faster to just add a few rules, such as:

- if p0 has-child p1 and p0 has-child p2 then p1 has-sibling p2.
- p1 uncle-of p3 if p1 is male & p1 has-sibling p2 & p2 has-child p3.

Triplestores are highly flexible, making the addition of new information not anticipated in the original database design far more straightforward. In fact, triplestore databases are so flexible that database designers do not have to create a schema up front but can build an ontology based

Modern triplestores have developed to the point where they offer the rigor of relational databases, the scalability of big data systems, and still support big complicated joins.

on the data they need to include, editing it as they go. But nothing prevents the designer from creating an initial ontology. Because of this structural flexibility it is easy to integrate databases in an almost lazy, bottom-up fashion.

In the traditional top-down master data approach, you spend an eternity getting the entire "truth" for all the data that you will integrate. With the triple store approach, you can keep (most of) the data in the original databases and slowly start building a set of triples and rules to integrate your data.

Complex Event Analysis? Triplestores Win Hands Down

We see a number of companies requiring event analysis with real-time, complex query capabilities. These companies are using large data warehouses with disparate RDF (Resource Description Framework)-based triple stores describing various types of events, where each event has at least two actors, usually a beginning and end time, and very often a geospatial component. These events are literally everywhere:

- In healthcare applications, we see hospital visits, drugstore visits, and medical procedures.
- In the communications industry, we see telephone call detail records including locations.
- In large corporations, email and calendar databases are basically social network databases filled with events in time and, in many cases, space.
- In the financial industry, every transaction is essentially an event.
- In the insurance industry, claims are important events that need more activity recognition.
- In the homeland security industry, basically everything focuses on events and actors.

So How Can Triplestores Help With This?

Some triplestores now offer social network analysis libraries and efficient geospatial and temporal indexing. With these capabilities they can do queries such as "find all meetings that happened in November 2010 within 5 miles of Berkeley that were attended by the three most influential people among Joe's friends and friends-of-friends." This kind of relationship analysis is becoming important in business both for the identification of macro trends and micro opportunities for sales to individual customers, and in governmental areas such as intelligence and defense.

This complex relationship analysis is nearly impossible to do with traditional RDMBSs, which are too inflexible to capture data on complex, evolving relationships effectively, while big data systems cannot accommodate the large numbers of joins required. Semantic technologies, however, can provide these insights and adapt their answers to changing conditions and increased data availability, making them ideal for the kind of pattern recognition analysis that is the heart of both market trend identification and intelligence.

Where Are Triplestores Used Today?

Triplestore technologies are already in use in several industries including pharmaceuticals, the defense industry (and the U.S. Department of Defense), telecommunications, media companies, and IT. They are used in such areas as:

- The analysis of the relative effectiveness of different cancer drugs in combination with other treatments on different patient populations
- The capture and analysis of detailed information on very large numbers of companies and the interrelationships among them
- The analysis of how all the customers of large cell phone providers use their phones and which, for instance, are good prospects for plan upgrades
- The integration of multiple complex databases such as those that enter a large enterprise as part of acquisitions

subject	predicate	object
person2	type	person
person2	first-name	Rose
person2	middle-initial	E
person2	last-name	Fitzgerald
person2	suffix	none
person2	alma-mater	Harvard
person2	birth-year	1890
person2	death-year	1995
person2	sex	female
person2	spouse	person1
person2	has-child	person17
person2	has-child	person15
person2	has-child	person13
person2	has-child	person11
person2	has-child	person9
person2	has-child	person7
person2	has-child	person6
person2	has-child	person4
person2	has-child	person3
person2	profession	home-maker

Figure 1: Conceptual Triplestore Model

A Combination Is Best

A successful combination of technologies is an ideal approach. Wholesale replacement of your RDBMS or NoSQL investment is a fool's errand. A more practical approach is using a triplestore to "add a brain" to your legacy system. For a NoSQL approach, a combined system could provide fast, scalable access to the full content, with the inference and aggregation from a triplestore that is needed for the added richness to round out the solution.

Dr. Jans Aasman is CEO of Franz, Inc., a supplier of graph database technology for the semantic web.