Dataconomy article – Triple Attributes: A New Way to Protect the Most Sensitive Information

Semantic Graph Databases are now common in many industries, including life sciences, healthcare, the financial industry and in government and intelligence agencies. Graphs are particularly valuable in these sectors because of the complex nature of the data and need for powerful, yet flexible data analytics.

Attributes, user attributes and static filters are a new mechanism for graph databases to protect sensitive information. This combination provides the right amount of power and flexibility to address high-security use cases, such as: HIPAA access controls, privacy rules for banks, security models for policing, intelligence and the government. In addition, this set of methods is far easier to use, provides more expressiveness than security methods in relational databases and doesn’t suffer from performance degradations.

Many industries today have critical and sensitive data that would be disastrous if exposed to the wrong people. In financial industries, it might be purely about privacy, but for other industries it might be a matter of immediate life or death. Wired Magazine* described a typical security problem in the area of policing and intelligence:

“When Sergeant Lee DeBrabander marked a case confidential in the Long Beach drug squad’s Palantir data analysis system in November 2014, he expected key details to remain hidden from unauthorized users’ eyes. In police work, this can be crucial- a matter of life and death, even. It often involves protecting

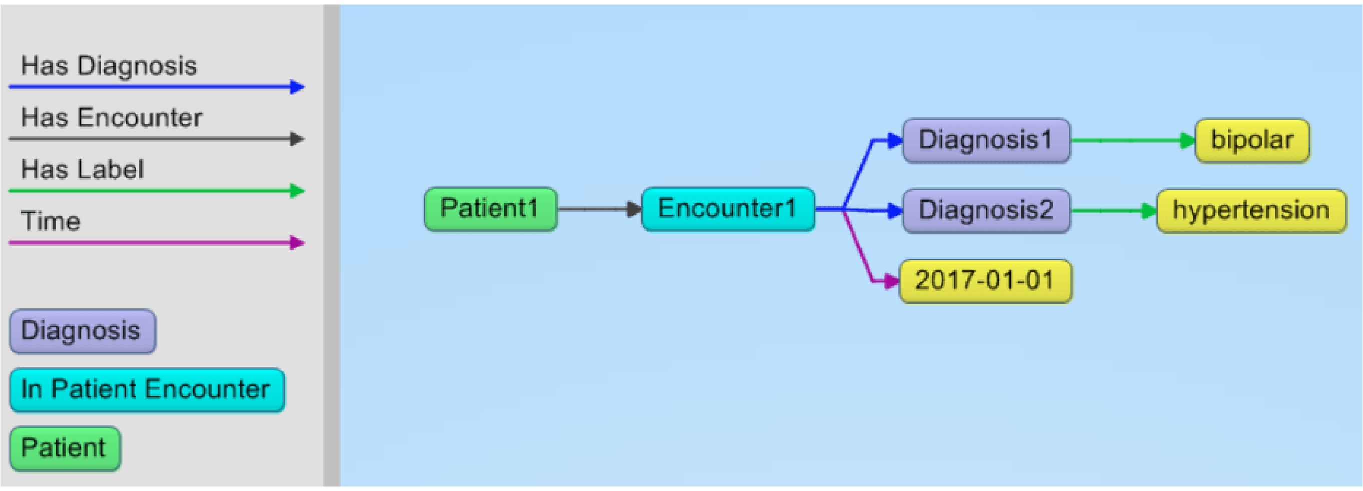
vulnerable witnesses, keeping upcoming operations hush hush, or protecting a fellow police officer who's working undercover. Yet not long after, someone working in the gang crimes division ran a car license plate mentioned in his case and was able to read the entire file. https://www.wired.com/story/how-peter-thiels-secretive-data-company-pushed-into-policing/

With respect to security, two issues come up. Although enterprises want the flexibility of graph databases, they also want the *fine grained* security that they have come to rely on with relational databases like Oracle. The second issue is that obtaining fine grained security is notoriously hard to implement with relational databases. For example, it takes a specialist to protect individual cells in a complex set of tables.

But both these issues can be addressed by creating Triple Attributes in combination with User Attributes and a Filter Policy language. This method provides a new elegant mechanism to implement the ultimate in graph database security.

Semantic Graph Databases, triples and RDF. According to recent reports the majority of enterprises now engage in projects that involve graph databases. Graph databases are the most fine grained way to organize data, whereas relational databases organize data around tables, rows and columns, Graph database organize data around nodes and links between nodes. Nodes usually represent real world or conceptual objects and links represent the relationships between these nodes. Semantic graph databases are a richer version of graph databases in that the nodes and links between nodes are URIs according to the W3C standard for RDF. Semantic graph databases store their graphs as triples, that is, each node-link-node combination in a graph is stored as a triple.

A simple example: say we want to express the fact that a patient had an encounter on January 1 2017. During that encounter we had two diagnoses. One for a mental illness and one for hypertension.



And here are the triples that created this picture.

:patient1 :hasEncounter :encounter1 .

:encounter1 :type :inPatientEncounter .

:encounter1 :time “2017-01-01” .

:encounter1 :hasDiagnosis :diagnosis1 .

:encounter1 :hasDiagnosis :diagnosis2 .

:diagnosis1 :hasLabel “bipolar” .

:diagnosis2 :hasLabel “hypertension” .

Triple Attributes. Triple attributes provide metadata for individual triples. That is, for every triple you may add a json object with a set of key/value pairs. Typical uses of triple attributes are date-time, weight, security level, classification level, trust level and provenance information but in general any user defined attribute is allowed. These attributes can be accessed in queries and graph algorithms and greatly expand the power of Semantic Graph databases.

User Attributes. When a user submits a query to the application server that is connected to the Semantic Graph database the query will get prefixed by all his user attributes. These attributes who are most likely stored in a LDAP like storage are also json objects that comprise a set of key/value pairs. These User Attributes are important for the Static Filters that will determine what triples can be used in the users queries and graph algorithms.

Static Filters. In itself triple attributes or user attributes do not provide security as they are 'just' meta data that can be used for many purposes. Security comes with the use of Static Filters. Static filters provide a rich and expressive language to compare triple attributes with user attributes and determine what triples can be used in queries and algorithms. Static filters are compiled to machine code and can work at the lowest level of the database to ensure no performance degradation.

Let us go back to the triple examples we showed above. Now imagine that in a hospital a user of the database needs a higher security level for mental illness than for other illnesses. And in addition: the user needs to have the right role or roles to see the right triples. Note how in the following example we add the attributes seclevel and role to only two triples.

:person1 :hasEncounter :encounter1 .

:encounter1 :type :inPatientEncounter .

:encounter1 :time "2017-01-01" .

:encounter1 :hasDiagnosis :diagnosis1 . { "seclevel" : "8",

"role" : ["clinician", psychiatrist"]}

:encounter1 :hasDiagnosis :diagnosis2 . { "seclevel" : "3", "role" : ["clinician"] }

:diagnosis1 :hasLabel "bipolar" . :diagnosis2 :hasLabel “hypertension” .

An example static filter:

```
(and    (attribute-set>= user.seclevel triple.seclevel)

          (overlap user.role triple.role))
```

An example query would be if someone with the role of administrator with a security level 2 trying to find the persons in the database that are bipolar. Obviously this query will fail in the example above.

Conclusion:

Semantic Graph databases allow visibility across the data. Organizations can share data and show how data is connected. With the added capability of Triple Attributes, User Attributes and Filter Policies – data can be made transparent to users based on roles. This provides the power of Graph databases with the security of need-to-know access. This approach has the flexibility to implement HIPAA for the healthcare industry, the privacy rules for the financial industry and the government models and policies for classified information.

Like this article? Subscribe to our weekly newsletter to never miss out!

Follow @DataconomyMedia