

Why Smart Cities Need AI Knowledge Graphs

A linked data framework can empower smart cities to realize social, political, and financial goals.



Smart cities are projected to become one of the most prominent manifestations of the Internet of Things (IoT). Current estimates for the emerging smart city market exceed \$40 trillion, and San Jose, Barcelona, Singapore, and many other major metropolises are adopting smart technologies.

The appeal of smart cities is binary. On the one hand, the automated connectivity of the IoT is instrumental in reducing costs associated with public expenditures for infrastructure such as street lighting and transportation. With smart lighting, municipalities only pay for street light expenses when people are present. Additionally, by leveraging options for dynamic pricing with smart parking, for example, the technology can provide new revenue opportunities.

Despite these advantages, smart cities demand extensive data management. Consistent data integration from multiple locations and departments is necessary to enable interoperability between new and legacy systems. Smart cities need granular data governance for long-term sustainability. Finally, they necessitate open standards to future-proof their perpetual utility.

Knowledge graphs—enterprise-wide graphs which link all data assets for internal or external use—offer all these benefits and more. They deliver a uniform, linked framework for sharing data in accordance with governance protocols, are based on open standards, and exploit relationships between data for business and operational optimization. They supply everything

smart cities need to realize their social, political, and financial goals. Knowledge graphs can use machine learning to reinsert the output of contextualized analytics into the technology stack, transforming the IoT's copious data into foundational knowledge to spur improved civic applications.

Read the full article at [Trajectory Magazine](#)



Adding Properties to Triples in AllegroGraph

AllegroGraph provides two ways to add metadata to triples. The first one is very similar to what typical property graph databases provide: we use the named graph of triples to store meta data about that triple. The second approach is what we have termed *triple attributes*. An attribute is a key/value pair associated with an individual triple. Each triple can have any number of attributes. This approach, which is built into AllegroGraph's storage layer, is especially handy for security and bookkeeping purposes. Most of this article will discuss triple attributes but first we quickly discuss the named graph (i.e. fourth element or quad) approach.

1.0 The Named Graph for Properties

Semantic Graph Databases are actually defined by the W3C standard to store RDF as 'Quads' (Named Graph, Subject, Predicate, and Object). The 'Triple Store' terminology has stuck even though the industry has moved on to storing quads. We believe using the named graph approach to store metadata about triples is richer model than the property graph database method.

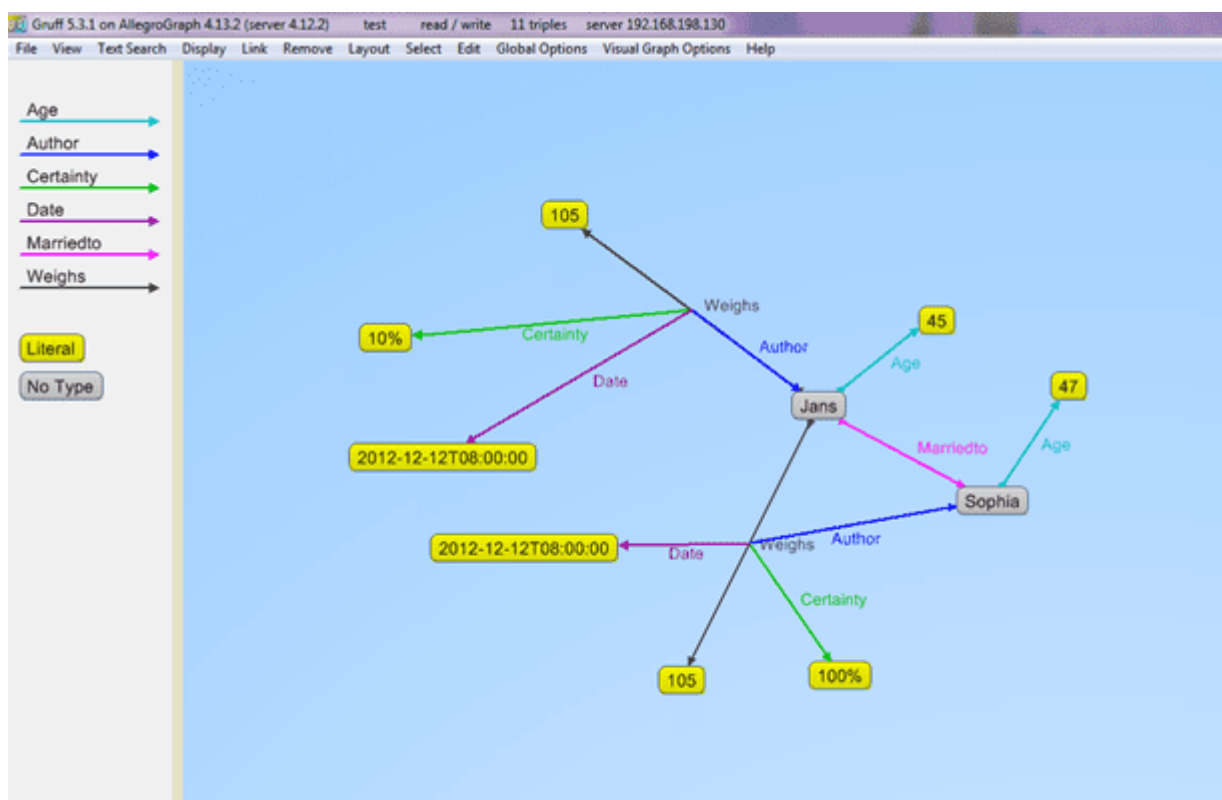
The best way to understand this is to give an example. Below we see two statements about Bruce weighing 105 kilos. The triple portions (subject, predicate, object) are identical but the named graphs (fourth elements) differ. They are used to provide additional information about the triples. The graph values are S1 and S2. By looking at these graphs we see that

- The author of the first triple (with graph S1) is Sophia and the author of the second (with graph S2) is Bruce (who is also the subject of the two triples).
- Sophia is 100% certain about her statement while Bruce is only 10% certain about his.

Using the named graph we can do even more than a property graph database, as the value of a graph can itself be a node, and is the subject of various triples which specify the original triple's author, date, and certainty. Additional triples tell us the ages of the authors and the fact that the authors are married.

Subject	Predicate	Object	Graph
Bruce	Weighs	105kg	S1
Bruce	Weighs	105kg	S2
Bruce	Age	45	None
Bruce	<u>MarriedTo</u>	Sophia	None
S1	Author	Sophia	None
S1	Date	2012-12-12T08:00:00	None
S1	Certainty	100 %	None
S2	Author	Bruce	None
S2	Date	2012-12-12T08:00:00	None
S2	certainty	10%	None
Sophia	Age	47	None

Here is the data displayed in Gruff, AllegroGraph's associated triple store browser:



Using named graphs for a triple's metadata is a powerful tool but it does have limitations: (1) only one graph value can be associated with a triple, (2) it can be important that metadata is stored directly and physically with the triple (with named graphs, the actual metadata is usually stored in additional triples with the graph as the subject, as in the example above), and (3) named graphs have competing uses and

may not be available for metadata.

2.0 The Triple Attributes approach

AllegroGraph uniquely offers a mechanism called *triple attributes* where a collection of user defined key/value pairs can be stored with each individual triple. The advantage of this approach is manifold, but the original use case was designed for triple level security for an Intelligence agency.

By having triple attributes physically connected to the triples in the storage layer we can provide a very powerful and flexible mechanism to protect triples at the lowest possible level in AllegroGraph's architecture. Our first example below shows this use case in great detail. Other use cases are for example to add weights or costs to triples, to be used in graph algorithms. Or we can add a recorded time or expiration times to a triple and use that to provide a time machine in AllegroGraph or do automatic clean-up of old data.

Example with Attributes:

Subject – <http://dbpedia.org/resource/Arif_Babayev>

Predicate – <<http://dbpedia.org/property/placeOfDeath>>

Object – <<http://dbpedia.org/resource/Baku>>

Named Graph – <<http://ex#trans@@1142684573200001>>

Triple Attributes – {"securityLevel": "high",
"department": "hr", "accessToken": ["E", "D"]}

This article provides an initial introduction to attributes and the associated concept of static filters, showing how they are set up and used. We start with a security example which also describes the basics of adding attributes to triples and

filtering query results based on attribute values. Then we discuss other potential uses of attributes.

2.1 Triple Attribute Basics: a Security Example

One important purpose of attributes, when they were added as a feature, was to allow for very fine triple-level security, so that triples would be visible or invisible to users according to the attributes of the triples and the permissions associated with the query being posed by the user.

Note that users as such do not have attributes. Instead, attribute values are assigned when a query is posed. This is an important point: it is natural to think that there can be an attribute SECURITY-LEVEL, and a triple can have attribute SECURITY-LEVEL=3, and USER1 can have an attribute SECURITY-LEVEL=2 and USER2 can have an attribute SECURITY-LEVEL=4, and the system can require that the user SECURITY-LEVEL attribute must be greater than the triple SECURITY-LEVEL for the triple to be visible to the user. But that is not how attributes work. The triples can have the attribute SECURITY-LEVEL=2 but users do not have attributes. Instead, the filter is made part of the query.

Here is a simple example. We define attributes and static attribute filters using AGWebView. We have a repository named repo. Here is a portion of its AGWebView page:

Repository repo — 0 statements

[\[edit description\]](#)

Load and Delete Data

- [Add a statement](#)
- [Delete statements](#)
- [Import RDF:](#)
 - [from an uploaded file](#)
 - [from a server-side file](#)
 - [from a text area input](#)

Explore the Repository

- [View triples](#)
- [View quads](#)
- [View repository's classes](#)
- [View repository's predicates](#)
- [View repository's named graphs](#)

Reports

- [Storage report](#)
- [Triple indices](#)
- [String table](#)
- [Full list of reports ...](#)


Multi-Master Replication

- [Convert store to a replication instance](#)

Warm Standby Replication

- [Control replication](#)

Repository Control

- [Export repository as](#)
- [Start a session](#) — support transactions and Prolog functors
- [Warmup store](#)
- [Back-up this repository](#)
- [Export duplicate statements](#)
- [Delete duplicate statements](#)
- [Suppress duplicate statements](#)
- [View active transactions](#)
- [Recognize geospatial datatypes automatically:](#) ☐
- [Control durability \(bulk-load mode\)](#)
- [Manage attribute definitions](#)
- [Set static attribute filter](#)
- [Manage triple indices](#)
 - [Optimize the repository](#) 



The red arrow points to the commands of interest: Manage attribute definitions and Set static attribute filter. We click on Set static attribute filter to define an attribute. We have filled in the attribute information (name *security-level*, minimum and maximum number allowed per triple, allowed values, and whether order or not (yes in our case):

AllegroGraph WebView

repository repo

[Repository](#) | [Queries](#) | [Utilities](#) | [Admin](#) | [User test](#)

New Attribute

Name

security-level

Min. number

0

Max. number

1

Allowed Values

0,1,2,3,4,5

Ordered

☒

Save

Cancel

Attribute Definitions

+ Add New

Name	Min. number	Max. number	Allowed values	Ordered	
No attributes have been defined					

We click Save and the attribute is defined:

AllegroGraph WebView

repository repo

[Repository](#) | [Queries](#) | [Utilities](#) | [Admin](#) | [User test](#)

Attribute Definitions

+ Add New

Name	Min. number	Max. number	Allowed values	Ordered	
security-level		1	0,1,2,3,4,5	✓	

Then we define a filter (on the Set static attribute filter page) :

AllegroGraph WebView

repository repo

[Repository](#) | [Queries](#) | [Utilities](#) | [Admin](#) | [User test](#)

Static Filter

Current filter

(attribute-set> user.security-level triple.security-level)

Edit filter

(attribute-set> user.security-level triple.security-level)

Save

Revert to current

Clear

We defined the filter `(attribute-set> user.security-level triple.security-level)` and clicked Save (the definition appears in both the Edit and the Current fields). The filter says that the “user” security level must be greater than the triple security level. We put “user” in quotes because the user security level is specified as part of the query, and has no direct connection to any specific user.

Here are some triples in a nqx file *fr.nqx*. The first triple has no attributes and the other three each has a security-level attribute value.

```
<http://www.franz.com#position> "intern" .
```

```
<http://www.franz.com#emp1>
```

```
<http://www.franz.com#position> "worker" {"security-level":  
"2"} .
```

```
<http://www.franz.com#emp2>
```

```
<http://www.franz.com#position> "manager" {"security-level":  
"3"} .
```

```
<http://www.franz.com#emp3>
```

```
<http://www.franz.com#position> "boss" {"security-level": "4"}  
.
```

We load this file into a repository which has the security-level attribute defined as above and the static filter mentioned above also defined. (Triples with attributes can also be entered directly when using AGWebView with the Import RDF from a text area input command).

Once the triples are loaded, we click View triples in AGWebView and we see no triples:

AllegroGraph WebView repository repo

[Repository](#) | [Queries](#) | [Utilities](#) | [Admin](#) | [User test](#)

Edit query

1 # View triples

2 SELECT ?s ?p ?o { ?s ?p ?o . }

Execute

Log Query

Show Plan

Save as

Add to repository

No results

This result is often surprising to users just beginning to work with attributes and filters, who may expect the first triple, abbreviated to [emp0 position intern], to be visible, but the system is doing what it is supposed to do. It will only show triples where the security-level of the user posing the query is greater than the security level of the triple. The user has no security level and so the comparison fails, even with triples that have no security-level attribute value. We will describe below how to ensure you can see triples with no attributes.

So we need to specify an attribute value to the user posing the query. (As said above, users do not themselves have attribute values. But the attribute value of a user posing a query can be specified as part of the query.) “User” attributes are specified with a prefix like the following:

prefix franzOption_userAttributes: <franz:%7B%22security-

```
level%22%3A%223%22%7D>
```

so the query should be

```
prefix franzOption_userAttributes: <franz:%7B%22security-  
level%22%3A%223%22%7D>
```

```
select ?s ?p ?o { ?s ?p ?o . }
```

We will show the results below, but first what are all the % signs and numbers doing there? Why isn't the prefix just `prefix franzOption_userAttributes: <franz:{"security-level":"3"}>`? The issue is that `{"security-level":"3"}` won't read correctly. It must be URL encoded. We do this by going to <https://www.urlencoder.org/> (there are other websites that do this as well) and put `{"security-level":"3"}` in the first box, click Encode and get `%7B%22security-level%22%3A%223%22%7D`. We then paste that into the query, as shown above.

When we try that query in AGWebView, we get one result:

AllegroGraph WebView

repository repo

[Repository](#) | [Queries](#) | [Utilities](#) | [Admin](#) | [User test](#)

Edit query

```

1 prefix franzOption_userAttributes: <franz:%7B%22security-level%22%3A%225%22%7D>
2 select ?s ?p ?o { ?s ?p ?o . }
3

```

Execute

Log Query

Show Plan

Save as

Add to repository

1 Result in 2.619 ms

Information

s	p	o
<http://www.franz.com#emp1>	<http://www.franz.com#position>	"worker"

If we encode {"security-level": "5"} to get the query

```

prefix  franzOption_userAttributes:  <franz:%7B%22security-
level%22%3A%225%22%7D>
select ?s ?p ?o { ?s ?p ?o . }

```

we get three results:

emp3	position	"boss"
emp2	position	"manager"
emp1	position	"worker"

since now the "user" security-level is greater than that of any triples with a security-level attribute. But what about

the triple with subject emp0, the triple with no attributes? It does not pass the filter which required that the user attribute be greater than the triple attribute. Since the triple has no attribute value so the comparison failed.

Let us redefine the filter to:

```
(or (attribute-set> user.security-level triple.security-level)
    (empty triple.security-level))
```

The screenshot shows the AllegroGraph WebView interface. At the top, there is a header bar with the text "AllegroGraph WebView" and "repository repo". Below this is a navigation bar with links: "Repository", "Queries", "Utilities", "Admin", and "User test". The main content area is titled "Static Filter". Under this title, there are two sections: "Current filter" and "Edit filter". Both sections contain the same filter expression:

```
(or (attribute-set> user.security-level triple.security-level)
    (empty triple.security-level))
```

. At the bottom of the "Edit filter" section, there are three buttons: "Save", "Revert to current", and "Clear".

Now a triple will pass the filter if either (1) the “user” security-level is greater than the triple security-level or

(2) the triple does not have a security-level attribute. Now the query from above where the user has attribute security-level:"5" will show all the triples with security-level less than 5 and with no attributes at all. That happens to be all four triples so far defined:

AllegroGraph WebView repository repo

Repository | Queries | Utilities | Admin | User test

Edit query

```
1 prefix franzOption_userAttributes: <franz:%7B%22security-level%22%3A%225%22%7D>
2 select ?s ?p ?o { ?s ?p ?o . }
3
```

Execute Log Query Show Plan Save as Add to repository

4 Results in 0.286 ms Information

s	p	o
<http://www.franz.com#emp0>	<http://www.franz.com#position>	"intern"
<http://www.franz.com#emp3>	<http://www.franz.com#position>	"boss"
<http://www.franz.com#emp2>	<http://www.franz.com#position>	"manager"
<http://www.franz.com#emp1>	<http://www.franz.com#position>	"worker"

The triple

emp0 position "intern"

will now appears as a result in any query where it satisfies the SPARQL select regardless of the security-level of the

“user”.

It would be a useful feature that we could associate attributes with actual users. However, this is not as simple as it sounds. Attributes are features of repositories. If I have a REP01 repository, it can have a bunch of defined attributes and filters but my REP02 may know nothing about them and its triples may not have any attributes at all, and no attributes are defined, and (as a result) no filters. But users are not repository-linked objects. While a repository can be made read-only or unreadable for a user, users do not have finer repository features. So an interface for providing users with attributes, since it would only make sense on a per-repository basis, requires a complicated interface. That is not yet implemented (though we are considering how it can be done).

Instead, users can have specific prefixes associated with them and that prefix and be included in any query made by the user.


But if all it takes to specify “user” attributes is to put the right line at the top of your SPARQL query, that does not seem to provide much security. There is a feature for users “Allow user attributes via SPARQL PREFIX `franzOption_userAttributes`” which can restrict a user’s ability to specify “user” attributes in a query, but that is a rather blunt instrument. Instead, the model is that most users (outside of trusted administrators) are not actually allowed to pose SPARQL queries directly. Instead, there is an intermediary program which takes the query a user requests and, having determined the status of the user and what attribute values should be given to the user, modifies the query with the appropriate `franzOption_userAttributes` prefixes and then sends the query on to the server, following which it captures the results and

sends them back to the requesting user. That intermediate program will store the prefix suitable for a user and thus associate “user” attributes with specific users.

2.2 Using attributes as additional data

Although triple security is one powerful use of attributes, security is far from the only use. Just as the named graph can serve as additional data, so can attributes. SPARQL queries can use attribute values just as static filters can filter out triples before displaying them. Let us take a simple example: the attribute `timeAdded`. Every triple we add will have a `timeAdded` attribute value which will be a string whose contents are a datetime value, such as “2017-09-11T:15:52”. We define the attribute:

Attribute Definitions

+ Add New					
Name	Min. number	Max. number	Allowed values	Ordered	
timeAdded	1	1			

Now let us define some triples:

```
<http://www.franz.com#emp0>
<http://www.franz.com#callRank> "2" {"timeAdded":
"2019-01-12T10:12:45" } .
<http://www.franz.com#emp0>
<http://www.franz.com#callRank> "1" {"timeAdded":
"2019-01-14T14:16:12" } .
<http://www.franz.com#emp0>
<http://www.franz.com#callRank> "3" {"timeAdded":
"2019-01-11T11:15:52" } .
```

```

    <http://www.franz.com#emp1>
<http://www.franz.com#callRank> "5" {"timeAdded":
"2019-01-13T11:03:22" } .
    <http://www.franz.com#emp0>
<http://www.franz.com#callRank> "2" {"timeAdded":
"2019-01-13T09:03:22" } .

```

We have a call center with employees making calls. Each call has a ranking from 1 to 5, with 1 the lowest and 5 the highest. We have data on five calls, four from emp0 and one from emp1. Each triples has a timeAdded attribute with a string containing a dateTime value. We load these into a empty repository named at-test where the timeAdded attribute is defined as above:



SPARQL queries can use the attribute magic properties (see <https://franz.com/agraph/support/documentation/current/triple-attributes.html#Querying-Attributes-using-SPARQL>). We use the attributesNameValue magic property to see the subject, object, and attribute value:

```

select ?s ?o ?value {
                                (?ta      ?value)

```

```

<http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>
  (?s ?p ?o) .
}

```

Repository | Queries | Utilities | Admin | User test

Edit query

```

1 select ?s ?o ?value {
2   (?ta ?value) <http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>   (?s ?p ?o) .
3 }

```

Execute Log Query Show Plan Save as Add to repository

5 Results in 0.363 ms Information

s	o	value
<http://www.franz.com#emp0>	"2"	"2019-01-13T09:03:22"
<http://www.franz.com#emp1>	"5"	"2019-01-13T11:03:22"
<http://www.franz.com#emp0>	"3"	"2019-01-11T11:15:52"
<http://www.franz.com#emp0>	"1"	"2019-01-14T14:16:12"
<http://www.franz.com#emp0>	"2"	"2019-01-12T10:12:45"

But we are really interested just in emp0 and we would like to see the results ordered by time, that is by the attribute value, so we restrict the query to emp0 as the subject and order the results:

```

select ?o ?value {
  (?ta ?value)
  <http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>
  (<http://www.franz.com#emp0> ?p ?o) .
} order by ?value

```

Edit query

```
1 select ?o ?value {  
2   (?ta ?value) <http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>    (<http://www.franz.com#emp0> ?p ?o) .  
3 } order by ?value
```

Execute

Log Query

Show Plan

Save as

Add to repository

4 Results in 0.630 ms

Information

o	value
"3"	"2019-01-11T11:15:52"
"2"	"2019-01-12T10:12:45"
"2"	"2019-01-13T09:03:22"
"1"	"2019-01-14T14:16:12"

There are the results for emp0, who is clearly having difficulties because the call rankings have been steadily falling over time.

Another example using timeAdded is employee salary data. In the Human Resources data, the salary of an employee is stored:

emp0 hasSalary 50000

Now emp0 gets a raise to 55000. So we delete the triple above and add the triple

emp0 hasSalary 55000

But that is not satisfactory because we have lost the salary

history. If the boss asks “How much was emp0 paid initially?” we cannot answer. There are various solutions. We could define a salary change object, with predicates effectiveDate, previousSalary, newSalary, and so on:

```
salaryChange017 forEmployee emp0
salaryChange017 effectiveDate "2019-01-12T10:12:45"
salaryChange017 oldSalary "50000"
salaryChange017 newSalary "55000"
```

emp0 hasSalaryChange salaryChange017

and that would work fine, but perhaps it is more setup and effort than is needed. Suppose we just have `hasSalary` triples each with a `timeAdded` attribute. Then the current salary is the latest one and the history is the ordered list. Here that idea is worked out:

```
<http://www.franz.com#emp0>    <http://www.franz.com#hasSalary>
"50000"^^<http://www.w3.org/2001/XMLSchema#integer>
{"timeAdded": "2017-01-12T10:12:45" } .

<http://www.franz.com#emp0>    <http://www.franz.com#hasSalary>
"55000"^^<http://www.w3.org/2001/XMLSchema#integer>
{"timeAdded": "2019-03-17T12:00:00" } .
```

What is the current salary? A simple SPARQL query tells us:

```

select ?o ?value {
                                (?ta      ?value)
<http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>
                                (<http://www.franz.com#emp0>
<http://www.franz.com#hasSalary> ?o) .
    } order by desc(?value) limit 1

```

Edit query

```
1 select ?o ?value {  
2   (?ta ?value) <http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>  
3   (<http://www.franz.com#emp0> <http://www.franz.com#hasSalary> ?o) .  
4 } order by desc(?value) limit 1
```

Execute

Log Query

Show Plan

Save as

Add to repository

1 Result in 0.388 ms

Information

o	value
"55000"	"2019-03-17T12:00:00"

The salary history is provided by the same query without the LIMIT:

```
select ?o ?value {  
    (?ta ?value)  
<http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>  
    (<http://www.franz.com#emp0>  
<http://www.franz.com#hasSalary> ?o) .  
    } order by desc(?value)
```

Edit query

```
1 select ?o ?value {  
2   (?ta ?value) <http://franz.com/ns/allegrograph/6.2.0/attributesNameValue>  
3   (<http://www.franz.com#emp0> <http://www.franz.com#hasSalary> ?o) .  
4 } order by desc(?value)
```

Execute

Log Query

Show Plan

Save as

Add to repository

2 Results in 0.413 ms

Information

o	value
"55000"	"2019-03-17T12:00:00"
"50000"	"2017-01-12T10:12:45"

This method of storing salary data may not easily support more complex questions which might be easily answered if we went the salaryChange object route mentioned above but if you are not looking to ask those questions, you should not do the extra work (and the risk of data errors) required.

You could use the graph of each triple for the timeAdded. All the examples above would work with minor tweaks. But there are many uses for the named graph of a triple. Attributes are available and using them for one purpose does not restrict their use for other purposes.

Earth Day – Franz Inc. and Geoscience Experts Recognize the Growing Importance of Semantic Knowledge Graphs for Earth Science

Semantically Linking Earth Observation Data Makes it FAIR for the Global Community of Geoscientists

In celebration of Earth Day, Franz Inc., an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology for Knowledge Graphs, today recognized how AllegroGraph, its semantic knowledge graph technology, is playing an essential part in making data FAIR (Findable, Accessible, Interoperable and Reusable) for the geoscience community. Since the current understanding of earth science processes is largely based on earth observation and numerical model data, making this data FAIR for all geoscientists and technologists is critical to facilitate future knowledge discovery about planet Earth.

Collecting, storing, monitoring and analyzing data from the core of the Earth up to the atmosphere provides critical knowledge about the planet and how living things interact with it. Scientists and technologists gather information about Earth from a range of sources, including: satellites, air- ground- and ocean-based sensors, physical sample data, etc., which are all recorded at a variety of temporal and spatial resolutions and need to be represented on the web for the global

scientific community to access and use. AllegroGraph's unique semantic graph capabilities allow diverse and complex data sources to be easily integrated with full search and cross-dataset queries possible.

"Our most pressing global environmental challenges cannot be solved by a single organization," said Dr. Annie Burgess, Lab Director, Earth Science Information Partners (ESIP). "Scientists require data collected across multiple disciplines, which are often managed by many different agencies and institutions. ESIP is a community of data and information technology professionals dedicated to ensuring those data are FAIR. To assist with that goal, the unique semantic graph capabilities of AllegroGraph are leveraged with the ESIP Community Ontology Repository, a community platform to manage and exchange terms and vocabularies that assists scientists to publish, discover and reuse data."

"To address important marine research, there is a critical need for ocean observatories to share data in a way that is easy to discover, use and integrate," said Carlos Rueda, Senior Software Engineer, Monterey Bay Aquarium Research Institute. "With this goal in mind, the Marine Metadata Interoperability Project developed the MMI Ontology Registry and Repository

(ORR), which leverages AllegroGraph to provide powerful interoperable semantic services that make the content on the web interconnected in a meaningful way for both humans and machines to consume.”

“We are at an exciting stage where there is a critical mass of experts and organizations around the globe with similar goals as well as the realization that we need knowledge-intensive applications,” said Dr. Lewis McGibbney, Data Scientist, Jet Propulsion Laboratory, California Institute of Technology and Co-Chair of the NASA ESDSWG Search Relevance Working Group. “The semantic technology stack is a crucial piece for building intelligent apps for knowledge-intensive use cases within the geoscience area.”

“Semantic graph technology is particularly well-suited to address the complex data integration, data access and analysis challenges surrounding Earth data science,” said Dr. Jans Aasman, CEO of Franz Inc. “We are thrilled that leading geoscience organizations are tapping into the power of AllegroGraph to share Earth science ontologies and data. We look forward to continuing to work with the community and help forward their important projects.”

A recent Gartner report explains the importance of using semantic technology to drive value out of data and included AllegroGraph as a graph database to consider for semantic technology solutions.

“Unprecedented levels of data scale and distribution are

making it almost impossible for organizations to effectively exploit their data assets. Data and analytics leaders must adopt a semantic approach to their enterprise data assets or face losing the battle for competitive advantage.” (Source: *Gartner, How to Use Semantics to Drive the Business Value of Your Data*, Guido De Simoni, November 27, 2018.) To view a summary of the report, go to <https://www.gartner.com/doc/3894095/use-semantics-drive-business-value>.

About ESIP

The Earth Science Information Partners (ESIP) is a community of innovative science, data and information technology practitioners. ESIP members catalyze connections across traditional institutional and domain boundaries to solve critical Earth science data stewardship, information technology and interoperability issues. Through this work, ESIP improves Earth science data management practices and makes Earth science data more discoverable, accessible and useful to researchers, policy makers and the public. Learn more at esipfed.org or follow @ESIPfed on Twitter.

About Monterey

Bay Aquarium Research Institute

Monterey Bay Aquarium Research Institute (MBARI) encompass the entire ocean, from the surface waters to the deep seafloor, and from the coastal zone to the open sea. The need to understand the ocean in all its complexity

and variability drives MBARI's research and development efforts.

About JPL

The Jet Propulsion Laboratory is a unique national research facility that carries out robotic space and Earth science missions. JPL helped open the Space Age by developing America's first Earth-orbiting science satellite, creating the first successful interplanetary spacecraft, and sending robotic missions to study all the planets in the solar system as well as asteroids, comets and Earth's moon. In addition to its missions, JPL developed and manages NASA's Deep Space Network, a worldwide system of antennas that communicates with interplanetary spacecraft. JPL is a federally funded research and development center managed for NASA by Caltech. From the long history of leaders drawn from the university's faculty to joint programs and appointments, JPL's intellectual environment and identity are profoundly shaped by its role as part of Caltech.

About AllegroGraph

AllegroGraph is a database technology that enables businesses to extract sophisticated decision insights and predictive analytics from highly complex, distributed data that cannot be uncovered with conventional databases. Unlike traditional relational databases or other NoSQL databases, AllegroGraph

employs semantic graph technologies that process data with contextual and conceptual intelligence. AllegroGraph is able run queries of unprecedented complexity to support predictive analytics that help organizations make more informed, real-time decisions. AllegroGraph is utilized by dozens of the top F500 companies worldwide

Semantic Knowledge Graphs are the Foundation for Artificial Intelligence

The foundation for Knowledge Graphs and AI lies in the facets of semantic technology provided by AllegroGraph.

Semantic Graph databases provide the core technology environment to enrich and contextualized the understanding of data. The ability to rapidly integrate new knowledge is the crux of the Knowledge Graph and depends entirely on semantic technologies.

About Franz Inc.

Franz Inc. is an early innovator in Artificial Intelligence (AI) and leading supplier of Semantic Graph Database technology with expert knowledge in developing and deploying Knowledge Graph solutions.

The foundation for Knowledge Graphs and AI lies in the facets of semantic technology provided by AllegroGraph and Allegro CL. The ability to rapidly integrate new knowledge is the crux of the Knowledge Graph and Franz Inc. provides the key technologies and services to address

your complex challenges. Franz Inc. is your Knowledge Graph technology partner.

All trademarks and registered trademarks in this document are the properties of their respective owners.

Webcast – Speech Recognition, Knowledge Graphs, and AI for Intelligent Customer Operations – April 3, 2019

Presenters – Burt Smith, N3 Results and Jans Aasman, Franz Inc.

In the typical sales organization the contents of the actual chat or voice conversation between agent and customer is a black hole. In the modern Intelligent Customer Operations center (e.g. N3 Results – www.n3results.com) the interactions between agent and customer are a source of rich information that helps agents to improve the quality of the interaction in real time, creates more sales, and provides far better analytics for management.

Join us for this Webinar where we describe a real world Intelligent Customer Operations center that uses graph based technology for taxonomy driven entity extraction, speech recognition, machine learning and predictive analytics to improve quality of conversations, increase sales and improve business visibility.

View the recorded webinar.

Why Is JSON-LD Important To Businesses?

Forbes – February 2019

Although you may not have heard of JavaScript Object Notation Linked Data (JSON-LD), it is already affecting your business. Search engine giant Google has mentioned JSON-LD as a preferred means of adding structured data to webpages to make them considerably easier to parse for more accurate search engine results. The Google use case is indicative of the larger capacity for JSON-LD to increase web traffic for sites and better guide users to the results they want.



Expectations are high for JSON-LD, and with good reason. It effectively delivers the many benefits of JSON, a lightweight data interchange format, into the linked data world. Linked data is the technological approach supporting the World Wide Web and one of the most effective means of sharing data ever devised.

In addition, the growing number of enterprise knowledge graphs fully exploit the potential of JSON-LD as it enables organizations to readily access data stored in document formats and a variety of semi-structured and unstructured data as well. By using this technology to link internal and

external data, knowledge graphs exemplify the linked data approach underpinning the growing adoption of JSON-LD – and the demonstrable, recurring business value that linked data consistently provides.

Read the full article at [Forbes](#).

Semantic Web and Semantic Technology Trends in 2019

Dataversity – January 2019

What to expect of Semantic Web and other Semantic Technologies in 2019? Quite a bit. DATAVERSITY engaged with leaders in the space to get their thoughts on how Semantic Technologies will have an impact on multiple areas.

Dr. Jans Aasman, CEO of Franz Inc. was quoted several times in the article:

Among the semantic-driven AI ventures next year will be those that relate to the healthcare space, says Dr. Jans Aasman, CEO of Semantic Web technology company Franz, Inc:

“In the last two years some of the technologies were starting to get used in production,” he says. “In 2019 we will see a ramp-up of the number of AI applications that will help save lives by providing early warning signs for impending diseases. Some diseases will be predicted years in advance by using genetic patient data to understand future biological issues, like the likelihood of cancerous mutations – and start preventive therapies before the disease takes hold.”

If that's not enough, how about digital immortality via AI Knowledge Graphs, where an interactive voice system will bring public figures in contact with anyone in the real world? "We'll see the first examples of Digital Immortality in 2019 in the form of AI Digital Personas for public figures," says Aasman, whose company is a partner in the Noam Chomsky Knowledge Graph:

"The combination of Artificial Intelligence and Semantic Knowledge Graphs will be used to transform the works of scientists, technologists, politicians, and scholars like Noam Chomsky into an interactive response system that uses the person's actual voice to answer questions," he comments.

"AI Digital Personas will dynamically link information from various sources – such as books, research papers, notes and media interviews – and turn the disparate information into a knowledge system that people can interact with digitally." These AI Digital Personas could also be used while the person is still alive to broaden the accessibility of their expertise.

On the point of the future of graph visualization apps, Aasman notes that:

"Most graph visualization applications show network diagrams in only two dimensions, but it is unnatural to manipulate graphs on a flat computer screen in 2D. Modern R virtual reality will add at least two dimensions to graph visualization, which will create a more natural way to manipulate complex graphs by incorporating more depth and temporal unfolding to understand information within a time perspective."

Read the full article at [Dataversity](#).

2019 Trends in Data Governance: The Model Governance Question

From an AI Business Article by Jelani Harper – November 2018

The propagation of the enterprise's ability to capitalize on data-driven processes—to effectively reap data's yield as an organizational asset, much like any other—hinges on data governance, which arguably underpins the foundation of data management itself.

There are numerous trends impacting that foundation, many of which have always had, and will continue to have, relevance as 2019 looms. Questions of regulatory compliance, data lineage, metadata management, and even data governance will all play crucial roles.

Franz's CEO, Dr. Jans Aasman was quoted:

Still, as Aasman denoted, "It's extremely complicated to make fair [machine learning] models with all the context around them." Both rules and human supervision of models can furnish a fair amount of context for them, serving as starting points for their consistent governance.

Read the full article at [AI Business](#).

Navigating time in knowledge graphs

Franz's CEO, Jans Aasman, recently wrote the following article for InfoWorld.



The temporal benefits of cognitive knowledge graphs can affect almost any business problem, including basic issues of data management such as data quality, data cleansing, and integration

The concept of time presents several distinct challenges for data management, particularly as it applies to databases or stores. Those difficulties are related to the nature of time, which is ongoing, and its expressions in repositories. The former means data are relevant both at state (a point in time) and over periods of time, which increases the complexity.

Read the Full Article

The Cornerstone of Data Science: Progressive Data

Modeling

From AI Business June 27, 2018

This article covers Single Schema, Universal Taxonomies, Time Series Analysis, Accelerating Data Science and features some thought leadership from Franz Inc.'s CEO, Jans Aasman:

'Contemporary data science and artificial intelligence requirements simply can't wait for this ongoing, dilatory process. According to Jans Aasman, CEO of Franz, they no longer have to. By deploying what Aasman called an "events-based approach to schema", companies can model datasets with any number of differences alongside one another for expedited enterprise value.'

'The resulting schema is simplified, uniform, and useful in multiple ways. "You achieve two goals," Aasman noted. "One is you define what data you trust to be in the main repository to have all the truth. The second thing is you make your data management a little more uniform. By doing those two things your AI and your data science will become better, because the data that goes into them is better."'

Dr. Aasman goes on to note:

'The events-based schema methodology only works with enterprise taxonomies—or at least with taxonomies spanning the different sources included in a specific repository, such as a Master Data Management hub. Taxonomies are necessary so that "the type of event can be specified," Aasman said.'

'Moreover, taxonomies are indispensable for clarifying terms and their meaning across different data formats, which may represent similar concepts in distinct ways. Therefore, practically all objects in a database should be "taxonomy based" Aasman said, because these hierarchical

classifications enable organizations to query their repositories via this uniform schema.'

Read the full article over at AI Business.



How AI Boosts Human Expertise at Wolters Kluwer



Wolters Kluwer, a long time AllegroGraph customer, recently spoke with Alex Woodie at Datanami to describe how they are using AI tools such as AllegroGraph:

Thousands of companies around the world rely on Wolters Kluwer's practice management software to automate core aspects of their businesses. That includes doctor's offices that use its software make healthcare decisions in a clinical setting, corporate law offices that use its software to understand M&A activities, and accounting firms that use its software to craft tax strategies for high net-worth clients.

Wolters Kluwer is embedding a range of AI capabilities – including deep learning and graph analytics – across multiple product lines. For example, its Legalview Bill Analyzer

software helps to identify errors in legal bills sent from outside law firms to the corporate counsels of large companies. The typical recovery rate for people reviewing bills manually is 1% to 2%. By adding machine learning technology to the product the recovery rate jumps to 7% to 8%, which can translate into tens of millions of dollars.

Wolters Kluwer is using graph analytic techniques to accelerate the knowledge discovery process for its clients across various professions. **The company has tapped Franz's AllegroGraph software** to help it drive new navigational tools for helping customers find answers to their questions.

By arranging known facts and concepts as **triples in the AllegroGraph database** and then exposing those structures to users through a traditional search engine dialog box, Wolters Kluwer is able to surface related insights in a much more interactive manner.

"We're providing this live feedback. As you're typing, we're providing question and suggestions for you live," Tatham said. **"AllegroGraph gives us a performant way** to be able to just work our way through the whole knowledge model and come up with suggestion to the user in real time."

Read the full article over at Datanami.