RRDF gotcha

mja@statgroup.dk

2016-02-17

Contents

Setup

First load the package.

```
library(rrdf)
library(rrdfancillary)
```

Example on what not to do

Here is how I understood what happens when the exactly same triples are inserted - only one triple is stored.

```
store3<- new.rdf(ontology=FALSE)

sparql.rdf( store3, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }
```

```
## <0 x 0 matrix>
```

```
SPARQLinsert<- '
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
INSERT DATA
{
  <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
  <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
}
'

cat(SPARQLinsert,"\n")
```

```
##
## PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
## INSERT DATA
## {
##   <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
##   <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
## }
##
```

```
update.rdf( store3, SPARQLinsert )
```

```
## [1] TRUE
```

```
sparql.rdf( store3, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }
```

```
##      s                              p                               o
## [1,] "http://example.org/subject1" "http://example.org/property1" "mytext"
##      lang datatype
## [1,] ""   "http://www.w3.org/2001/XMLSchema#string"
```

Now, of course, it is always better only to store the values once, when it is intended to store one copy. However, as I thougth that only one trippel is stored, I was less carefull in some of the code.

Here is what Apache/Jena does when using the RRDF interface.

```
store<- new.rdf(ontology=FALSE)
add.data.triple( store, subject="http://example.org/subject1", predicate="http://example.org/property1"
add.data.triple( store, subject="http://example.org/subject1", predicate="http://example.org/property1"
sparql.rdf( store, "select ?s ?p ?o where {?s ?p ?o}"  )
```

```
##      s                              p                               o
## [1,] "http://example.org/subject1" "http://example.org/property1" "mytext"
## [2,] "http://example.org/subject1" "http://example.org/property1" "mytext"
```

```
sparql.rdf( store, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }"
```

```
##      s                              p                               o
## [1,] "http://example.org/subject1" "http://example.org/property1" "mytext"
## [2,] "http://example.org/subject1" "http://example.org/property1" "mytext"
##      lang datatype
## [1,] ""   "http://www.w3.org/2001/XMLSchema#string"
## [2,] "en" "http://www.w3.org/1999/02/22-rdf-syntax-ns#langString"
```

The same tripel appears twice!

Mixing `INSERT DATA` and RRDF add.data.triple gives same result - two triples.

```
store4<- new.rdf(ontology=FALSE)

sparql.rdf( store4, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }
```

```
## <0 x 0 matrix>
```

```
SPARQLinsert<- '
PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
INSERT DATA
{
  <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
}
'

cat(SPARQLinsert,"\n")
```

```
##
## PREFIX xsd:  <http://www.w3.org/2001/XMLSchema#>
## INSERT DATA
## {
##   <http://example.org/subject1>  <http://example.org/property1> "mytext"^^xsd:string .
## }
##
```

```
update.rdf( store4, SPARQLinsert )
```

```
## [1] TRUE
```

```
sparql.rdf( store4, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }
```

```
##      s                               p                                o
## [1,] "http://example.org/subject1" "http://example.org/property1" "mytext"
##      lang datatype
## [1,] ""   "http://www.w3.org/2001/XMLSchema#string"
```

```
add.data.triple( store3, subject="http://example.org/subject1", predicate="http://example.org/property1

sparql.rdf( store4, "select ?s ?p ?o (lang(?o) as ?lang) (datatype(?o) as ?datatype) where {?s ?p ?o  }
```

```
##      s                               p                                o
## [1,] "http://example.org/subject1" "http://example.org/property1" "mytext"
##      lang datatype
## [1,] ""   "http://www.w3.org/2001/XMLSchema#string"
```