EQL BOOLEAN QUERIES

This document describes the EQL boolean queries syntax used in QuOntoEQL system.

It's possible to express EQL boolean queries handling two levels:

1) The 1st level is the principal query (in SQL)

2) The 2nd level is the SPARQL query in FROM clause of the EQL query (i.e. the SparqlBooleanTable).

In the 1st level it is used the following SQL STANDARD syntax:

SELECT CASE WHEN <CONDITION> **THEN** true **ELSE** false **END**;
(Where the <CONDITION> is for example: exists (<Query EQL>))

In the 2nd level it is used the following SPARQL syntax for SparqlBooleanTable:

ASK {<WHERE CLAUSE IN SPARQL>}.
(Where the <WHERE CLAUSE IN SPARQL> is a SPARQL basic graph pattern)

EXAMPLES:

1. Is there a not full professor that comes from Roma?

```
SELECT CASE WHEN exists(
                select professors.x
                from SparqlTable(
                        select ?x ?y
                        where {?x rdf:type 'Professor'.
                                ?x address ?y }) professors
                where professors.y like '%Roma%' and
                        professors.x not in (
                                select fullP.x
                                from SparqlTable(
                                        select  ?x
                                        where {?x rdf:type 'FullProfessor'}) fullP))
THEN true ELSE false END;
```

2. Is there a full professor or an associate professor that comes from Avezzano?

```
SELECT CASE WHEN exists(
                select prof.x
                from SparqlTable( select?x
                        where{?x rdf:type 'FullProfessor'.
                                ?x address ' Avezzano'}) prof
                union
                select prof.x
```

```
            from SparqlTable( select?x
                                where{?x rdf:type 'AssociateProfessor'.
                                      ?x address ' Avezzano'}) prof;)
THEN true ELSE false END;
```

3. I want to know all the publications made on 10-10-2007 and if this day Luca and Andrea have made at least one publication:

```
SELECT pub.x
FROM SparqlTable(select ?x
          where {?x rdf:type 'Publication'.
                 ?x publicationDate '10-10-2007'}) pub,
       SparqlBooleanTable(ASK{{?x rdf:type 'Publication'.
                               ?x publicationAuthor 'Luca'.
                               ?x publicationDate '10-10-2007'}
                               union
                              {?y rdf:type 'Publication'.
                               ?y publicationAuthor 'Andrea'.
                               ?y publicationDate '10-10-2007'}}) pubauthor;
```

The SparqlBooleanTable is processed as the SparqlTables, but the query returned by UNFOLDIND is put in a SQL boolean query like the following:

```
(SELECT CASE WHEN exists (<Query returned by the UNFOLDING>) THEN true else false END as value);
```

So it's added the following condition:

```
studentworker.value = true
```

4. I want to know if all those who are graduated from Sapienza on 2007 are less than 30 years old:

```
SELECT CASE WHEN 30 > all (SELECT graduates.a
                  FROM SparqlTable (
                              SELECT ?a ?u
                              WHERE {(?x degreeFrom ?u) degreeYear '2007'.
                                     ?x age ?a}) graduates
                WHERE graduates.u like '%Sapienza%') THEN true ELSE false END;;
```

So we can put all type of conditions in EQL boolean query not only the conditions with "EXISTS" or "NOT EXISTS".

5. I want to know if there was a student named 'Luca' graduated on 2007 and if in this year there was a student named 'Mario' too:

```
SELECT CASE WHEN 'Luca' in (
          SELECT allstud.n
```

```
        FROM SparqlTable(SELECT ?n
                         WHERE{?x rdf:type 'Student'.
                               ?x name ?n.
                              (?x degreeFrom ?u) degreeYear '2007'}) allstud,
             SparqlBooleanTable(ASK {?x rdf:type 'Student'.
                                     ?x name 'Mario'.
                                    (?x degreeFrom ?u) degreeYear '2007'}) stud1)
THEN true ELSE false END;
```

Of course it isn't possible to do any join between SparqlTables and SparqlBooleanTables.