

1 Other queries

1.1 Ranking

Suppose I have the following triples:

```
:A :value 60 .
:B :value 23 .
:C :value 89 .
:D :value 34 .
```

and I want to obtain a ranking where C would be 1, A 2, D 3 and B 4.

The ranking can be obtained as:

```
SELECT ?x ?v ?ranking {
  ?x :value ?v .
  { SELECT (GROUP_CONCAT(?x;separator="") as ?ordered) {
    { SELECT ?x {
      ?x :value ?v .
    } ORDER BY DESC(?v)
    }
  }
}
BIND (str(?x) as ?xName)
BIND (strbefore(?ordered,?xName) as ?before)
BIND ((strlen(?before) / strlen(?xName)) + 1 as ?ranking)
} ORDER BY ?ranking
```

Another solution by Joshua Taylor would be:

```
SELECT ?x ?v (COUNT(*) as ?ranking) WHERE {
  ?x :value ?v .
  [] :value ?u .
  FILTER( ?v <= ?u )
}
GROUP BY ?x ?v
ORDER BY ?ranking
```

RDF Data Cube Profile

```
cex:cube a cex:ValidationProfile ;
  cex:name "RDF Data Cube" ;
  cex:ontologyBase cexOntology:cube.ttl ;
  cex:expandSteps
    ([ cex:name "Closure" ;
      cex:uri cubeURIEExpand:closure.ru
    ]
     [ cex:name "Flatten" ;
      cex:uri cubeURIEExpand:flatten.ru
    ]
    ) ;
  # Integrity queries from RDF Data Cube
  cex:integrityQuery
    [ cex:name "IC-1. Unique DataSet (Q1a)" ;
      cex:uri cubeURI:q1a-integrity.sparql
    ] ;
  cex:integrityQuery
    [ cex:name "IC-1. Unique DataSet (Q1b)" ;
```

```
    cex:uri cubeURI:qlb-integrity.sparql
  ] ;
  # Other integrity queries
.
```

Computex Profile

```
cex:computex a cex:ValidationProfile ;
cex:name "Computex" ;
cex:ontologyBase cexOntology:computex.ttl ;
cex:import profileURI:cubeProfile.ttl ;
cex:expandSteps
(
  [ cex:name "Copy Raw" ;
    cex:uri cexURIEExpand:q4-copyRaw.sparql
  ]
  # Other UPDATE queries
) ;
cex:integrityQuery
[ cex:name "Adjusted computation must have a value" ;
  cex:uri cexURI:ql-adjustedValueExistsObs.sparql
]
# Other integrity queries
.
```