SPARQL Queries for Publications and Authors

Publications

List publications with common citation information

The query returns a list of all publications along with common citation information. The publication must have a venue type of bibo:Journal, and the journal must have a label and an issn, otherwise the publication will not be included in the results. The publication may be of any type (AcademicArticle, Editorial, etc). The most specific type of the article is identified and its label is included in the results. The publication must have a date. The three lines get the VIVO date time value, and its date string, and then parses the date string to return the first four characters of the date string, which is included in the results as year. Five additional attributes are considered optional. Publications will be included in the results regardless of whether these attributes (doi, start page, end page, volume, and issue) have values.

List publications with common citation information

```
SELECT ?doi ?pub_label ?type_label ?venue_label ?issn ?issue ?startPage ?endPage ?volume ?year
WHERE{
    ?pub vivo:hasPublicationVenue ?venue .

    ?venue rdf:type bibo:Journal .
    ?venue bibo:issn ?issn .
    ?venue rdfs:label ?venue_label .

    ?pub rdfs:label $pub_label .
    ?pub vitro:mostSpecificType ?type .
    ?type rdfs:label ?type_label .

    ?pub vivo:dateTimeValue ?dtv .
    ?dtv vivo:dateTime ?dt .
    BIND(SUBSTR(str(?dt),1,4) AS ?year)

    OPTIONAL { ?pub bibo:doi ?doi . }
    OPTIONAL { ?pub bibo:pageStart ?startPage . }
    OPTIONAL { ?pub bibo:pageEnd ?endPage . }
    OPTIONAL { ?pub bibo:volume ?volume . }
    OPTIONAL { ?pub bibo:issue ?issue . }
}
```

Count Publications by Year

The query creates a year variable for each publication and uses it to count publications by year. It uses the same selection criteria as the query above.

Count publications by year

```
SELECT ?year (COUNT(DISTINCT ?pub) AS ?count)
WHERE{
    ?pub vivo:hasPublicationVenue ?venue .

    ?venue rdf:type bibo:Journal .
    ?venue bibo:issn ?issn .
    ?venue rdfs:label ?venue_label .

    ?pub rdfs:label $pub_label .

    ?pub vivo:dateTimeValue ?dtv .
    ?dtv vivo:dateTime ?dt .
    BIND(SUBSTR(str(?dt),1,4) AS ?year)
}
GROUP BY ?year
ORDER BY DESC(?year)
```

Authors

List authors by number of publications

The WHERE clause begins by finding people who have one or more authorships. Each person in the result must have an rdfs:label. The query then groups by author. A COUNT aggregation is used to get the publication count. The results are ordered by descending publication count – the authors with the highest number of publications will be listed first. Some people have more than one rdfs:label – they may have name variations and/or they may have names with or without various language tags. An aggregation on the author label returns a single author name. In this query we use STR to remove any language tags, and then use MIN to get the first name variant in alphabetical order. If the person has just one rdfs:label, that label is returned as the name.

Note: When GROUP BY is used, aggregations must be used in the SELECT statement to indicate what will be done with multiple values of each attribute to appear in the result.

Note: The use of the COUNT aggregation with a GROUP BY is a standard SPARQL pattern for generating a frequency table. In this query we are creating a frequency table of publications by author.

List authors by number of publications

```
SELECT (MIN(STR(?author_label)) AS ?author_name) (COUNT(DISTINCT ?authorship) AS ?pub_count)
WHERE {
    ?author vivo:relatedBy ?authorship .
    ?author a foaf:Person .
    ?authorship a vivo:Authorship .

    ?author rdfs:label ?author_label .
}
GROUP BY ?author
ORDER BY DESC(?pub_count)
```

People with less than five publications

In the query below we select people that have at least one publication, but less than five. VIVO uses "relatedBy" to associate people with authorships that indicate that the person is an author of the paper. Note that the query insures that ?person is a foaf:Person and ?a is a vivo:Authorship. relatedBy is used in other contexts so it is very important that we clarify which relatedBy assertions we are interested in.

People with less than five publications

```
SELECT ?person
WHERE
{
    ?person vivo:relatedBy ?a .
    ?person a foaf:Person .
    ?a a vivo:Authorship .
}
GROUP BY ?person
HAVING (COUNT(?a) < 5)
```