Motivating Use Case

The education commissioner in Wales is interested in finding out the effect of poverty on student performance. If there is a significant overlap between children growing up in poor households and those that perform badly on tests, the government would consider rethinking their approach to education reform.

Nathan, a K-12 education analyst in the commissioner's office, has been assigned the task of finding this information. To find any correlation, he would compare the proportion of students from low income households among the failing students.

He would perform this task by submitting queries to a secure SPARQL federation engine. Various SPARQL endpoints have registered with this engine. Among them are two that has data held by two UK government agencies - the Department of Work and Pensions (DoWP), and the Department for Education (DoE) – that are of interest to Nathan. The DoWP administers the welfare program and has aggregated financial information of those that live close to and below the poverty line. DoE keeps a record of test scores for all students attending government run schools in the UK, which goes back several years.

The DoE allows access to test results data to its employees, who are at analyst grade or above. Because DoE has administrative oversight of the Welsh commissioner's office, Nathan is allowed access to the DoE endpoint; but only to data that pertains to students in Wales. Similarly to the DoE, the DoWP permits access to its endpoint to its own employees based on job title. It also permits access to outsiders who have been cleared to do so through an approval process. However, as part of the UK government's Transparency Programme, the UK government has made many government datasets publicly available in RDF. Nathan is able to access this information without any out of band agreements.

First, Nathan logs into the federation engine with credentials that attest that he is an analyst. He submits a query to the federation engine to find all students in Wales who are not classified as proficient.

```
Prefix DoE: <http://data.gov.uk/doe>
SELECT (COUNT(DISTINCT(?student)))
WHERE { ?student DoE:country        "Wales" .
                 DoE:classification "not_proficient"
      }
```

The result here gives him the total number of students who are not proficient. Now he submits another query to find the number of students in this larger who are from poor households.

```
Prefix Gov:   <http://data.gov.uk>
Prefix DoE:   <http://data.gov.uk/doe>
Prefix DoWP:  <http://data.gov.uk/dowp>
SELECT (COUNT(DISTINCT(?student)))
WHERE { ?student   DoE:country          "Wales"  .
                   DoE:classification   "not_proficient"  .
                   Gov:address           ?address
        ?address DoWP:designation     "below_poverty_level"
      }
```

The federation engine splits this query into two sub-queries that will be sent to the DoE and DoWP endpoints, respectively.

```
Prefix Gov:     <http://data.gov.uk>
Prefix DoE:     <http://data.gov.uk/doe>
SELECT  ?student
WHERE { ?student   DoE:country            “Wales”  .
                   DoE:classification   “not_proficient”  .
                   Gov:address            ?address
        }
```

```
Prefix Gov:     <http://data.gov.uk>
Prefix DoWP:  <http://data.gov.uk/dowp>
SELECT   ?address
WHERE { ?person   Gov:address           ?address   .
        ?address  DoWP:designation    “below_poverty_level”  .
                   Gov:address            ?address
```

Based on the results of the two queries, Nathan is able to find the information he is looking for.

Furthermore, Nathan may be interested in finding out the effect of poverty on student performance over time. To do this, he might compare the percentages of students from poor households in the failing population at different grade levels. If the percentage increases over time, this would mean that poverty has a cumulative effect. The above queries can be modified as follows to get this information.

```
Prefix Gov:     <http://data.gov.uk>
Prefix DoE:     <http://data.gov.uk/doe>
Prefix DoWP:  <http://data.gov.uk/dowp>
SELECT (COUNT(DISTINCT(?student)))
WHERE { ?student   DoE:country            “Wales”  .
                   DoE:school_type      “school_type”
                   DoE:classification   “not_proficient”  .
                   Gov:address            ?address
         ?address DoWP:designation     “below_poverty_level”
        }
```

OR

```
Prefix Gov:     <http://data.gov.uk>
Prefix DoE:     <http://data.gov.uk/doe>
Prefix DoWP:  <http://data.gov.uk/dowp>
SELECT (COUNT(DISTINCT(?student)))
WHERE { ?student   DoE:country            “Wales”  .
                   DoE:in_grade           “grade”
                   DoE:classification   “not_proficient”  .
                   Gov:address            ?address
         ?address DoWP:designation     “below_poverty_level”
        }
```

where school_type can take values {elementary, middle, high} and grade can take values {K, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12}. The query by grade gives a more granular result.