Ireland Primary Schools

GROUP 7

APPROACH:

Our project aimed to make relationship between Ireland primary school information and county geographic information. From GeoHive a useful property is county's boundary represented in WKT, and from Ireland primary school there are latitude and longitude of schools. Somehow by connecting these two datasets by these properties, and making use of WKT, we made a new ontology as well as merged two datasets for creating new individuals.

Sources:

1. GeoHive County 100m: http://data.geohive.ie/downloadAndQuery.html
2. Ireland Primary Schools: https://data.gov.ie/dataset/primary-schools

Assumptions:

The GeoHive county data only have 26 counties which is less than the real county count that Ireland Has, for which we didn't know the reason. We have to adopt the 26-county dataset.
We assume that county's boundaries are correctly defined in WKT, so we can use standard geometry operations such as intersects, contains to the WKT literals.
We assume that primary school can only locate in one county, otherwise there will be some conditions that is very annoying and sparqls will not be right sometimes.

We assume that primary school with 0 boy is female school, with 0 girl is male school, with at least 1 girl and 1 boy is mixed school.
We assume that Roll_No from the school CSV is the unique id of school.
We assume that there is only 3 ethos for school according to the CSV.

Uplift:

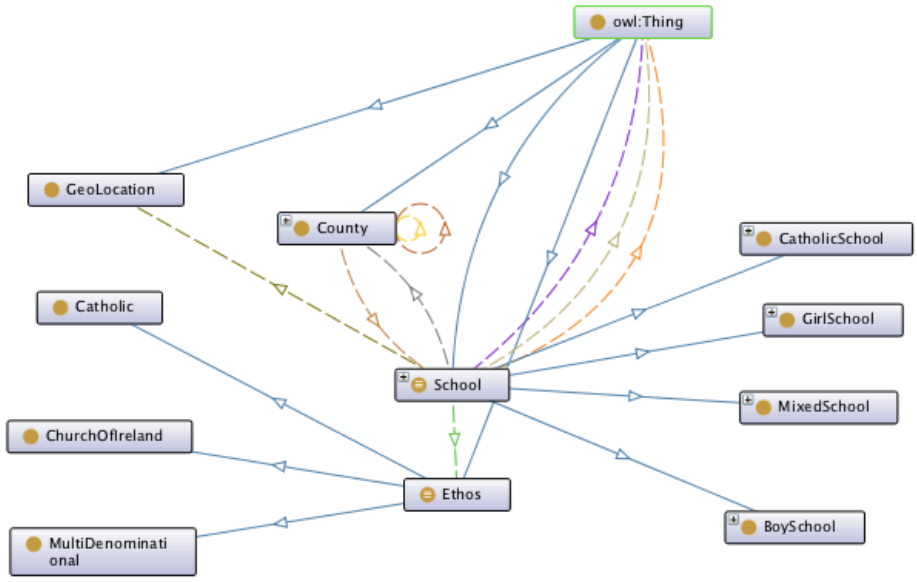
We believe that code is a better tool, so we are writing code to do basically everything.
For school, we add Jena CSV library and use functionality inside to read and parse csv file.
For county, we use Eris geometry library to parse WKT and calculate area and the relationships such as contains(Point), intersects(the other geometry).
Finally, the tool datasets are merged by code using Jena Ontology related APIs.

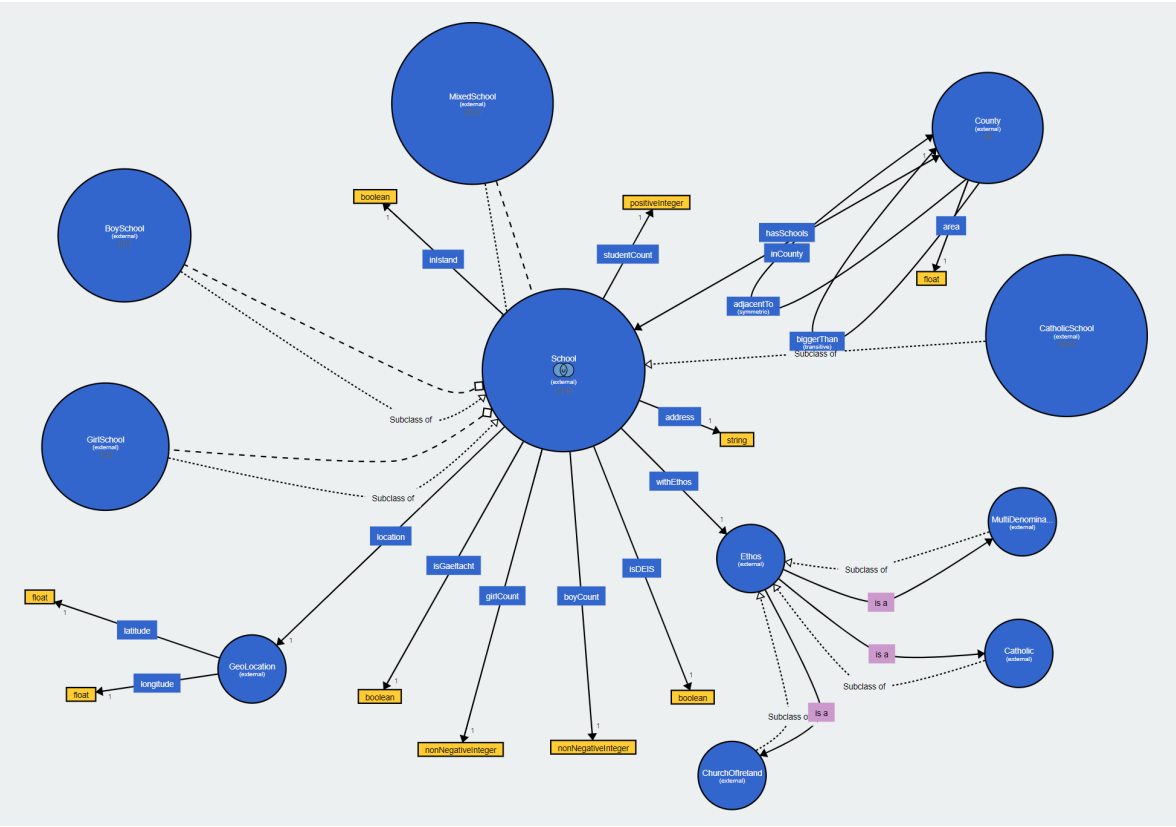
DESIGN OVERVIEW:

Classes & Properties:

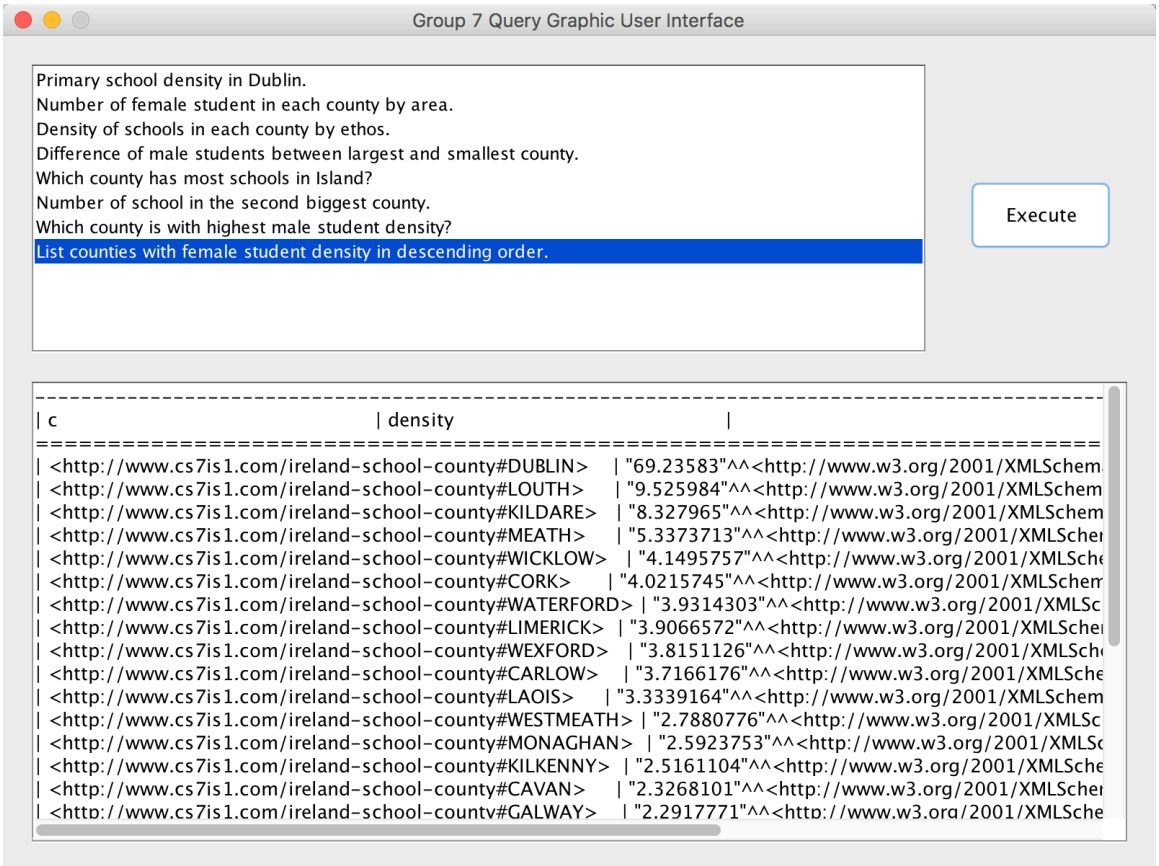
Two major classes are County and School, with other data or object properties associated with the major classes. Some special properties include:

biggerThan: transitive
hasSchools vs inCounty: inverseOf
adjacentTo: Symmetric
withEthos: Ethos is an enumerated class





Query Interface:



The interface is a simple Java Swing based GUI. It lists down the question which a user can choose to execute. The user can select a question from the list and press the Execute button. The results will be displayed in the bottom panel of the window in tabular format.

Queries:

Queries and SPARQLs are listed in the Appendix 1.

Challenges:

Geometry Operation:

During the operating of geometry operations, we found that the result of adjacentTo is not the same as the real Ireland condition. Thus we found that, the WKT literal actually was in 100m resolution and the relationships between county were not well represented in WKT. Some adjacent relationships were not included by perform intersects. We thought that the publisher of the GeoHive dataset should be carefully dealing with the geographic relationship between counties.

LODE Annotation:

For LODE compatible annotations, we tried to contain as much needed information as possible. But somehow some information just did not show. For example, we attached the description property and even add a picture showing the classes hierarchy in the ontology, but there was still nothing in the generated LODE website. But we checked other samples, there worked. We made a lot effort and at least important information like version, creator was successfully showed.

Protégé Weakness:

When we tried to use protégé to open the final ttl with individuals, we found that protégé was very inefficient, it froze all the time, and took a long time to open. Actually our ttl file was only 2.5 mb. But with jena, our application runs very fast, the query typically takes less than 1 second to execute.

Self-Reflections:

LINGHAO MA
I am the Java code writer, and major report writer.
I led the creation of ontology.
I conduct the uplift and mergence, as well as individual creation.

I am always an advocator of automation. Therefore I insisted to use code to do pretty much everything. I think the result is good, only execute java code again we can do the whole thing from scratch. It also makes it easier to change things and incrementally add new features.

ABHIMANYU M. HAZARIKA

Created several SPARQL queries, contributed in initial dataset selection & mapping and part of report writing.

SIMON QUIGLEY

Created the initial query GUI, and created several SPARQL queries making use of both datasets

DHRUV KABRA

Implemented several queries in SPARQL, along with class hierarchy diagram and helped in Editing of Report.

VINAY CHANDRAGIRI

Formulated queries combining both datasets and help a bit in choosing the apt dataset, SPARQL queries and also formulating ontology.

RADHE SHYAM YADAV

Implementation, Creation of SPARQL queries, deciding the final property diagram and supporting the team Members.

ASHISH LOCHAN

Contributed in selecting the final dataset, Ontology ideation, ontology reasoning using hermit, Questions formulation and creating the final report. we could have used several other datasets as well which would have more relevantly contributed to the ontology.

Conclusions:

The project presents an interface with scope of querying the ontology with provided SPARQL queries. It explored the whole process of ontology creation including dataset searching and analysis, uplift, ontology design, dataset merging, interface creation and SPARQL querying, etc.
The ontology model is concise and good. The pity is that the dataset is not very accuracy, for example, the WKT. If we have other options, we can actually find better dataset, or we can fix the issue of current dataset.
For now, the queries cover static queries. We can work more on the interface and SPARQL queries to have flexible queries and get dynamic answers.

Appendix 1: Questions and SPARQLS

Common prefix:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX owl: <http://www.w3.org/2002/07/owl#>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX base: <http://www.cs7is1.com/ireland-school-county#>
```

Primary school density in Dublin.

```
SELECT ((COUNT(?school))/?a AS ?density)
WHERE {
 ?school base:inCounty base:DUBLIN.
 base:DUBLIN base:area ?a
}
GROUP BY ?a
```

Number of female student in each county by area.

```
SELECT ?c (SUM (?s) AS ?girlCount) ?area
WHERE {
    ?school base:inCounty ?c.
    ?school base:girlCount ?s.
    ?c base:area ?area
}
GROUP BY ?c ?area
ORDER BY DESC (?area)
```

Density of schools in each county by ethos.

```
SELECT ?c (?e AS ?ethos) (COUNT(DISTINCT ?s) / ?a as ?density)
WHERE {
    ?s base:inCounty ?c.
    ?c base:area ?a.
    ?e rdfs:subClassOf base:Ethos.
    ?s base:withEthos ?ei.
    ?ei a ?e.
}
GROUP BY ?e ?c ?a
ORDER BY ?c
```

Difference of male students between largest and smallest county.

```
SELECT (?count1 - ?count2 AS ?difference) (?count1 AS ?max) (?count2 AS ?min)
WHERE{
{
    SELECT ?c1 (SUM(?b1) AS ?count1) ?area1
    WHERE {
        ?school base:inCounty ?c1.
        ?school base:boyCount ?b1.
        ?c1 base:area ?area1.
    }
    GROUP BY ?c1 ?area1
    ORDER BY DESC(?area1)
    LIMIT 1
}
{
```

```
   SELECT ?c2 (SUM(?b2) AS ?count2) ?area2
    WHERE {
        ?school base:inCounty ?c2.
        ?school base:boyCount ?b2.
        ?c2 base:area ?area2.
    }
    GROUP BY ?c2 ?area2
    ORDER BY ?area2
    LIMIT 1
}
}
```

Which county has most schools in Island?

```
SELECT ?c (COUNT(?school) AS ?count)
WHERE {
    ?school base:inCounty ?c.
}
GROUP BY ?c
ORDER BY DESC (?count)
LIMIT 1
```

Number of school in the second biggest county.

```
SELECT ?c ?count
WHERE
{
    SELECT ?c (COUNT(?school) AS ?count) ?a
    WHERE {
     ?school base:inCounty ?c.
     ?c base:area ?a
    }
    GROUP BY ?c ?a
    ORDER BY DESC (?a)
    LIMIT 2
}
ORDER BY ?a
LIMIT 1
```

Which county is with highest male student density?

```
SELECT ?c ((SUM(?s))/?a AS ?density)
WHERE {
    ?school base:inCounty ?c.
    ?school base:boyCount ?s.
    ?c base:area ?a
}
GROUP BY ?c ?a
ORDER BY DESC (?density)
LIMIT 1
```

List counties with female student density in descending order.

```
SELECT ?c ((SUM(?s))/?a AS ?density)
WHERE {
    ?school base:inCounty ?c.
    ?school base:girlCount ?s.
    ?c base:area ?a
}
GROUP BY ?c ?a
ORDER BY DESC (?density)
```