Technical Report - Group K

David Kavanagh - 18327890
Tanmay Kaushik - 18308341
Andrew Mc Donald - 18318748
Isobel Mahon - 17331358

Approach to Ontology Modelling:

Description of Competency Questions that ontology answers:

To carry out this task we, as a group, came up with several questions that required at least two of the datasets to be queried to get the answer. As we had chosen five different datasets we also made a big effort not to rely on any one dataset too much and to get a good eclectic mix. The ten questions we came up with are as follows:

1. *How many new homes were approved for loans when the annual interest rate was over 4.5?*
2. *What was the average price for new properties in Dublin when the interest rate was less than 3.0?*
3. *When the average new property price in Dublin was between 200000 and 250000, how many new loans were approved?*
4. *For how many years between 1999 and 2015 were the property prices in Dublin above 200000 and there were more than 100000 loans approved?*
5. *In the year with the most national school pupils in Carlow, what was the consumer interest rate?*
6. *Where were property prices highest when the interest rates were highest?*
7. *What was the average price approved by national banks for second hand houses when national new property price was greater than 100000?*
8. *What was the average property price of new houses in Dublin in the year when loan approvals changed the least?*
9. *What were the average values of a loan approved by Dublin Banks for second hand homes in Dublin when the total value of loans approved for second hand homes for the year was over €10,000,000,000?*
10. *What were the average prices for new properties in Dublin in years when interest rates were higher than all-time average interest rates (avg of interest rates 2003-2022)?*

These questions all make use of more than one dataset to answer and so they are a good test for our ontology. What's more, is that these are questions that likely would get asked in a real-world scenario and this backs up the case for the usefulness of our ontology.

Description of datasets selected for application:

For our project, we chose to examine five datasets. These datasets, while all very relevant to each other and containing incredibly useful information, were quite concise in that they contained a very specialised set of fields. This was the reason why we elected to use as many as we did.

The first dataset lists the average price of all new houses built between 1969 to 2015. It also separates them into their respective regions. These regions are not uniform but they include the major cities/towns in Ireland; Dublin, Cork, Galway, Limerick, Waterford, "Other

Areas" and "National". While this type of classification makes sense for housing as the vast majority of houses are built in these areas, the lack of uniformity makes it difficult to link to other data. Hence why the uplifting of this dataset was so important.

The second dataset deals with loan approvals from 1999 to 2015. While it does only contain a small sample of years, the quality of the data for those years is acceptable. It contains the number of loans that were approved for new houses and second-hand houses as well as the total value of loans for each type of home. The total number of loans and the total value of loans accompany the previous entries so that this dataset provides complete documentation of the situation of loan approvals for this time period.

The third dataset contains the variable interest rate on new housing loans and the interest rate on outstanding customer loans. This dataset is totally different from the previous one in that it delivers a small volume of information for a large amount of time as not only does it cover from 2003 to this year, but it also provides the value for each month for those years.

The fourth dataset is a much more complete bank of information. For every year from 1995, it lists the name of all school programmes, one example being "Pupils with special needs in mainstream national schools", and the number of classes that pertain to this programme. Another column in the dataset then outlines the number of pupils in each programme and a final column shows which county the entry is referring to. Unfortunately, the designated regions for schools in Ireland are quite unique. There are, for example, four different designations for Dublin; Dublin City, South Dublin, Fingal and Dun Laoghaire-Rathdown. This means that the dataset must be processed before it can be used with the other datasets we have chosen.

The fifth and final dataset contains the average value of loans granted by various parties for all years from 1999 to 2015. The groups that have granted these loans are National Banks and Building Societies, National Local Authorities, Dublin Banks and Building Societies and finally Dublin Local Authorities. Once again, this dataset has its own means of distinctions and categorisations, namely National and Dublin. While it is desirable in dealing with loans and housing, this does not immediately mesh with the other datasets and this had to be addressed in the mappings.

Assumptions made:

Over the course of the project, we made the following assumptions:

1. We assume that the data collected and shared by https://data.gov.ie/ is comprehensive and accurate.
2. For one of the datasets we chose to uplift - the dataset about loan approvals - there were columns that contained data about the value of loans dispersed by different organisations. However, it was not clear in the specification document online, if the values were in Millions or some other denomination. Therefore, based on some further

research, we compared the value with other datasets and made an informed decision that the values were provided in Millions.

References to sources used/reused: e.g. SIOC, FOAF

[1] Boris Villazón Terrazas, 'R2RML: RDB to RDF Mapping Language Schema', *W3C*, Sep. 17, 2012. https://www.w3.org/ns/r2rml (accessed Nov. 20, 2022).

[2] 'XML Schema', Oct. 15, 2014. https://www.w3.org/2001/XMLSchema (accessed Nov. 20, 2022).

[3] 'About: http://dbpedia.org/ontology/'. https://dbpedia.org/ontology/ (accessed Nov. 20, 2022).

[4] 'About: year'. https://dbpedia.org/ontology/year (accessed Nov. 20, 2022).

[5] 'RDFWeb: xmlns.com'. http://xmlns.com/ (accessed Nov. 20, 2022).

Discussion of data mapping process:

Initially, we chose 4 datasets that we wanted to map and uplift for our project. After our presentation, we were advised to add more attributes and relationships to our data, so we added a new dataset that contained information about schools, courses, regions and a few other things.

Based on the 5 datasets, we created an ontology model and used our ontology model for data mapping and the datasets to produce attributes and triples. Each dataset had a different mapping assigned to it with common links between them such as the year attribute. Based on the mappings, we created property files for each mapping and uplifted them to produce RDF Output files. For the mapping process, we used a variety of ontologies like FOAF, XSD, DBPEDIA, etc.

The process of data mapping was an iterative one since we had to update and modify it regularly based on the requirements of queries, interface, dataset and a few other factors. When the data mapping was finalised, we used the R2RML program available to us to uplift the mapping.

Explanation of the use of inverse, symmetric & transitive properties

Inverse

As the ontology is modelled in a directed graph, inverse properties are used to describe relations in other directions than just domain to range. In our ontology, this was put into practice in the "IncludesCounty" property and the "BelongsToRegion" property where the county says which region it's a part of and the region says which counties constitute it.

Symmetric:

Symmetric properties mean that for any object that is connected to a subject by a given predicate, the object is also connected to the subject by that predicate in the opposite direction, eg. if A is friends with B, B must also be friends with A. It found its way into our ontology in the form of the "AffectedBy" property which says that the number of loan approvals affects the variable interest rate and the variable interest rate affects the number of loan approvals.

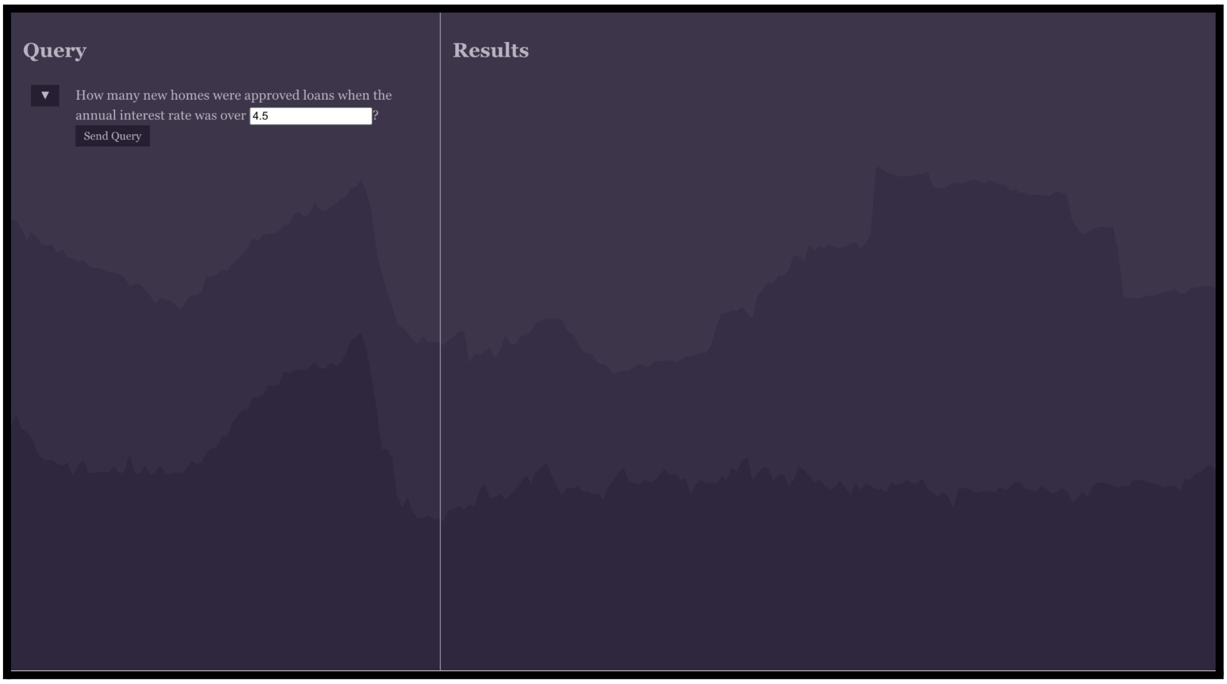
Transitive:

Transitive properties mean that if a subject-1 is connected by a predicate to object-1, and object-1 is connected as the subject by the same predicate to object-2, then subject-1 is connected by that predicate to object-2, eg. if X > Y and Y > Z then X > Z. The transitive property is seen in our ontology in the form of "ContainsRegion" property which states that any region contained by a larger region also contains the regions that the smaller region contains. Ireland contains Leinster and Leinster contains Dublin so Ireland contains Dublin.

Overview of Design:

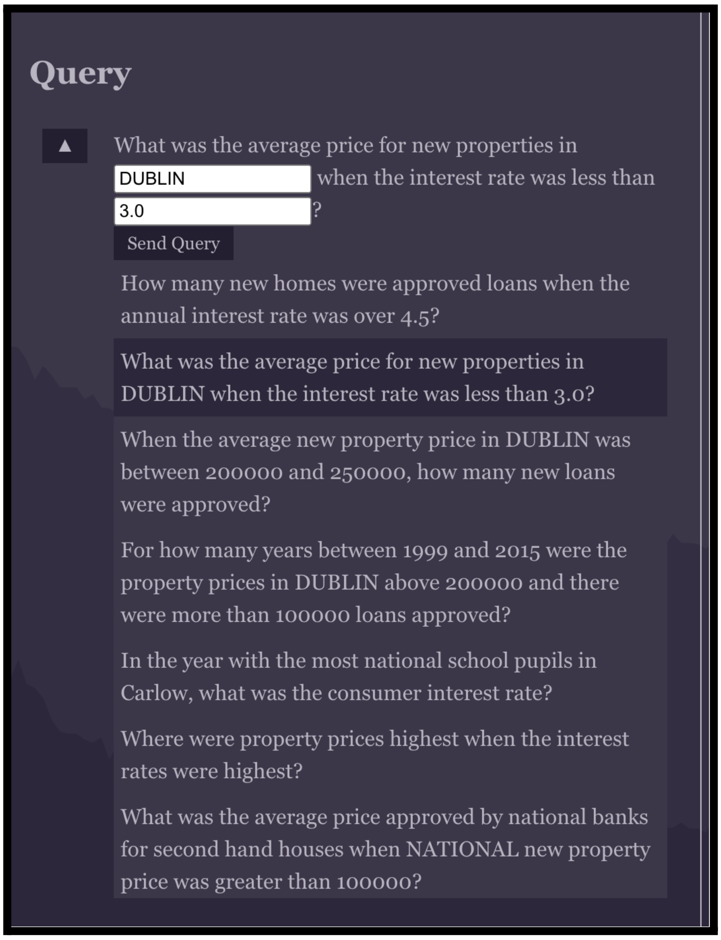
Description of Application Query Interface:

Our application query interface is a web app built using Javascript and React. It sends HTTP POST requests containing SPARQL queries to our database, hosted with GraphDB. The web app is shown below, as it appears when opened first:



How many new homes were approved loans when the annual interest rate was over 4.5?

Landing Page of the Web-Application



Query

What was the average price for new properties in DUBLIN when the interest rate was less than 3.0?

How many new homes were approved loans when the annual interest rate was over 4.5?

What was the average price for new properties in DUBLIN when the interest rate was less than 3.0?

When the average new property price in DUBLIN was between 200000 and 250000, how many new loans were approved?

For how many years between 1999 and 2015 were the property prices in DUBLIN above 200000 and there were more than 100000 loans approved?

In the year with the most national school pupils in Carlow, what was the consumer interest rate?

Where were property prices highest when the interest rates were highest?

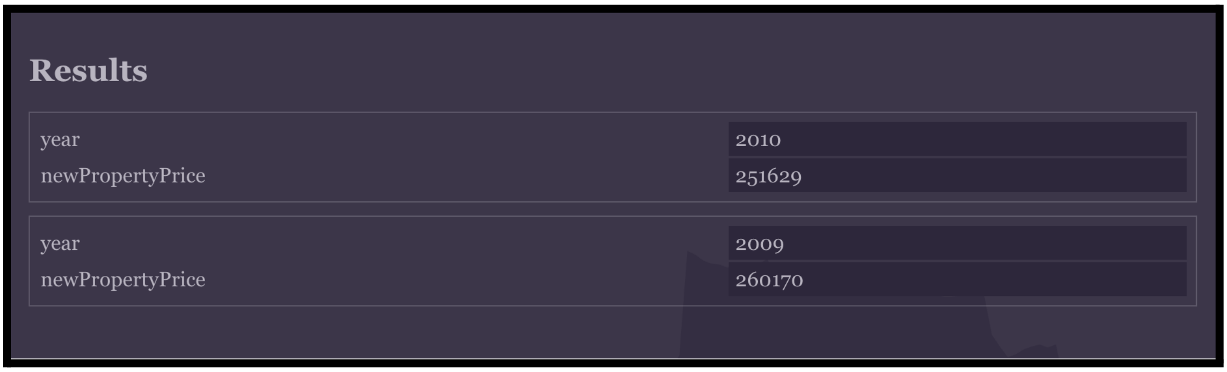
What was the average price approved by national banks for second hand houses when NATIONAL new property price was greater than 100000?

Drop-Down Menu enlisting all the queries

The query to be sent is chosen by a dropdown of questions. Each question has at least one editable field. The editable fields have two types: text and number. These are defined by the corresponding HTML form input types.

The send query button builds the query with the specified values and sends it to the database.

The results are then displayed on the web app. Both the dropdown and the results section are scrollable.



Results

year	2010
newPropertyPrice	251629

year	2009
newPropertyPrice	260170

Results Pane displaying the output of the queries

Description of Queries:

Our queries were written to answer the competency questions determined earlier in the project, and edited as needed to better match our ontologies and mappings. These are the questions as they ended up being answered:

1. How many new homes were approved for loans when the annual interest rate was over 4.5?
2. What was the average price for new properties in Dublin when the interest rate was less than 3.0?
3. When the average new property price in Dublin was between 200,000 and 250,000, how many new loans were approved?
4. For how many years between 1999 and 2015 were the property prices in Dublin above 200,000 and there were more than 100,000 loans approved?
5. In the year with the most national school pupils in Carlow, what was the consumer interest rate?
6. Where were property prices highest when the interest rates were highest?
7. What was the average price approved by national banks for second-hand houses when the national new property price was greater than 100,000?
8. What was the average property price of new houses in Dublin in the year when loan approvals changed the least?
9. What were the average values of a loan approved by Dublin Banks for second-hand homes in Dublin when the total value of loans approved for second-hand homes for the year was over €10,000,000,000?
10. What were the average prices for new properties in Dublin in years when interest rates were higher than all-time average interest rates (avg of interest rates 2003-2022)?

The regions and numbers in these questions are examples; all of the queries are editable in the UI, with the exception of Query 6.

Query 6 was one of the more complex query to design, as it contains two nested SELECTS. The innermost select gets the highest interest rate, the next SELECT finds the year that that interest rate took place and finds the highest property price value from that year, and the outermost SELECT finds and returns the region that had that property price.

Query 8 was also relatively difficult, as it relied on us finding the minimum change in loan approvals. We had to calculate the change in loan approval from the previous year for every year, get the minimum of those, and then calculate the changes again so we could find which object the minimum change belonged to.

Discussion of challenges faced while ontology modelling or creating queries and mappings:

Challenges faced while ontology modelling:
Creating the ontology was quite a unique process in comparison to other parts of this project. The main reason being the fact that there is no "right answer" to an ontology and that it must be adapted and constantly updated throughout the completion of the project.

The constant need to go back and update the ontology cost a lot of time. What's more is that a lot of the work done turned out to be redundant as a lot of it was later changed or altogether removed as designs evolved.

Challenges faced while creating queries:
While writing the queries we realised that some parts of the mapping and ontology were created in such a way that made writing queries quite difficult, so we had to continue updating them as we wrote the queries, which slowed us down a bit.

When creating the UI, we had some trouble getting the fetch queries to work. Our first issue was that the GraphDB CORS policy was disallowing us from receiving the query responses. This was solved by changing the GraphDB settings to enable CORS.

We were initially using GET requests, with the queries in the URL, which worked for some test queries, but stopped working when we started referencing particular predicates. We later realised this was because of an issue with the encoding of the "<" character, so we switched to using POST requests with the queries in the body, which fixed the problem.

We wanted to add a functionality that permits users to customise the queries. This was hard to design in the interface but we developed this feature to enhance the user experience.

Challenges faced while designing mappings:
The major challenge that we faced designing mappings was the numerous changes that had to be made regularly. We had to make various changes to the ontology during the project which resulted in updating and changing the mappings regularly. During the process of running the SPARQL queries, we discovered new connections and different ways of designing the mappings that will produce complex results easily. Additionally, effective communication about the new changes was a challenge as well who were working on different parts of the project.

How we organised our project:

1. We had in-person weekly meetings to divide work, set deadlines and discuss any issues that were encountered during the week.
2. We also organised online meetings if any member of the group wanted to discuss an important part of the project. We used Zoom for the online meetings.

3. Our team divided the project into small parts and assigned lead roles for each part to each group member.
    a. Andrew - Lead on Ontology Design
    b. David - Lead on Competency Questions design
    c. Isobel - Lead on UI and Query Design
    d. Tanmay - Lead on data selection, application scope and Lead on Mapping and Uplift
4. For documentation, we used Google Drive and Google Docs to collaborate on the technical report, self-reflection reports, managing datasets and tracking our progress.
5. For version control and technical collaboration, we used GitHub which contained our Mappings, Datasets, Outputs, Queries, Ontology and User Interface. The repository can be accessed here: https://github.com/taaanmay/KDE-Group-K

Conclusion:

Self-reflection of the group on strengths and weaknesses of

Ontology model

The strengths of our ontology lies in its simplicity. It's very easy to understand and every class and relationship makes sense. The properties are intuitive and there are no great leaps needed to fully comprehend the material. The weakness that comes with this is that the ontology hasn't got any novel or 'interesting' ideas. The domain is quite mundane and the ontology reflects this. Overall this ontology succeeds in documenting and visualising our project and that is the most important strength possible.

Queries

Our queries are easy to understand because of the URIs that we have defined. This allows us to complete complex tasks efficiently through simple SPARQL queries. We believe the queries we created have real-world applications as these are questions that house buyers in Ireland may wish to be answered. Certain queries return values in a specific range while others return a single lowest or highest value. If we had additional time to work on this project, we may have added more queries & added additional relevant datasets.

Interface

Our interface is overall quite strong. It is visually appealing and user-friendly; the dropdown for questions makes it very easy to choose between the different questions, and the editable fields are in-line in the question, making editing the queries very intuitive. The results are clearly displayed and easy to see and understand.

There are things we would have changed or added if we had more time: right now, not all variables are editable. We limited the editable fields to fields that could easily be typed by a user, so we were not able to use fields like school program type, which needs values like "All mainstream national school programmes", which are too long to be reliably and easily entered in a text box. If we had more time, we would have created dropdowns for values like these, to allow for a greater variety of queries.

We also would have liked to automatically generate natural language versions of the results fields, eg. convert to “Total Loans Approved” rather than “totalLoansApproved”, but couldn’t due to time constraints.

Additionally, we also wanted to develop a SPARQL Query Editor that would allow the user to create custom queries. However, that was a part of an extra milestone for us and due to time constraints, we could not implement that.