Housing Inspection Explorer: A Tool for Viewing Inspection Scores Among Public and Multihousing Units

Team Name: Housing Inspection Explorer Team

Project Type: Simple Project

Team Members:

- Phillips, Christopher Trevor [ctp180001]
- Nazneen, Amtul [axn180041]

1. INTRODUCTION

The Housing Inspection Explorer (HIE) provides insight into the Multifamily and Public Housing Inspection Scores published by the Public Housing Authority (PHA). PHA performs the inspections for metropolitan and micropolitan statistical areas, which are geographical entities defined by the U.S. Office of Management and Budget (OMB).

HIE has features which allows users to view all inspections scores within a particular state, locate the top states and counties with highest or lowest aggregated inspection scores, and all inspections that were performed in a given year. It also gives insight into cities that have acceptable Housing Inspection Scores, and the estimated next inspection dates for specific housing properties. Thus, the tool provides an intuitive means to view and analyze Housing Inspection Scores in a convenient graphical reporting interface.

2. TARGET AUDIENCE

HIE gives access to the published PHA Housing Inspection Scores through interactive dynamic reports and can be used by agencies to publish Federal statistics. HIE also assists potential home buyers and property owners to determine if a given location is decent, safe, sanitary and in good repair. Thus, the target audience for the HIE is comprised of both government agents/representatives and independent citizens, both of whom gauge the "wellness" of housing properties for their various respective purposes.

3. DESCRIPTION OF DATA SOURCES

Datasets under consideration are:

- Dataset 1258- Public Housing Physical Inspection Scores (No. of triples: 1278872)
- Dataset 1259- Multifamily Housing Physical Inspection Scores (No. of triples: 1455325)

Housing properties are identified using a unique property identification number (or Property ID), and each housing property is associated with various attributes such as location (address, city etc.), inspection date, and inspection score. Thus, each triple in both datasets has the following ontological form:

[Property ID] -> [Attribute] -> [Value]

An example of this can be seen for a single housing property below. The housing property identified by "Entry35679" acts as the subject for several triples, each of which has a relevant attribute and associated value (e.g. [Entity35879] -> [Zip] -> [59405]).

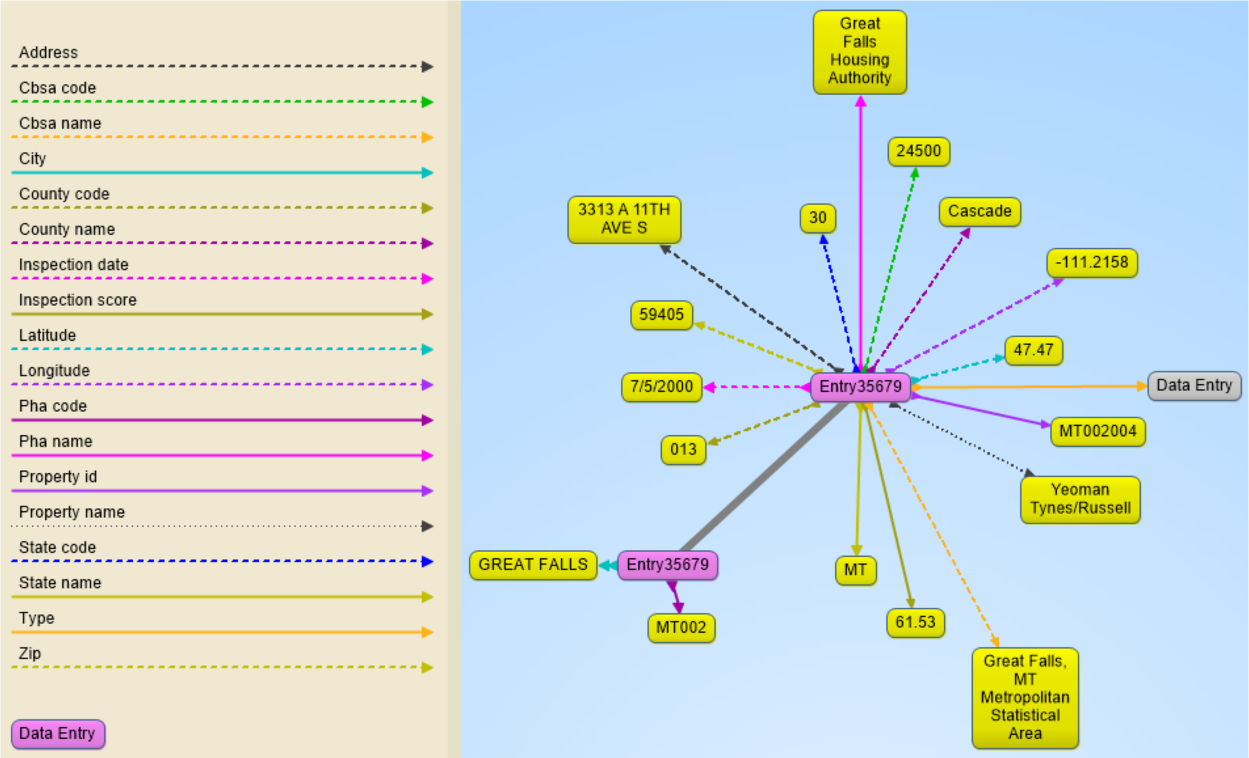


Figure 1: A single property with all associated attributes and values.

Full descriptions of all attributes in both datasets are given in the tables below.

Dataset 1258 – Attributes

Attribute	Description
Property_id	Property identification number
Property_name	Full property name
address	Property street address
city	Property city name
Cbsa_name	Core-based statistical area name
Cbsa_code	Core-based statistical area code
County_name	County name
County_code	County FIPS code
State_name	State postal abbreviation
State_code	State FIPS code
zip	ZIP Code
latitude	Property latitude
longtitude	Property longitude
Pha_code	Public Housing Authority code
Pha_name	Formal PHA name
Inspection_score	Overall inspection score
Inspection_date	Date of inspection

Dataset 1259 – Attributes

Attribute	Description
Property_id	Property identification number
Property_name	Full property name
address	Property street address
city	Property city name
Cbsa_name	Core-based statistical area name
Cbsa_code	Core-based statistical area code
County_name	County name
County_code	County FIPS code
State_name	State postal abbreviation
State_code	State FIPS code
zip	ZIP Code
latitude	Property latitude
longtitude	Property longitude
Inspection_score	Overall inspection score
Inspection_date	Date of inspection

4. DATA INTEGRATION

The next step was to integrate both datasets in order to leverage their combined information. However, the SPARQL endpoint on the LOGD website (where the datasets originally resided) stopped functioning, so it was necessary to create our own SPARQL endpoint using Apache Jena Fuseki. Thus, both datasets were downloaded in their raw N-Triples formats and uploaded to a local instance of Apache Jena Fuseki server. This provided a local SPARQL endpoint by which the HIE could query the datasets.

Combining Datasets

Dataset 1258 and 1259: Upon integration, we were able to derive overall information about the inspection scores across the U.S. for both the Public Housing and Multifamily Housing Datasets by combining on a shared attribute. We accomplished this by combining on the "State_name" attribute common to both datasets, which lists the state associated with a housing property (e.g. "CA", "TX", etc.).

Join Attribute	Description
State_name	State Postal Abbreviation

5. DATA PRODUCT RESULTS

5.1 SPARQL Queries

By integrating with the '**state_name**' attribute, SPARQL Queries were executed against the Fuseki SPARQL Endpoint. SPARQL Queries along with their expected results are discussed below.

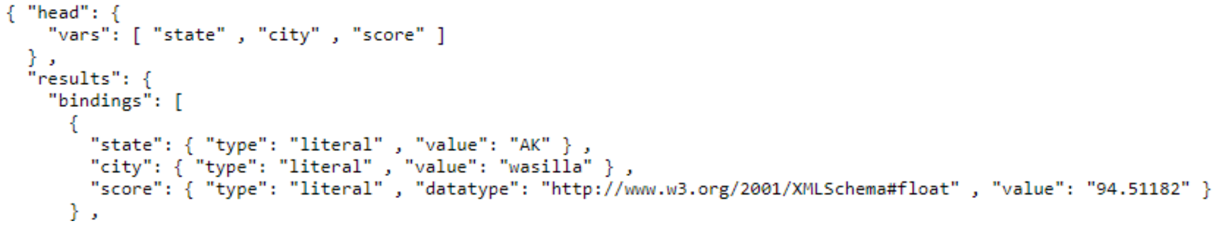
Query: Average Scores by State

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
```

```
SELECT ?state (AVG(xsd:float(?inspection_score)) AS ?score)
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1258>
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1259>
WHERE
{
  {
    ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1258/state_name> ?state.
    ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_score> ?inspection_score.
  }
  UNION
  {
    ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1259/state_name> ?state.
    ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_score> ?inspection_score.
  }
}
GROUP BY ?state
ORDER BY ?state
```

Result

Gives average Housing Inspection Scores for each state (that is, the scores obtained by averaging all Housing Inspection Scores within the state).



```
{ "head": {
    "vars": [ "state" , "city" , "score" ]
  } ,
  "results": {
    "bindings": [
      {
        "state": { "type": "literal" , "value": "AK" } ,
        "city": { "type": "literal" , "value": "wasilla" } ,
        "score": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#float" , "value": "94.51182" }
      } ,
```

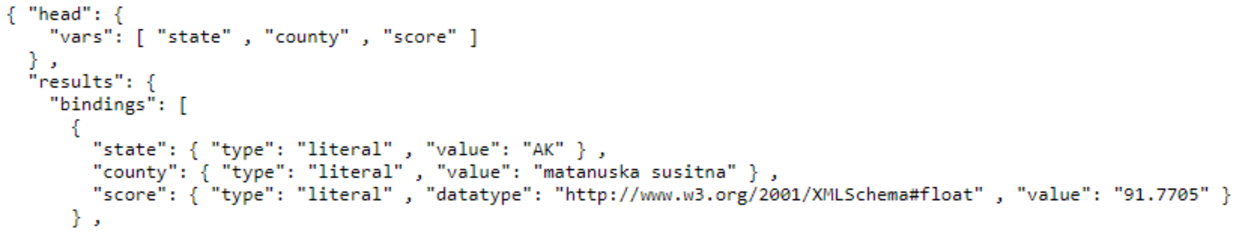
Query: Average Scores by County

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?state ?county (AVG(xsd:float(?inspection_score)) AS ?score)
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1258>
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1259>
WHERE
{
  {
    SELECT ?state (LCASE(?county_name_mc) as ?county) (LCASE(?city_mc) as ?city)
?inspection_score
    WHERE{
      {
        ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/state_name> ?state.
```

```
          ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/county_name> ?county_name_mc.
          ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_score> ?inspection_score.
        }
        UNION
        {
          ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/state_name> ?state.
          ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/county_name> ?county_name_mc.
          ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_score> ?inspection_score.
        }
      }
    }
}
GROUP BY ?state ?county
ORDER BY ?state ?county
```

Result

Gives average Housing Inspection Scores for all counties, grouped by state (that is, the scores obtained by averaging all Housing Inspection Scores within the county, grouped by state). A similar query is used for calculate average scores for all cities grouped by state by replacing all instances of county with city within the query.



```
{ "head": {
    "vars": [ "state" , "county" , "score" ]
  } ,
  "results": {
    "bindings": [
      {
        "state": { "type": "literal" , "value": "AK" } ,
        "county": { "type": "literal" , "value": "matanuska susitna" } ,
        "score": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#float" , "value": "91.7705" }
      } ,
```

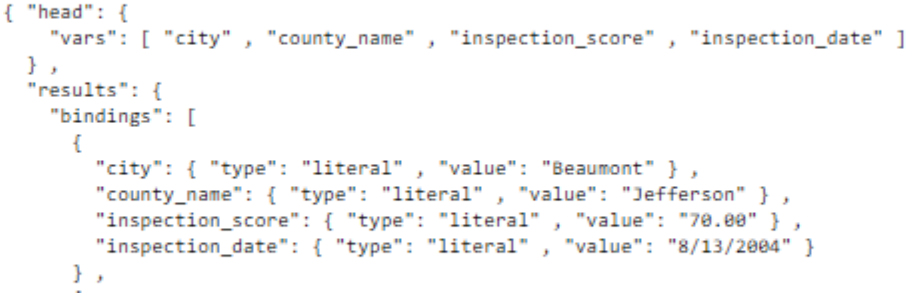
Query: Inspection Information

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?city ?county_name ?inspection_score ?inspection_date
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1259>
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1258>
WHERE{
  {
   SELECT ?state_name ?inspection_score ?inspection_date
(CONCAT(UCASE(SUBSTR(?city_mc, 1, 1)), LCASE(SUBSTR(?city_mc, 2))) as ?city)
(CONCAT(UCASE(SUBSTR(?county_name_mc, 1, 1)), LCASE(SUBSTR(?county_name_mc,
2))) as ?county_name)
      WHERE{
  {
```

```
?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/state_name> ?state_name FILTER
regex(?state_name, "state")  .
?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/city> ?city_mc.
?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/county_name> ?county_name_mc.
?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_score> ?inspection_score.
?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_date> ?inspection_date
  FILTER regex(?inspection_date, "year")
  } UNION
    {
?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/state_name>?state_name FILTER
regex(?state_name, "state")  .
?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/city> ?city_mc.
?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/county_name> ?county_name_mc.
?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_score> ?inspection_score.
?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_date> ?inspection_date
  FILTER regex(?inspection_date, "year")
  }
}
  }
}
```

Result

Gives information about all the inspections that happened in a given year and for a given state.



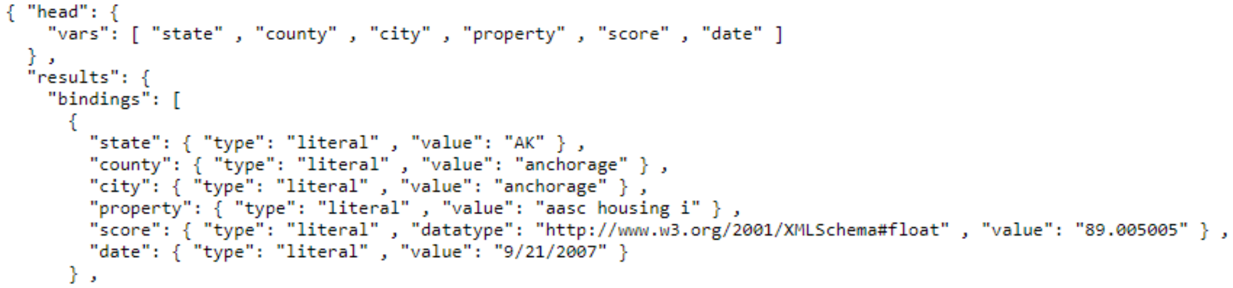
```
{ "head": {
    "vars": [ "city" , "county_name" , "inspection_score" , "inspection_date" ]
  } ,
  "results": {
    "bindings": [
      {
        "city": { "type": "literal" , "value": "Beaumont" } ,
        "county_name": { "type": "literal" , "value": "Jefferson" } ,
        "inspection_score": { "type": "literal" , "value": "70.00" } ,
        "inspection_date": { "type": "literal" , "value": "8/13/2004" }
      } ,
```

Query: Average Scores, Latest Inspection Dates by Property

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?state ?county ?city ?property (AVG(xsd:float(?inspection_score)) AS ?score)
(MAX(?inspection_date) AS ?date)
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1258>
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1259>
WHERE
{
    {
        SELECT ?state (LCASE(?county_name_mc) as ?county) (LCASE(?city_mc) as ?city)
(LCASE(?property_name_mc) as ?property) ?inspection_date ?inspection_score
        WHERE{
            {
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/state_name> ?state.
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/county_name> ?county_name_mc.
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/city> ?city_mc.
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/property_name> ?property_name_mc.
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_date> ?inspection_date.
                ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_score> ?inspection_score.
            }
            UNION
            {
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/state_name> ?state.
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/county_name> ?county_name_mc.
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/city> ?city_mc.
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/property_name> ?property_name_mc.
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_date> ?inspection_date.
                ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_score> ?inspection_score.
            }
        }
    }
}
GROUP BY ?state ?county ?city ?property
ORDER BY ?state ?county ?city ?property
```

Result

Gives average Housing Inspection Scores and latest inspection dates for all properties, grouped by state, county, and city.



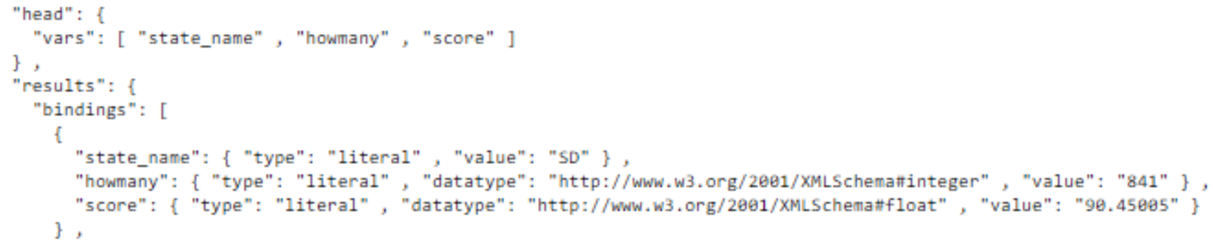
```
{ "head": {
    "vars": [ "state" , "county" , "city" , "property" , "score" , "date" ]
  } ,
  "results": {
    "bindings": [
      {
        "state": { "type": "literal" , "value": "AK" } ,
        "county": { "type": "literal" , "value": "anchorage" } ,
        "city": { "type": "literal" , "value": "anchorage" } ,
        "property": { "type": "literal" , "value": "aasc housing i" } ,
        "score": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#float" , "value": "89.005005" } ,
        "date": { "type": "literal" , "value": "9/21/2007" }
      } ,
```

Query: Top/Bottom 5 Scoring States

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT ?state_name (COUNT(*) AS ?howmany) (AVG(xsd:float(?inspection_score)) AS
?score)
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1259>
FROM <http://data-gov.tw.rpi.edu/vocab/Dataset_1258>
WHERE{
  {
    SELECT ?state_name ?inspection_score (LCASE(?city_mc) as ?city)
(LCASE(?county_name_mc) as ?county_name)
    WHERE{
      {
        ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/state_name> ?state_name.
        ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/inspection_score> ?inspection_score.
        ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/city> ?city_mc.
        ?s1 <http://data-gov.tw.rpi.edu/vocab/p/1259/county_name> ?county_name_mc.
      }
      UNION
      {
        ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/state_name> ?state_name.
        ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/inspection_score> ?inspection_score.
        ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/city> ?city_mc.
        ?s2 <http://data-gov.tw.rpi.edu/vocab/p/1258/county_name> ?county_name_mc.
      }
    }
  }
}
GROUP BY ?state_name
HAVING (?howmany >= 100)
ORDER BY DESC(?score)
LIMIT 5
```

Result

Gives top 5 States with the Overall Highest Housing Inspection Scores. The same query was used and modified to get the 5 states with the Overall Lowest Housing Inspection Scores

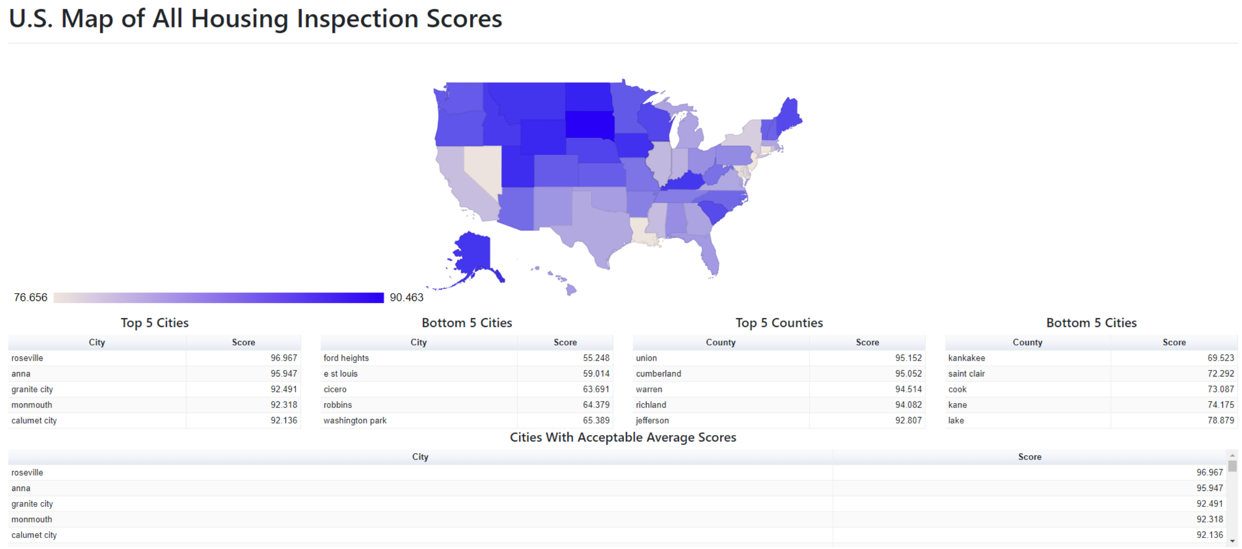


```
 "head": {
   "vars": [ "state_name" , "howmany" , "score" ]
 } ,
 "results": {
   "bindings": [
     {
       "state_name": { "type": "literal" , "value": "SD" } ,
       "howmany": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#integer" , "value": "841" } ,
       "score": { "type": "literal" , "datatype": "http://www.w3.org/2001/XMLSchema#float" , "value": "90.45005" }
     } ,
```

5.2 Visualizations

With the necessary SPARQL queries written, the final step was to create the HIE itself. This was accomplished by creating four separate visualizations which comprise the HIE (using the Google Charts library), where each visualization is accessible via an index page. These are discussed in greater detail below.

Visualization: U.S. Map of All housing Inspection Scores



Top 5 Cities

City	Score
roseville	96.967
anna	95.947
granite city	92.491
monmouth	92.318
calumet city	92.136

Bottom 5 Cities

City	Score
ford heights	55.248
e st louis	59.014
cicero	63.691
robbins	64.379
washington park	65.389

Top 5 Counties

County	Score
union	95.152
cumberland	95.052
warren	94.514
richland	94.082
jefferson	92.807

Bottom 5 Cities

County	Score
kankakee	69.523
saint clair	72.292
cook	73.087
kane	74.175
lake	78.879

Cities With Acceptable Average Scores

City	Score
roseville	96.967
anna	95.947
granite city	92.491
monmouth	92.318
calumet city	92.136

Figure 2: Geochart with selectable states for viewing city/county scores and cities with acceptable scores

Result

This visualization displays the top/bottom 5 scoring cities within a state, the top/bottom 5 scoring counties within a state, and all cities with acceptable scores within a state. The user clicks on a state in the interactive map of all 50 U.S. states, and the charts below the map automatically populate with the relevant information.

Visualization: Top States and Inspection Scores

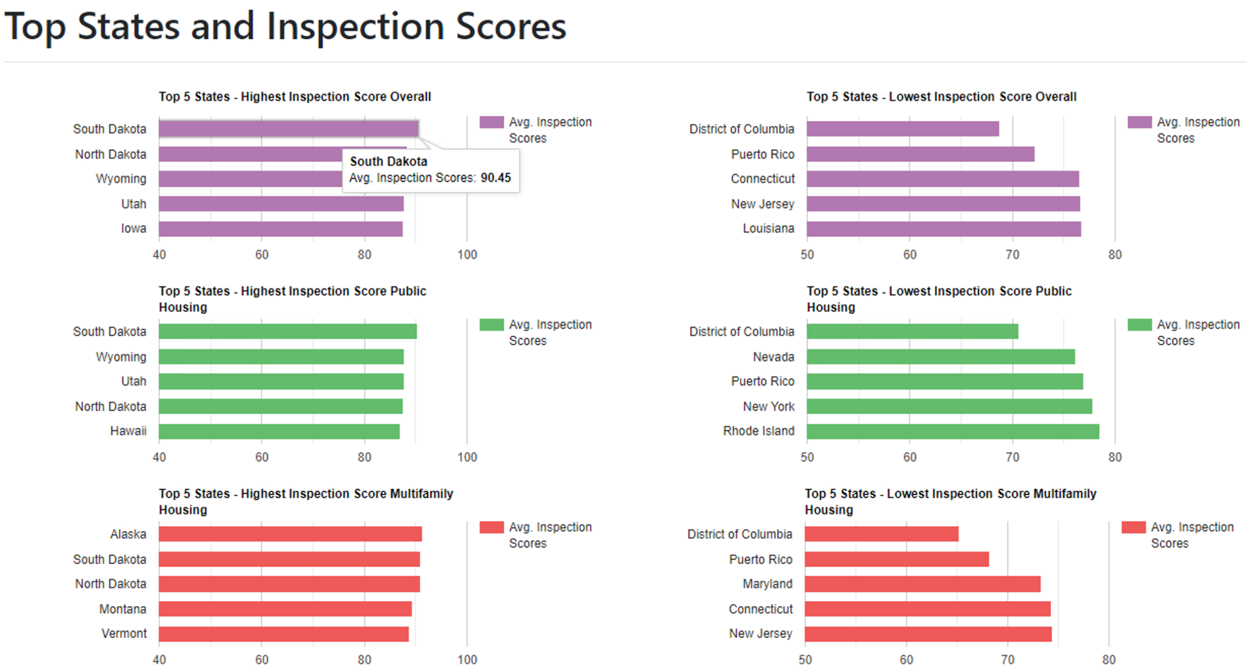


Figure 3: Top 5 States with the Overall Highest/Lowest Housing Inspection Scores

Result

This visualization displays the top/bottom 5 scoring states across the entire U.S. Hovering over each state's corresponding bar generates a tooltip with the average score for that particular state.

Visualization: State/County Score & Next Inspection Date Viewer

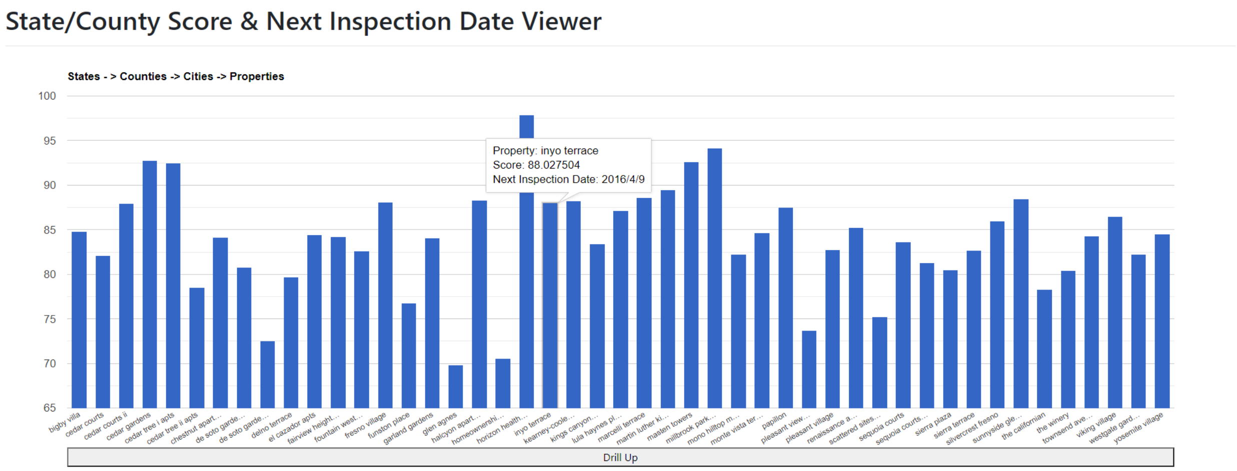
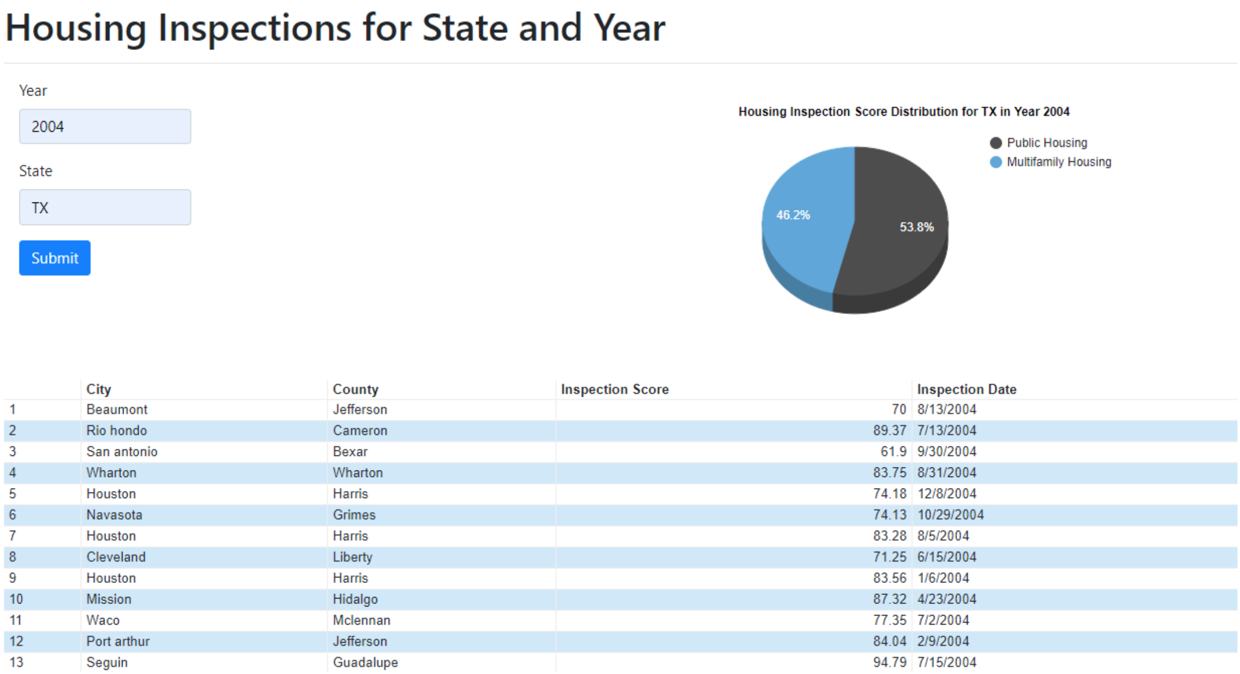


Figure 4: Drilldown chart (State -> County -> City -> Property hierarchy) for scores, inspection dates

Result

This visualization allows the user to view average scores at various levels within its drilldown hierarchy (e.g., state level, county level, etc.), and to drill down to individual housing properties. At this lowest level, the user may view the estimated date of next inspection for a particular housing property by hovering over its corresponding bar and viewing the date value in its tooltip.

Visualization: Housing Inspections for State and Year



Housing Inspections for State and Year

Year: 2004

State: TX

Submit

	City	County	Inspection Score	Inspection Date
1	Beaumont	Jefferson	70	8/13/2004
2	Rio hondo	Cameron	89.37	7/13/2004
3	San antonio	Bexar	61.9	9/30/2004
4	Wharton	Wharton	83.75	8/31/2004
5	Houston	Harris	74.18	12/8/2004
6	Navasota	Grimes	74.13	10/29/2004
7	Houston	Harris	83.28	8/5/2004
8	Cleveland	Liberty	71.25	6/15/2004
9	Houston	Harris	83.56	1/6/2004
10	Mission	Hidalgo	87.32	4/23/2004
11	Waco	Mclennan	77.35	7/2/2004
12	Port arthur	Jefferson	84.04	2/9/2004
13	Seguin	Guadalupe	94.79	7/15/2004

Figure 5: All the inspections that happened in the given year and for a given state

Result

This visualization displays specific housing inspection dates for a year and state specified by the user.

6. SUMMARY

The initial step involved identifying the datasets that could be used for the project. We scanned the LOGD catalog and after preliminary analysis on the available datasets, we chose Datasets 1258 and 1259 for our final concept.

After this, we began analyzing our datasets in more detail to get a deeper understanding of the data and the joining attributes. SPARQL Queries were constructed and executed against Apache Jena Fuseki to get the data for the features defined in our concept.

Proceeding towards the implementation phase, we loaded our triples from the two datasets into Fuseki. We then leveraged the Google Charts library for the visualization of results, and used AJAX calls to Fuseki to populate our visualizations via the relevant SPARQL queries.

Through these steps, we were successful in implementing the HIE, which allows users to effectively view and analyze Housing Inspection Scores across the two datasets we leveraged. The only issue encountered was the need to create our own Fuseki instance when the LOGD SPARQL endpoint ceased to function, but establishing a Fuseki instance was thankfully straightforward. Thus, this project taught us the various steps involved in creating a reporting application leveraging the Semantic Web, from initial steps (identifying potential datasets), intermediate steps (establishing a Fuseki instance), and final steps (creating the actual reporting application using Google Charts).