# WORKING WITH REAL-WORLD LINKED DATA ECOSYSTEMS

Let's apply the course concepts we have studied in understanding linked data ecosystems in the real world

## Opening prompt / problem statement

Linked Data is **a set of design principles for sharing machine-readable interlinked data on the Web**. When combined with Open Data (data that can be freely used and distributed), it is called Linked Open Data (LOD).

Imagine you are an enterprise of service provider that needs to convert your data or offering to Linked Data. For example, you could be a news media publication that wants to expose the entities you write about as URLs and Linked Data, not dissimilar to what the New York Times and BBC tried to do.

## Context and opportunity

To achieve and create Linked Data, technologies should be available for a common format (RDF), to make either conversion or on-the-fly access to existing databases (relational, XML, HTML, etc.). It is also important to be able to setup query endpoints to access that data more conveniently.

## Ways of getting RDF

There are at least 4 different ways of fetching RDF, with some noteworthy examples and modalities below:


RDF Dump


REST Service


SPARQL Endpoint


URI Derefencing

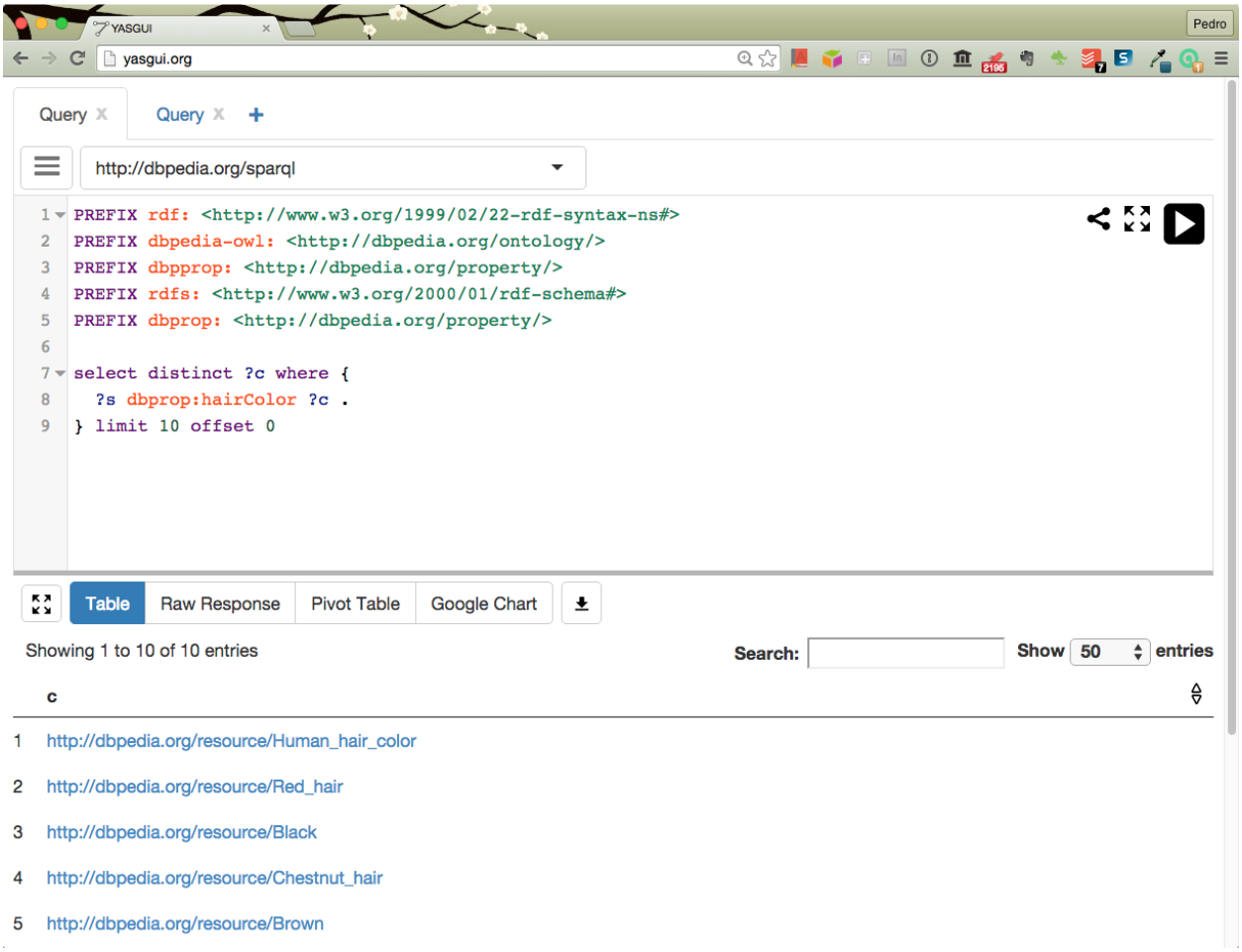
## Example: OpenLink Browser

One early example of Linked Data in practice was the OpenLink browser, which could be used to retrieve detailed results with semantic metadata about entities (such as Los Angeles):


## Case Questions

i) Earlier in the case, we said that a browser such as OpenLink may be one intuitive way of getting Linked Data. However, in the event that you build such a browser and the entity is not found, what should the browser do? Consider various design choices, and list their pros and cons, both from a system developer's standpoint and a user's standpoint.

(ii) Another way of retrieving data is using a SPARQL endpoint. However, here the assumption is that the user knows how to navigate the endpoint. Here's one example:




What are the pros and cons of this approach? Can you name a niche market (or markets) that might prefer this kind of approach over the browser?

(iii) Let's consider again the 4 different ways of getting RDF that we presented earlier in the case. Select one example from each of the four and do a detailed comparison. Are there any decisions you would have made differently? For example, would you have exposed data.gov as an RDF dump? Why or why not?

(iv) Suppose you were asked to consult with a medium sized company whose data is currently sitting in databases and 'traditional' data stores but that has decided to make data available internally as Linked Data. How would you go about doing this? Would you still need URLs? Would the URLs be accessible through a browser such as Chrome?