Describe the benefits that Linked Data could bring

Linked Data is sharable, extensible, and easily re-usable.

Linked Data supports multilingual functionality for data and user services, such as the labeling of concepts identified by language-agnostic URIs.

Linked Data allows anyone to contribute unique expertise in a form that can be reused and recombined with the expertise of others.

Linked Data technology can help organizations improve their internal data curation processes and maintain better links between.

The concept of Linked Data

Linked data is a global initiative to publish and interlink structured data on the web using a clever combination of simple, standardised technologies, such as:

- Uniform Resource Identifiers (URI) - To name things
- Resource Description Framework (RDF) - To represent things
- HTTP Infrastructure - to obtain those representations

It is also a community effort to publish open data sets as Linked Data on the Web (to which anyone can refer to).
It also involves developing clients that consume linked data from the web

A set of principles for linked data, formulated by Tim Berners Lee, are as follows:

- Use URIs as names for things
- Use HTTP URI's so that either a person or an app can look up these names.
- When someone or an app searches for a URI, useful information is provided.
- Links are included to other URI's so they can discover more information

The concept of the Semantic Web

In short, the semantic web is a 'web of data'.

- As of current, the web has made data available, by providing an infrastructure where data can be accessed and documents can be retrieved and represented.
- However, there is no semantic interpretation of the data (ie. understanding what the data means). As of current, documents are linked, but data itself isn't.
- Currently, we have a web of documents, whereby humans have developed for other humans, but the documents bear little meaning for machines, (the links between documents bore little meaning for machines).

- Also, the documents have little structured information.
- The semantic web aims to link data rather than documents, and to create a global database of information.
- In the semantics web, there is a high degree of structure, with explicit semantics between links and documents.
- In the semantic web, the collection of technologies provide an environment where applications can query that data, draw inferences using vocabularies, etc., which would provide a huge advantage for application developers.
- WRITE MORE HERE

The Semantic Web Stack

- The Semantic Web Stack, also known as Semantic Web Cake or Semantic Web Layer Cake, illustrates the architecture of the Semantic Web.
- The Semantic Web Stack is an illustration of the hierarchy of languages, where each layer exploits and uses capabilities of the layers below. It shows how technologies that are standardized for Semantic Web are organized to make the Semantic Web possible. It also shows how Semantic Web is an extension (not replacement) of classical hypertext web.
- The Semantic Web stack builds on the W3C's Resource Description Framework (RDF)
- The Semantic Web Stack shows that Hypertext Web technologies, Standardized Semantic Web technologies, Unrealized Semantic Web technologies are used to create Semantic Web.

In what way OWL builds on RDF and what benefits this brings

- RDF provides a way of building graphs from triples, but doesn't constrain the graph too much. **The problem is that RDF is an untyped mechanism for building graphs**
- This is a problem in interpretation and scaling.
- **RDFS** is an 'extension' is RDF, adding schema vocabulary to the mix, however RDFS too weak to describe resources in sufficient detail
- We need a language that is more expressive and has a richer vocabulary compared to RDFS, and for that reason we introduce Ontology Web Language (OWL).

From slides:
OWL provides several other mechanisms for defining classes

• **equivalentClass** allows you to state that two classes are synonymous.
• disjointWith allows you to state that an instance of this class cannot be an instance of another.
• **unionOf** allows you specify that a class contains things that are from more than one class.

• **intersectionOf** allows you to specify that a class contains things that are both in one
and the other.
• **complementOf** allows you specify that a class contains things that are not other things

From internet:

- RDFS allows you to express the relationships between things by standardizing on a flexible, triple-based format and then providing a comparatively smaller vocabulary (such as rdf:type or rdfs:subClassOf) which can be used to say things about concepts in your area(s) of interest.

- OWL is similar, but bigger, better, and badder. OWL lets you say much more about your data model; it shows you how to work efficiently with database queries and automatic reasoners; and it provides useful annotations for bringing your data models into the real world.

- Vocabulary

  Of the differences between RDFS and OWL, arguably the most important is just that OWL provides a much larger vocabulary.

- Logical Consistency

  In contrast to RDFS, OWL tells you how you *can* and *cannot* use certain vocabulary. In other words, whereas RDFS provides no real constraint mechanisms, OWL does.

- Constraints and Computability

  Unlike RDFS, OWL lets you decide how expressive you want to be, given the computational realities involved.

- Annotations, the meta-meta-data

  OWL can easily use linked data models together into a mutually coherent network of ontologies.OWL is sure to satisfy all of the meta-meta-data-modeling needs.