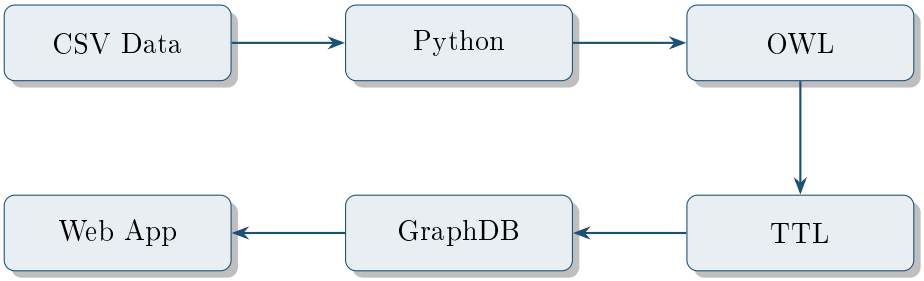
PKG2020 Knowledge Graph

Viva Preparation Guide

Complete Walkthrough of Project Components



Project Statistics

23 Classes	**14 Object Props**	**17+ Data Props**	**2.2M Triples**

Knowledge Representation and Reasoning
Fall 2025

Contents

1 Project Overview

What is this Project?

This project converts the **PKG2020S4 PubMed Knowledge Graph** dataset from flat CSV files into a semantic **OWL ontology** with linked data capabilities, published via a **SPARQL endpoint**.

1.1 Domain: Biomedical Research Data

The dataset contains:

- **Articles**: Research publications identified by PubMed IDs (PMIDs)
- **Authors**: Researchers identified by AND_IDs
- **Organizations**: Universities, research labs, hospitals
- **Career Data**: Employment and education history
- **BioEntities**: Genes, diseases, chemicals, mutations
- **NIH Funding**: Research project associations

1.2 Deployed URLs

Web Application	`https://krr-685beba13d3f.herokuapp.com`
GraphDB Endpoint	`https://x1327f4041a654297998.sandbox.graphwise.ai`
GitHub Repository	`https://github.com/Zain-ul-abdeen-773/Knowlege-Graphs-Project`

2 Key Concepts for Viva

2.1 T-Box vs A-Box

T-Box (Terminological Box)

The **schema/vocabulary** - defines classes, properties, and axioms.

- Classes: Article, Author, Gene, Disease
- Properties: writtenBy, hasAffiliation, mentionsBioEntity
- Axioms: "Every Article has at least 1 Author"

File: `pkg2020_tbox_only.owl`

A-Box (Assertion Box)

The **instance data** - actual individuals and their relationships.

- Article_12345678 is an Article
- Article_12345678 writtenBy Author_ABC
- Author_ABC hasAffiliation Affiliation_XYZ

File: `pkg2020_final.owl` (contains both T-Box + A-Box)

2.2 Defined Classes

Key Point

A **defined class** has **necessary AND sufficient conditions**. The reasoner can automatically classify individuals into defined classes!

Class	Definition	Type
ActiveAuthor	Author ⊓ ∃careerStartYear.int	Intersection
AnonymousAuthor	Author ⊓ ¬ActiveAuthor	Complement
ResearchEntity	Author ⊔ Article	Union
ProlificAuthor	Author ⊓ writtenBy.min(5)	Cardinality

2.3 Property Types

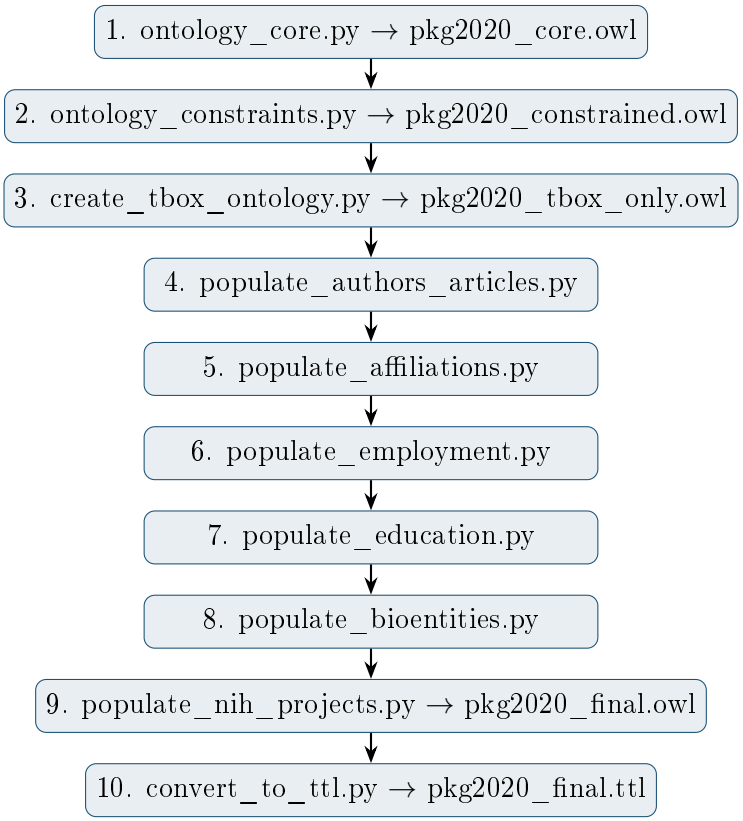
Type	Example	Meaning
Functional	hasPrimaryAuthor	Max 1 value
Inverse Functional	hasPMID	Unique identifier
Symmetric	sameAs	A→B means B→A
Transitive	hasPart	A→B→C means A→C

3 Project Files Explained

3.1 Directory Structure

```
Project/
        data/                       # CSV source files
              OA01_Author_List.csv      # Authors & articles
              OA02_Bio_entities_Main.csv # Genes, diseases
              OA03_Bio_entities_Mutation.csv
              OA04_Affiliations.csv     # Author affiliations
              OA05_Researcher_Employment.csv
              OA06_Researcher_Education.csv
              OA07_NIH_Projects.csv     # NIH funding
        owl/                        # Generated OWL files
              pkg2020_tbox_only.owl     # Schema only (T-Box)
              pkg2020_hand_annotated.owl # 10+ test individuals
              pkg2020_with_swrl.owl     # SWRL rules
              pkg2020_final.owl         # Complete ontology
              pkg2020_final.ttl         # Turtle for GraphDB
        scripts/                    # Python scripts
              [16 Python files]
        docs/                       # Documentation
```

3.2 Python Scripts Pipeline



4 Rubric Compliance Checklist

4.1 Classes Requirements

Requirement	Status	Location
20+ classes	23 classes	create_tbox_ontology.py:16-107
Enumeration class		ontology_core.py:25-37 `PublicationStatus = OneOf([...])`
Cardinality restrictions		ontology_constraints.py:21-29 `writtenBy.min(1, Author)`
Union class		ontology_constraints.py:38-43 `Author | Article`
Intersection class		ontology_constraints.py:31-36 `Author & careerStartYear.some(int)`
Complement class		ontology_constraints.py:45-50 `Author & Not(ActiveAuthor)`

4.2 Properties Requirements

Requirement	Status	Location
7+ object properties	14	create_tbox_ontology.py:110-181
Functional property		ontology_core.py:55-59 `hasPrimaryAuthor`
Inverse functional		ontology_constraints.py:17-19 `hasPMID`
3+ range restrictions		Multiple files
7+ data properties	17+	create_tbox_ontology.py:184-273

5 SPARQL Competency Queries

5.1 15 Competency Questions

#	Question	Key Pattern
CQ1	Authors at multiple institutions	GROUP BY, HAVING
CQ2	Most prolific authors	COUNT, ORDER BY
CQ3	Author collaborations	Self-join on Article
CQ4	Articles with genes	Class filter (pkg:Gene)
CQ5	Articles with species	OPTIONAL
CQ6	Gene-mutation correlations	Multiple FILTER
CQ7	Bio-entity distribution	COUNT by type
CQ8	Top organizations	COUNT affiliations
CQ9	Affiliations by country	GROUP BY country
CQ10	Top education institutions	COUNT education
CQ11	Employment timeline	FILTER years
CQ12	Authors with education	EXISTS
CQ13	NIH funded authors	hasProject pattern
CQ14	Principal investigators	piName property
CQ15	Complete author profile	Multiple OPTIONAL

5.2 Sample Query: CQ1

Authors at Multiple Institutions

```
PREFIX pkg: <http://example.org/pkg2020/ontology.owl#>

SELECT ?author ?lastName (COUNT(DISTINCT ?org) AS ?count)
WHERE {
    ?author a pkg:Author .
    ?author pkg:lastName ?lastName .
    ?author pkg:hasAffiliation ?aff .
    ?aff pkg:affiliatedWith ?org .
}
GROUP BY ?author ?lastName
HAVING (COUNT(DISTINCT ?org) > 1)
ORDER BY DESC(?count)
LIMIT 20
```

6 SWRL Rules (Bonus)

What is SWRL?

Semantic Web Rule Language - allows IF-THEN rules on OWL ontologies.

```
Antecedent (Body) -> Consequent (Head)
```

6.1 7 SWRL Rules Implemented

#	Rule
1	Author(?a) ∧ hasProject(?a, ?p) ∧ NIHProject(?p) → FundedAuthor(?a)
2	Author(?a) ∧ hasEmployment(?a, ?e) ∧ hasEducation(?a, ?d) → EstablishedResearcher(?a)
3	Article(?art) ∧ writtenBy(?art, ?a1) ∧ writtenBy(?art, ?a2) ∧ diff(?a1,?a2) → CollaborativeArticle(?art)
4	Article(?art) ∧ mentionsBioEntity(?art, ?g) ∧ Gene(?g) ∧ mentionsBioEntity(?art, ?d) ∧ Disease(?d) → GeneDiseaseLinkArticle(?art)
5	Author(?a1) ∧ Author(?a2) ∧ educatedAt(?e1, ?inst) ∧ educatedAt(?e2, ?inst) → isAlumniPeerOf(?a1, ?a2)

File: `scripts/create_swrl_rules.py`

7 Reasoning & Consistency Checking

7.1 How Reasoning Works

7.2 Reasoning Results

When the reasoner runs on hand-annotated individuals:

Individual	Has Property?	Classified As
Author_HAND_001	careerStartYear = 2010	ActiveAuthor
Author_HAND_002	No careerStartYear	AnonymousAuthor
Article_HAND_001	1 author	SingleAuthorArticle
Article_HAND_002	3 authors	MultiAuthorArticle

7.3 Python Code for Reasoning

reasoning.py

```
from owlready2 import *

onto = get_ontology("pkg2020_final.owl").load()

with onto:
    sync_reasoner(infer_property_values=True)

# Check consistency
print("Ontology is CONSISTENT")

# Check inferred classes
for a in onto.ActiveAuthor.instances():
    print(f"ActiveAuthor: {a.name}")
```

8 Web Application (Bonus)

8.1 Technology Stack

Component	Technology
Backend	Flask (Python)
Frontend	HTML + D3.js
Database	GraphDB SPARQL Endpoint
Deployment	Heroku
SPARQL Client	SPARQLWrapper

8.2 API Endpoints

Endpoint	Purpose
`/`	Dashboard with statistics
`/sparql`	Raw SPARQL query interface
`/api/stats`	Graph statistics JSON
`/api/query`	Execute SPARQL query
`/api/competency-queries`	All 15 CQs
`/api/graph-data`	D3.js visualization data

8.3 Visualization

The web app includes an interactive D3.js force-directed graph showing:

- All 23 ontology classes as nodes
- Object properties as edges
- Color-coded by category
- Interactive zoom and drag

9 Expected Viva Questions

9.1 Basic Questions

1. **What is T-Box vs A-Box?**
   T-Box = schema (classes, properties). A-Box = data (individuals).

2. **What is a defined class?**
   Has necessary & sufficient conditions. Reasoner classifies automatically.

3. **What is a functional property?**
   Can have at most one value. Example: hasPrimaryAuthor.

4. **What is SWRL?**
   Rule language for OWL. IF-THEN rules for inference.

5. **How many triples in your graph?**
   2.2+ million triples.

9.2 Technical Questions

1. **Explain your enumeration class.**
   PublicationStatus = OneOf([Published, Preprint, Retracted, InReview])

2. **Explain your intersection class.**
   ActiveAuthor = Author $\cap$ $\exists$careerStartYear.int

3. **Explain your complement class.**
   AnonymousAuthor = Author $\cap$ $\neg$ActiveAuthor

4. **How did you handle external linking?**
   DBpedia URIs for organizations, Wikidata for institutions.

5. **What reasoner did you use?**
   HermiT via OWLReady2's sync_reasoner().

9.3 Conceptual Questions

1. **Why convert CSV to RDF?**
   Semantic relationships, inference, linked data, SPARQL.

2. **What is 5-star linked data?**
   1) Web, 2) Machine-readable, 3) Open format, 4) URIs, 5) Links to other data.

3. **Benefits of knowledge graphs?**
   Complex queries, inference, integration, interoperability.

10 Quick Reference Card

Numbers to Remember

Classes	23 (20+ required)
Object Properties	14 (7+ required)
Data Properties	17+ (7+ required)
Triples	2.2 million
Competency Questions	15
SWRL Rules	7
Hand-Annotated Individuals	13

Key Files

T-Box Only	pkg2020_tbox_only.owl
Hand Annotated	pkg2020_hand_annotated.owl
SWRL Rules	pkg2020_with_swrl.owl
Complete Data	pkg2020_final.owl / .ttl

Key Concepts

Enumeration	OneOf([...]) - fixed set of values
Intersection	$\sqcap$ or & - AND condition
Union	$\sqcup$ or | - OR condition
Complement	$\neg$ or Not() - negation
Functional	Max 1 value allowed
Inverse Functional	Unique identifier

Good Luck with Your Viva!