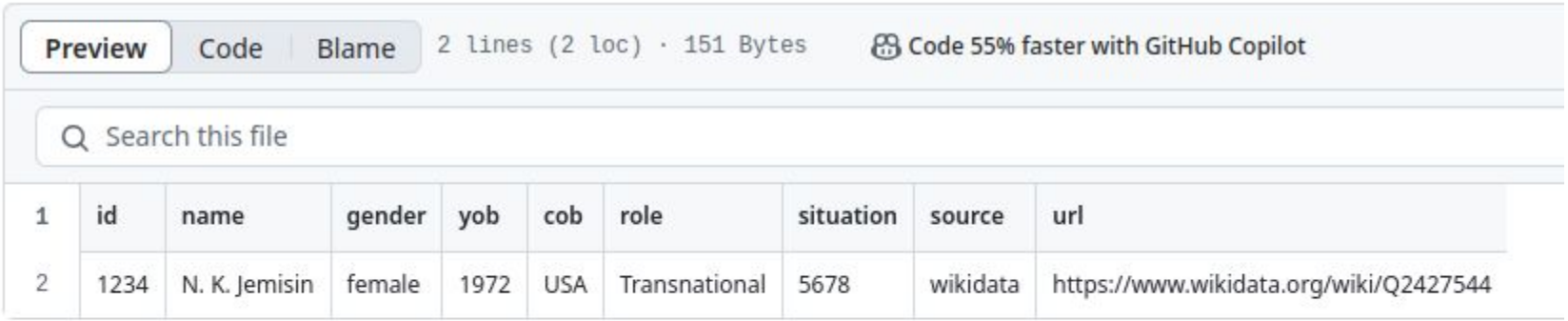
Popolare il World Literature KG con SPARQL-Anything

SB

Let’s SPARQL-Anything (a recap)

1. A dataset



1	id	name	gender	yob	cob	role	situation	source	url
2	1234	N. K. Jemisin	female	1972	USA	Transnational	5678	wikidata	https://www.wikidata.org/wiki/Q2427544

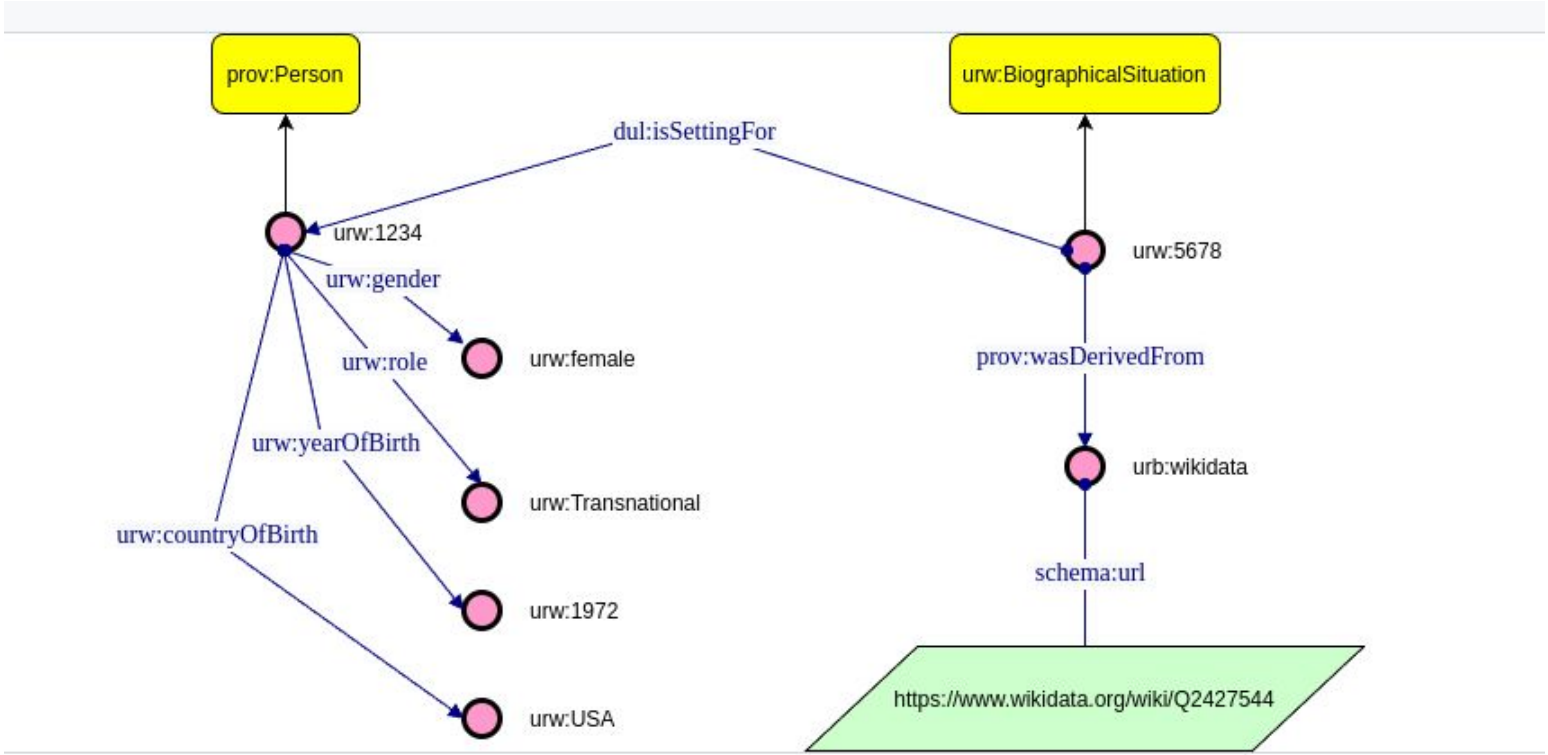
2. A target KG output

```
urw:1234 a prov:Person;
    rdfs:label 'Nora Keita Jemisin';
    urw:gender urw:female;
    dul:hasRole urw:Transnational;
    urw:yearOfBirth urw:1972;
    urw:countryOfBirth urw:USA .

urw:5678 a urw:BiographicalSituation ;
    dul:isSettingFor urw:USA , urw:1972 , urw:Transnational , urw:female , urw:artist1234 ;
    prov:wasDerivedFrom urb:wikidata

urb:wikidata  schema:url  "https://www.wikidata.org/wiki/Q2427544" .
```

2. A target KG output



3. A query to convert the dataset in the KG

```
PREFIX schema: <http://schema.org/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX dul: <http://www.ontologydesignpatterns.org/ont/dul/DUL.owl#>
PREFIX xml: <http://www.w3.org/XML/1998/namespace>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX prov: <http://www.w3.org/ns/prov#>

CONSTRUCT {
  ?writer a prov:Person;
        rdfs:label ?name;
          urw:gender ?wgender;
          dul:hasRole ?wrole;
          urw:yearOfBirth ?wyob;
          urw:countryOfBirth ?wcob .
  ?sit a urw:BiographicalSituation;
    dul:isSettingFor ?writer , ?wgender , ?wrole , ?wyob , ?wcob ;
    prov:wasDerivedFrom ?source .
  ?source schema:url ?url .

} WHERE {
  # Artists from the Tate Gallery open data!
  SERVICE <x-sparql-anything:csv.headers=true,location=./data/author_data.csv> {
    []  xyz:id ?id;
            xyz:name ?name;
          xyz:gender ?gender;
          xyz:role ?role;
          xyz:yob ?yob;
          xyz:cob ?cob ;
       xyz:situation ?situation;
       xyz:url ?url ;
       xyz:source ?src .

  }
  BIND (IRI(CONCAT(STR(urw:),"artist", ?id )) AS ?writer) .
  BIND (IRI(CONCAT(STR(urw:), ?gender )) AS ?wgender) .
  BIND (IRI(CONCAT(STR(urw:), ?role )) AS ?wrole) .
  BIND (IRI(CONCAT(STR(urw:), ?yob )) AS ?wyob) .
  BIND (IRI(CONCAT(STR(urw:), ?cob )) AS ?wcob) .
  BIND (IRI(CONCAT(STR(urw:), ?situation )) AS ?sit) .
  BIND (IRI(CONCAT(STR(urb:), ?src )) AS ?source) .
}
```

SB

3.1. The output format

```
CONSTRUCT {
  ?writer a prov:Person;
        rdfs:label ?name;
          urw:gender ?wgender;
          dul:hasRole ?wrole;
          urw:yearOfBirth ?wyob;
          urw:countryOfBirth ?wcob .
  ?sit a urw:BiographicalSituation;
    dul:isSettingFor ?writer , ?wgender , ?wrole , ?wyob , ?wcob ;
    prov:wasDerivedFrom ?source .
  ?source schema:url ?url .
```

3.2. The triplification of your dataset

```
} WHERE {
  # Artists from the Tate Gallery open data!
  SERVICE <x-sparql-anything:csv.headers=true,location=./data/author_data.csv> {
    []  xyz:id ?id;
            xyz:name ?name;
          xyz:gender ?gender;
          xyz:role ?role;
          xyz:yob ?yob;
          xyz:cob ?cob ;
       xyz:situation ?situation;
       xyz:url ?url ;
       xyz:source ?src .

  }
```

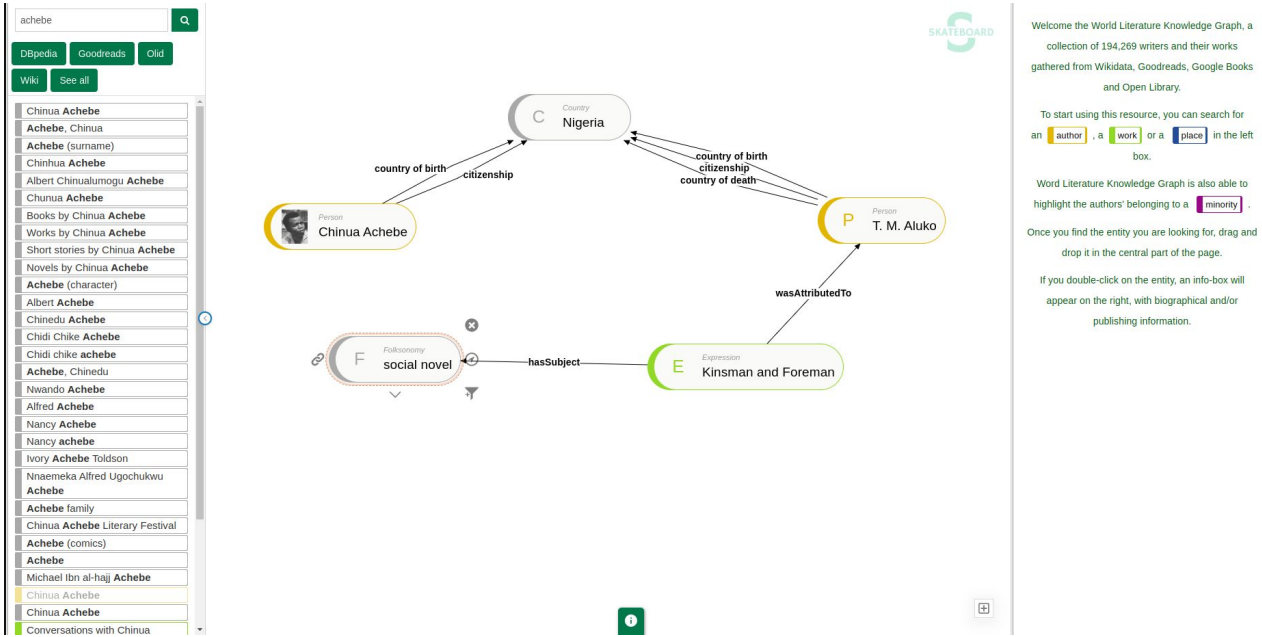
3.3. Variable alignment and iri assignment

```
  BIND (IRI(CONCAT(STR(urw:),"artist", ?id )) AS ?writer) .
  BIND (IRI(CONCAT(STR(urw:), ?gender )) AS ?wgender) .
  BIND (IRI(CONCAT(STR(urw:), ?role )) AS ?wrole) .
  BIND (IRI(CONCAT(STR(urw:), ?yob )) AS ?wyob) .
  BIND (IRI(CONCAT(STR(urw:), ?cob )) AS ?wcob) .
  BIND (IRI(CONCAT(STR(urw:), ?situation )) AS ?sit) .
  BIND (IRI(CONCAT(STR(urb:), ?src )) AS ?source) .
}
```

Your assignment

The World Literature Knowledge Graph

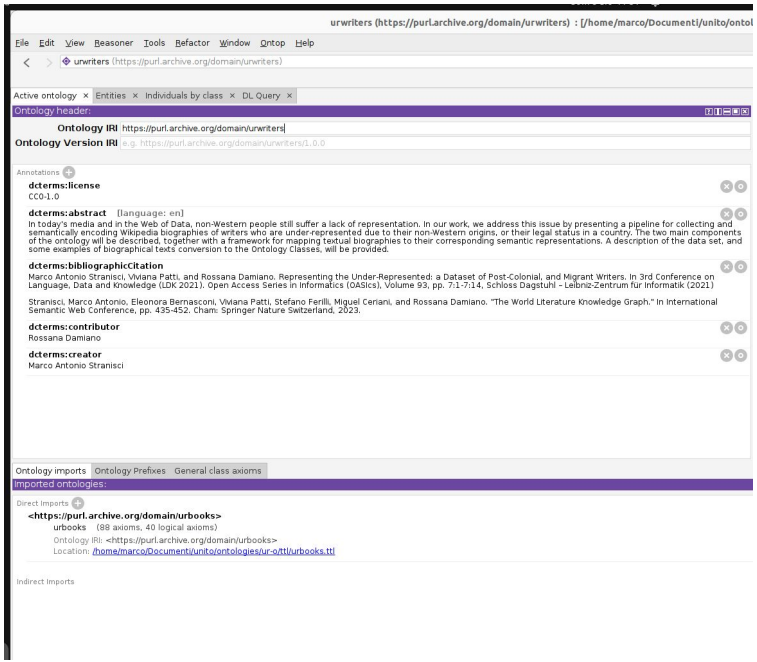
A resource for the exploration of underrepresented writers



SB

The Under-Represented Writers Ontology

Modelled on a semantic model for the alignment of different platforms



The objective

Allow people to suggest writers and works that are not included in the World Literature Knowledge Graph

Your assignment

1. Choose a piece of knowledge to be encoded
2. Check it against it the ontology and, if necessary, modify it
3. Create a Google form for gathering information from users
4. Process the output file of the G form to convert it in the KG
5. Write the PySPARQL-Anything query
6. Document everything in the github folder

Bonus track

Implement a scraper that extracts information from different archives and populates the KG

Timeline

Today: form groups, choose the assignment, and design your activity

Friday (11:00-11:30): we solve eventual issues and clarify doubts

Activity repository: https://github.com/world-literature-kg/sa2wl-kg
Ontologies repository: https://github.com/world-literature-kg/ontologies