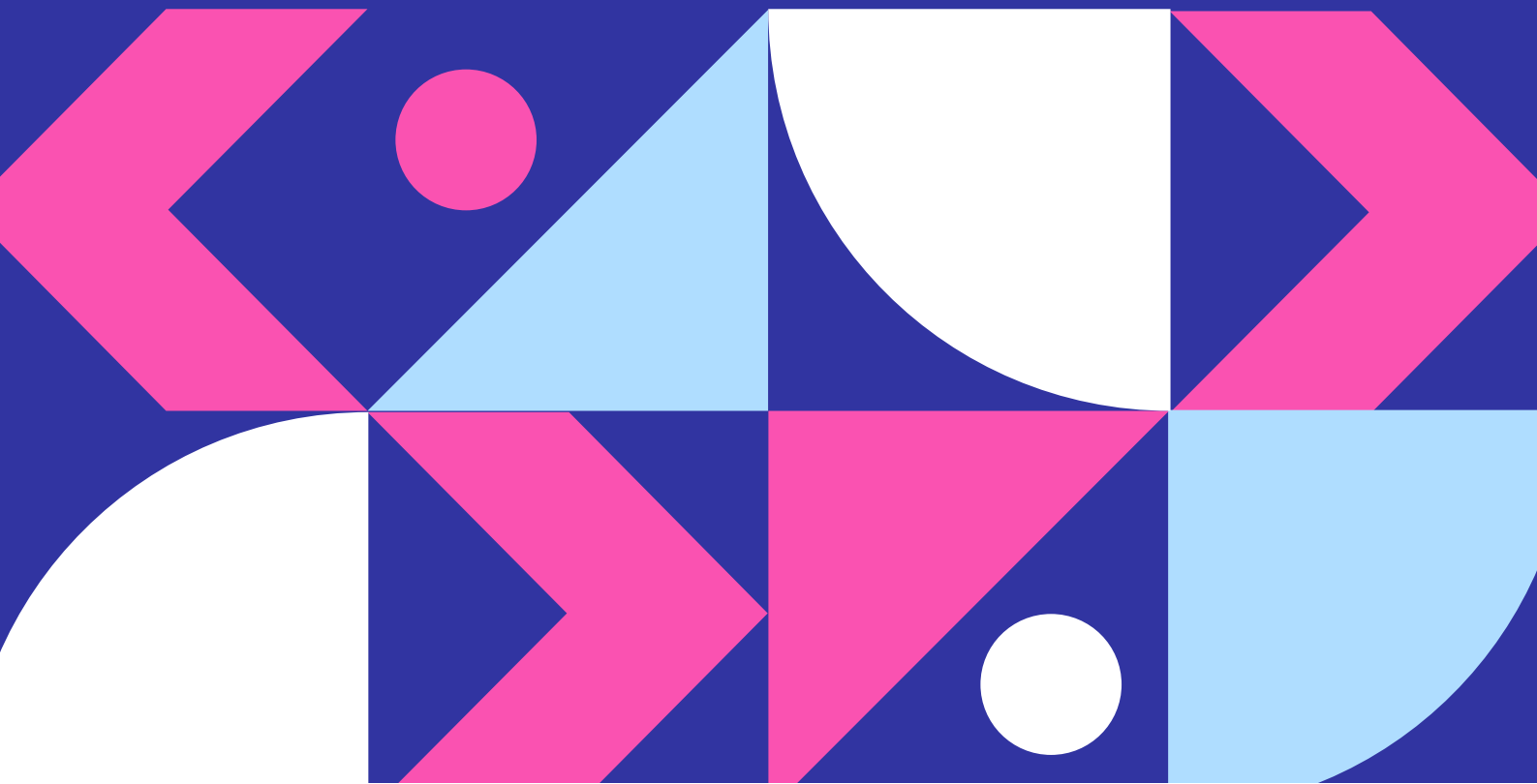
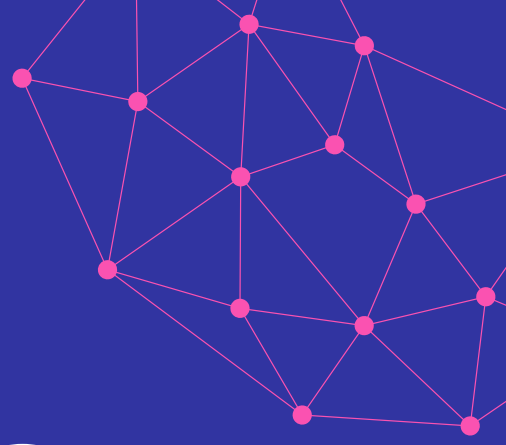


SoCK Use Cases

Explore the use of completeness patterns



#1

G20 Nations



Meet Lisa, a web developer working on a web development project for the G20 Presidency event in Indonesia. Lisa is responsible for creating a feature that shows all the members of the G20. Gradually, Lisa decides to use the DBpedia knowledge graph. However, Lisa is doubtful whether all the data she needs is available on DBpedia. Therefore, Lisa considers doing a validation check first to ensure all information related to the countries that are members of the G20 includes in the knowledge graph.

SoCK Home Patterns **Instances** Use Cases

Create a New Pattern Instance

Name
G20 Nations (DBpedia)

Creator
anonymous

Description
The shapes will check if there are 20 members in G20 event in DBpedia entity.

SHACL pattern
Population Completeness - POC1

SHACL pattern description
A population is complete whenever the number of its members equals COUNT. The use of sh:inversePath makes it possible to count the subjects of TYPE-PROPERTY.

SHACL shapes
Press Shift + Tab for indentation

```
ex:G20NationsShape
  a sh:NodeShape;
  sh:targetNode dbc:G20_nations;
  sh:property [ a sh:PropertyShape;
    sh:path [
      sh:inversePath dct:subject ];
    sh:minCount 20 ].
```

Cancel **Create**

Figure 1 Form to create completeness pattern instance



SoCK Home Patterns Instances Use Cases

G20 Nations (DBpedia)

Description
The shapes will check if there are 20 members in G20 event in DBpedia entity.

Creator
anonymous

Date Created
Wednesday June, 22nd 2022

SHACL Pattern
Population Completeness - POC1

SHACL Shapes

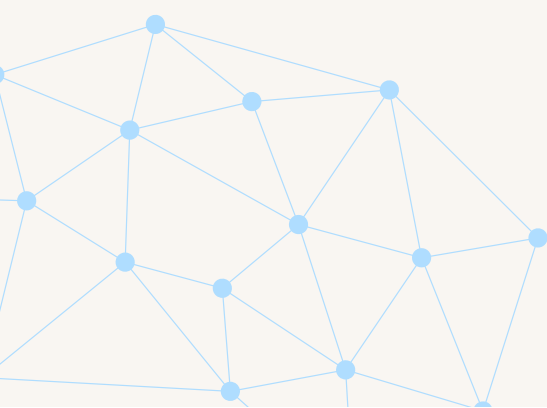
```
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix wdt: <http://www.wikidata.org/prop/direct/> .

ex:G20NationsShape
  a sh:NodeShape;
  sh:targetNode dbc:G20_nations;
  sh:property [ a sh:PropertyShape;
    sh:path [
      sh:inversePath dct:subject ];
    sh:minCount 20 ].
```

[Copy](#) [Download as TTL](#)

Figure 2 Instance pattern for G20 nations

Through the SoCK website, Lisa looks for a completeness pattern matching the case she is facing. Finally, Lisa found that the completeness pattern which fits her case is the **population completeness** pattern. Through this pattern, she creates an instance that will check whether the dbc:G20_nations entity has 20 members including into G20 event entity, as seen in Figure 1. After completing the instance, Lisa can check the result by searching for the name of that instance. Then, Lisa can view the instance in Figure 2 and download the SHACL shape to use as validation against the dbc:G20_nations entity on DBpedia.




#2

American Films

Meet Bobby, a freelancer who recently landed a film website development project. In particular, all films shown on the website are from the United States of America. At first, Bobby is confused about where he can get the data. Eventually, Bobby discovers that the Wikidata knowledge graph has the movie data he needs. He has the idea to show the film's title along with a brief description or synopsis. Then, he must check if each entity from a movie class on Wikidata has a label and a description.



Using the SoCK website, Bobby could find a pattern of completeness that fits his case: the **label and description completeness** pattern. Through its pattern, he creates an instance that checks whether the entities from the "American Film" class have a label and description properties in English according to their country of origin. In the process of creating the instance, Bobby customized the target section so that it works into the structure of the Wikidata. The result of creating an instance by him can be seen in Figure 3.

 **SoCK** Home Patterns **Instances** Use Cases

Label Description Completeness on American Films – Wikidata

Description
The SHACL shape will check the entities of American films have a label and description property in English.


Creator
anonymous

Date Created
Monday June, 13th 2022

SHACL Pattern
Label & Description Completeness - LDC01

SHACL Shapes

```
ex:AmericanFilmLabelDescriptionShapes
  a sh:NodeShape ;
  sh:target [
    a ex:WikidataCustomTarget ;
    ex:instanceOf wd:Q11424 ;
    ex:countryOfOrigin wd:Q30 ;
  ] ;
  sh:property [
    sh:path rdfs:label ;
    sh:qualifiedMinCount 1 ;
    sh:qualifiedValueShape [
      sh:languageIn ("en") ;
    ] ;
  ] ;
```

 Copy


 Download as TTL

Figure 3 Instance pattern for American films label and description


#3

Number of TV Series Seasons



Meet Mino, an entry-level data scientist. His boss has just assigned Mino to analyze data related to a TV series. In the initial step, he needs to do data crawling on the TV series data. One of the data needed is the number of seasons of the series. Mino finds that the Wikidata knowledge graph stores information related to TV series, one of which was the number of seasons and information from that season. However, he worries that the Wikidata knowledge graph information is incomplete. Therefore, he decides to validate the completeness of the data at first.

Mino uses the SoCK website to find out which pattern of completeness fits the problem. Mino finally gets a completeness pattern that fits his problem: the **property completeness** pattern to check for the completeness of a specific property (in this case, the number of seasons) on a particular entity (TV series). With its pattern, he successfully created an instance of property completeness for checking a number of seasons from one of his favorite TV series, Attack on Titan. The instance could check whether its entity has complete information for each season according to the actual situation.

 **SoCK**

HomePatternsInstancesUse Cases

Property Completeness on Number of "Attack on Titan" Series Seasons

Description
The SHACL shape will check if there are 4 seasons included in the entity of "Attack on Titan" series (Q22126305) in Wikidata

Creator
Jilham

Date Created
Wednesday June, 22nd 2022


SHACL Pattern
Property Completeness - PC01

SHACL Shapes

```
@prefix wd: <http://www.wikidata.org/entity/> .
@prefix wdt: <http://www.wikidata.org/prop/direct/> .

ex:AttackonTitanShape a sh:NodeShape ;
  sh:targetNode wd:Q22126305 ;
  sh:property ex:AttackonTitanNumberOfSeasonsPropertyShape .

ex:AttackonTitanNumberOfSeasonsPropertyShape a sh:PropertyShape ;
  sh:path wdt:P527 ; # has part or parts
  sh:minCount 4 .
```

 Copy


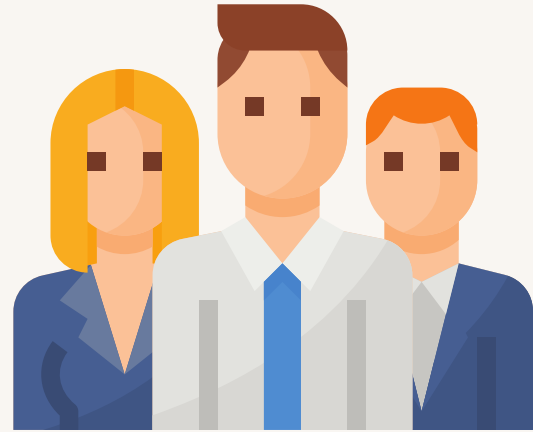
 Download as TTL

Figure 4 Instance pattern for number of TV series seasons


#4

Employee Data Checking

Meet Danny, a data engineer at a well-known fintech company. Danny wanted to do routine maintenance on the employee database. The data is stored in a graph structure. Employee data is an instance of the "Person" class and several other classes. He wants to check the quality of employee data in terms of the completeness of its required attributes.



Danny finds a SoCK website that provides a pattern for the completeness case on a knowledge graph along with its instances. From the web, Danny searched for a suitable instance and found the instance, as seen in Figure 5. The instance includes a **schema completeness** pattern which checks that every human/person entity must have the property of name, date of birth, and gender. Danny could use it to validate the employee data. However, that instance uses Schema.org vocabulary. Therefore, Danny modifies the target class and all the properties according to the use of vocabulary in the database. He adds more specific constraints, such as the properties of citizenship, date started, and division. It allows him to adjust the instance based on his use case.

[Home](#) [Patterns](#) [Instances](#) [Use Cases](#)

Person Schema

Description

The shapes will check if the instances of Person class have the property of name, date of birth, and gender using schema.org vocabulary.

Creator

Jilham

Date Created

Monday June, 27th 2022

SHACL Pattern

Schema Completeness - SC01

SHACL Shapes

```
ex:PersonShape
  a sh:NodeShape;
  sh:targetClass schema:Person ;
  sh:property ex:NamePropertyShape,
ex:DateOfBirthPropertyShape, ex:GenderPropertyShape .

ex:NamePropertyShape a sh:PropertyShape ;
  sh:path schema:name ;
  sh:minCount 1 .

ex:DateOfBirthPropertyShape a sh:PropertyShape ;
  sh:path schema:birthDate ;
```



 **Copy**  **Download as TTL**

Figure 5 Instance pattern for instances of the "Person" class