

TAKE HOME EXAM

Bölüm1: RDF & SPARQL

DotNetRDF yazılımındaki rdfEditor'ü kullanarak aşağıdaki kodu içeren bir RDF dosyası hazırlayınız ve yine DotNetRDF yazılımındaki SPARQL GUI'yi kullanarak aşağıdaki sorguları çalıştıracak SPARQL sorgularını yazınız.

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
xmlns:foaf="http://xmlns.com/foaf/0.1/">

<foaf:Kisi rdf:about="#Ali">
  <foaf:isim>Asim Kara</foaf:isim>
  <foaf:soyisim>Kara</foaf:soyisim>
  <foaf:ilkisim>Ali</foaf:ilkisim>
  <foaf:cinsiyet>erkek</foaf:cinsiyet>
</foaf:Kisi>

<foaf:Kisi rdf:about="#Oya">
  <foaf:isim>Berin Oya</foaf:isim>
  <foaf:soyisim>Oya</foaf:soyisim>
  <foaf:ilkisim>Berin</foaf:ilkisim>
  <foaf:cinsiyet>kiz</foaf:cinsiyet>
  <foaf:taniyor><foaf:Kisi rdf:ID="Ali" /></foaf:taniyor>
  <foaf:taniyor><foaf:Kisi rdf:ID="Veli" /></foaf:taniyor>
</foaf:Kisi>

<foaf:Kisi rdf:about="#Veli">
  <foaf:isim>Veli Tatli</foaf:isim>
  <foaf:taniyor><foaf:Kisi rdf:ID="Oya" /></foaf:taniyor>
  <foaf:taniyor>
    <foaf:Kisi><foaf:isim>Yoda</foaf:isim></foaf:Kisi>
  </foaf:taniyor>
</foaf:Kisi>
</rdf:RDF>
```

Sorgu1: (foaf:Kisi)'ye dahil olan bütün isimleri (foaf:isim) bulunuz.

Sorgu2: Bütün kız ve cinsiyeti tanımlanmamış kişileri bulunuz.

Sorgu3: Birbirlerini karşılıklı taniyan(foaf:taniyor) kişileri bulunuz.

Sorgu4: Ali'yi tanımayan bütün kişileri(foaf:Kisi) bulunuz.

Bölüm2: World Ontolojisinin Protege'de Yaratılması

Protégé başlatma

1. Protégé açınız ve "Create new OWL ontology" seçiniz.
2. Ontology IRI'sını <http://sws.ifi.uio.no/ont/world.owl> yapınız.
3. Ontolojinizi kaydediniz..
4. Ontology Formatını RDF/XML olarak seçiniz.

Sınıf, object özellikleri ve data özelliklerinin yaratılması

"Thing"ın 3 tane alt sınıfını yaratınız:

- Country,
- City and
- Region.

Aşağıdaki object özelliklerini and data özelliklerini yaratınız:

Object özellikleri:

- isCityInCountry,
- isCapitalOfCountry,
- isCityInRegion and
- isRegionInCountry.

Data özellikleri:

- hasPopulation,
- hasHeadOfState and
- hasLocalName.

Altsınıfları ve alt özellikleri yaratınız

- Bir ülkenin capital'inin her zaman o ülkenin bir şehri olması gerektiğini belirtiniz. Bunu yapmak için isCapitalOfCountry'nin alt özelliği olarak isCityInCountry seçiniz.
- "CityState" isimli City'nin alt sınıfı olan bir sınıf yaratınız..

Özelliklerin domain and range'lerini ayarlayınız

- isCityInCountry'nin domain'i City ve range'i Country olmalıdır.
- isRegionInCountry ve isCityinRegion'nin de domain ve range'ini ayarlayınız.

Disjoint sınıflar

- Bütün sınıflar birbirleriyle disjoint olmalıdır.

Daha fazla kısıtlama(restriciton) ekleyiniz

- Bir city'nin exactly bir country'si olmalıdır. Bunu belirtmek için Country sınıfına restriction (cardinality) ekleyiniz. Filler 1 olmalıdır.

```
isCityInCountry exactly 1 Country
```

- Bir city'nin birden fazla region'da olamayacağını belirten kısıtlama ekleyiniz. Tip: maxCardinality kullanınız.