**Query Q3:**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX dbpedia-owl: <http://dbpedia.org/ontology/>

PREFIX dbpprop: <http://dbpedia.org/property/>

PREFIX dbpedia: <http://dbpedia.org/resource/>

PREFIX sc: <http://schema.org/>

CONSTRUCT {?privateUniversitiesInCali rdf:type sc:CollegeOrUniversity .

?privateUniversitiesInCali sc:name ?universityName .} WHERE

{

{

?privateUniversitiesInCali rdf:type dbpedia-owl:University .

}

{

?privateUniversitiesInCali <http://dbpedia.org/ontology/type> <http://dbpedia.org/resource/Private\_university> .

}

UNION

{

?privateUniversitiesInCali <http://dbpedia.org/property/type> <http://dbpedia.org/resource/Private\_university> .

}

{

?privateUniversitiesInCali dbpedia-owl:state <http://dbpedia.org/resource/California>

}

UNION

{

?privateUniversitiesInCali dbpprop:state <http://dbpedia.org/resource/California>

}

?privateUniversitiesInCali rdfs:label ?universityName .

FILTER (LANGMATCHES(LANG(?universityName), 'en'))

}

GROUP BY ?universityName

**Query 4:**

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>

PREFIX schema: <http://schema.org/>

SELECT ?sub WHERE

{

?sub a schema:Person

}

**Query 10:**

INSERT DATA

{

<<http://dbpedia.org/page/University_of_Southern_California>> <<http://schema.org/alumni>> <http://dbpedia.org/resource/C.\_L.\_Max\_Nikias>

}

**Description**

1. The program queries dbpedia to retrieve the universities data to fetch the URI and their names. This data is stored in our local repository and is to be used for later.
2. I chose the CollegeOrUniversity in schema.org. Name and URI are the properties used to represent the data in triples.
3. Construct query created a view with URI of the colleges and the names of it. It created for all the private ones in LA as per the requirement.
4. Since we are using the local repository to query all the organization it is going to be a null result. Our local repository doesn’t even contain any data for it to be queried upon.
5. Loading the schema.org in the Ntriples format is successful as per the requirement.
6. This one returned null too. Despite we are querying for all the organizations (all the universities) to our local repository we receive only a null because it cannot infer that universities are organizations too.
7. The local repository is enabled with forward chaining. This will ensure that the necessary data is inferred.
8. Now we receive about 92 universities as the organizations. This is valid since Forward Chaining inference is able to infer while adding all the universities to the local repository that they are organizations too.
9. Inserting is done successfully.
10. We are now querying the local repository without and with the forward chaining inferencer.

Without: Null result is obtained. It is unable to infer that alumni are persons too.

With: After turning on the inference it is able to conclude that alumni are persons too. So when we query for all the persons in our local repository alumni can be inferred as a person and hence receiving Max Nikias as a row.