Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

DEPARTMENT OF INFORMATION TECHNOLOGY

COURSE CODE: DJ19ITL801 **DATE:** 15-02-2024

COURSE NAME: Semantic Web Technology Laboratory **CLASS:** BE IT

NAME: Rishikesh Sharma **SAP ID:** 60003200102

EXPERIMENT NO. 4

CO/LO: Query ontologies using SPARQL

AIM: To execute SPARQL Queries on RDF Dataset.

DESCRIPTION OF EXPERIMENT:

SPARQL (SPARQL Protocol and RDF Query Language) is the query language we use to shape and return linked data from a triplestore. SPARQL queries contain triple patterns, much like the data itself, which utilise the relationships to quickly navigate any linked data. This language is common for all linked data so queries can traverse across multiple RDF databases at once. The easiest way to start learning SPARQL is by example so let's start with a simple query and build from it. The following query returns every triple in the triplestore:

```
SELECT ?subject ?predicate ?object
WHERE {
  ?subject ?predicate ?object .
}
```

This query selects all triples matching the pattern:? subject ?predicate ?object which is all triples. The? indicates a variable so? example is a variable called "example". These variables match every possible entity, predicate or literal that fit the patterns in the query. In this query we are selecting every variable in the pattern so we can represent that with a * character. The only difference in results is that this query only returns 200 triples. We usually set limits in queries like these as there can be billions of matching triples! If you want the next batch, you can add OFFSET 200 on the next line. This allows you to batch process results if needed. Obviously to retrieve useful results you will need to fix variables. Fixing the predicate we can get twenty entities with their labels:

Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

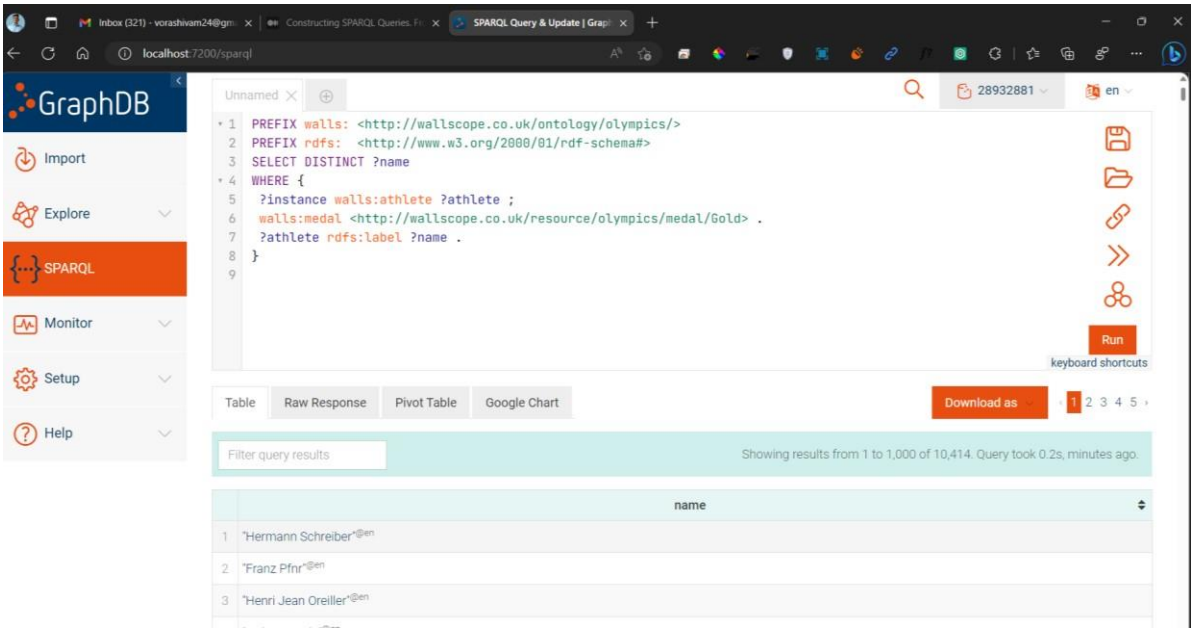
```
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
SELECT *
WHERE {
  ?entity rdfs:label ?name .
}
LIMIT 20
```

Entities are connected to their human-readable names with the predicate rdfs:label so? entity rdfs:label ?name matches any triple with the predicate rdfs:label and returns ?entity and ?name. We have defined a prefix at the top of the query so that we can make the query more readable. Removing the top line and replacing rdfs:label with <http://www.w3.org/2000/01/rdfschema#label> will return the same results. In larger queries you will notice the benefits of prefixing as the same prefix can be used multiple times in one query.

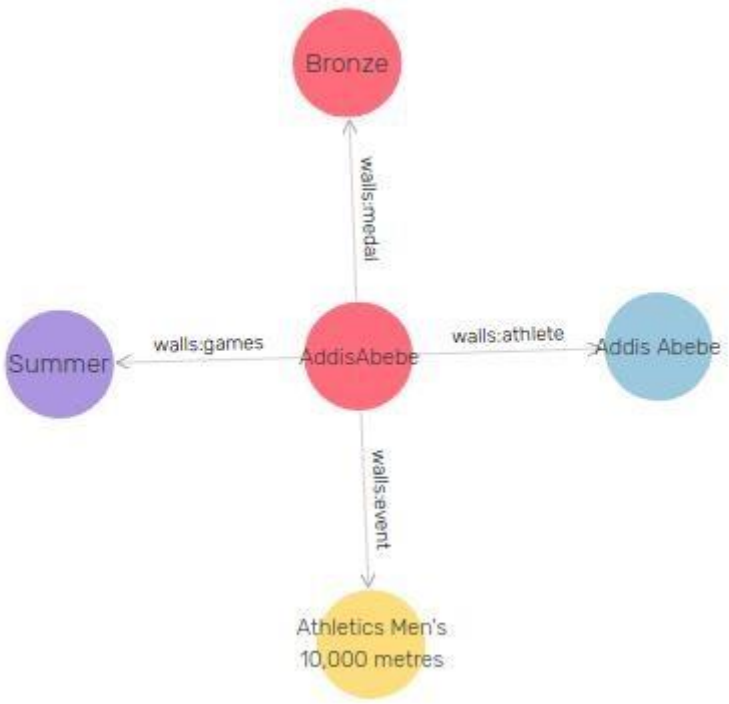
SOFTWARE/TOOLS USED: Graph DB

OBSERVATION:

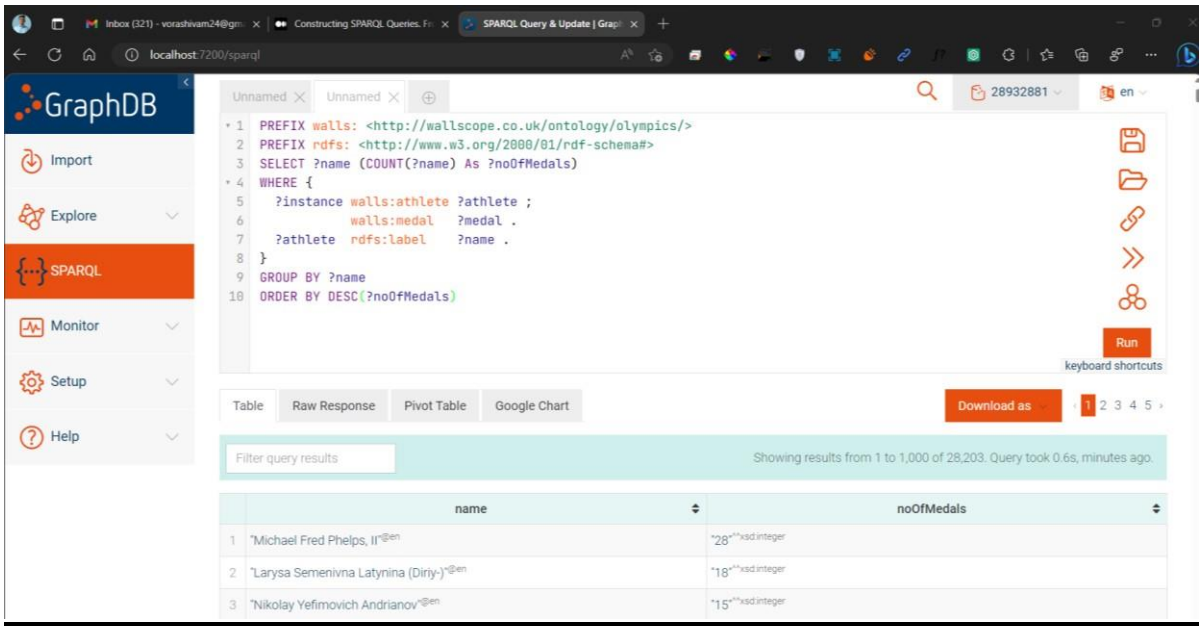
Query 1



Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

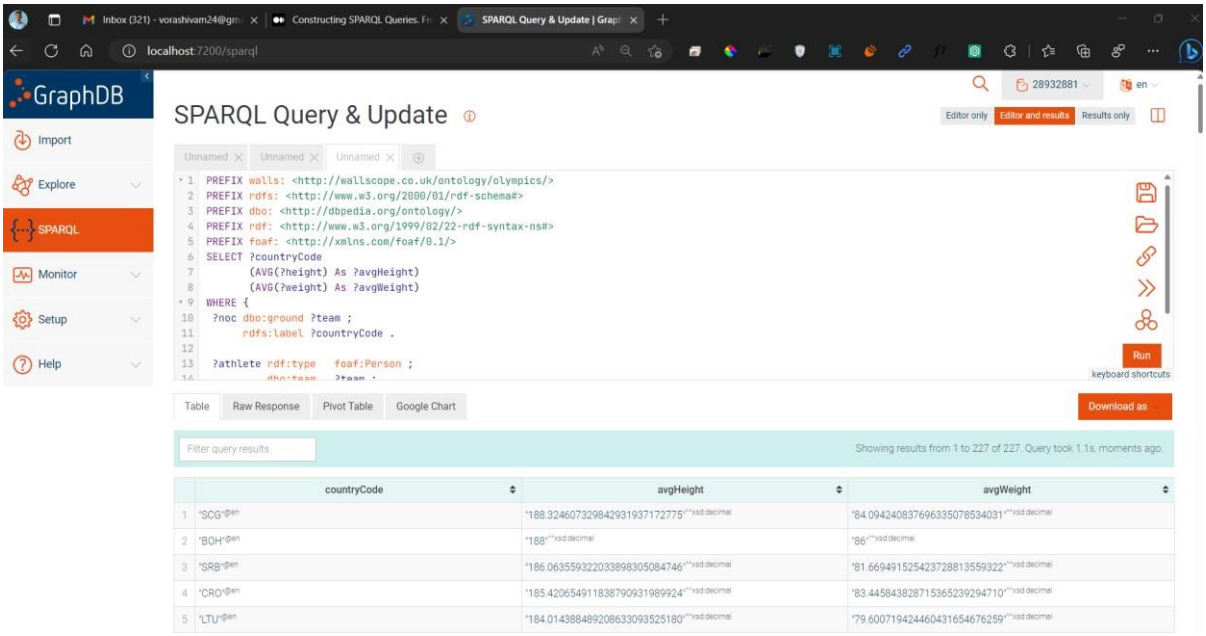


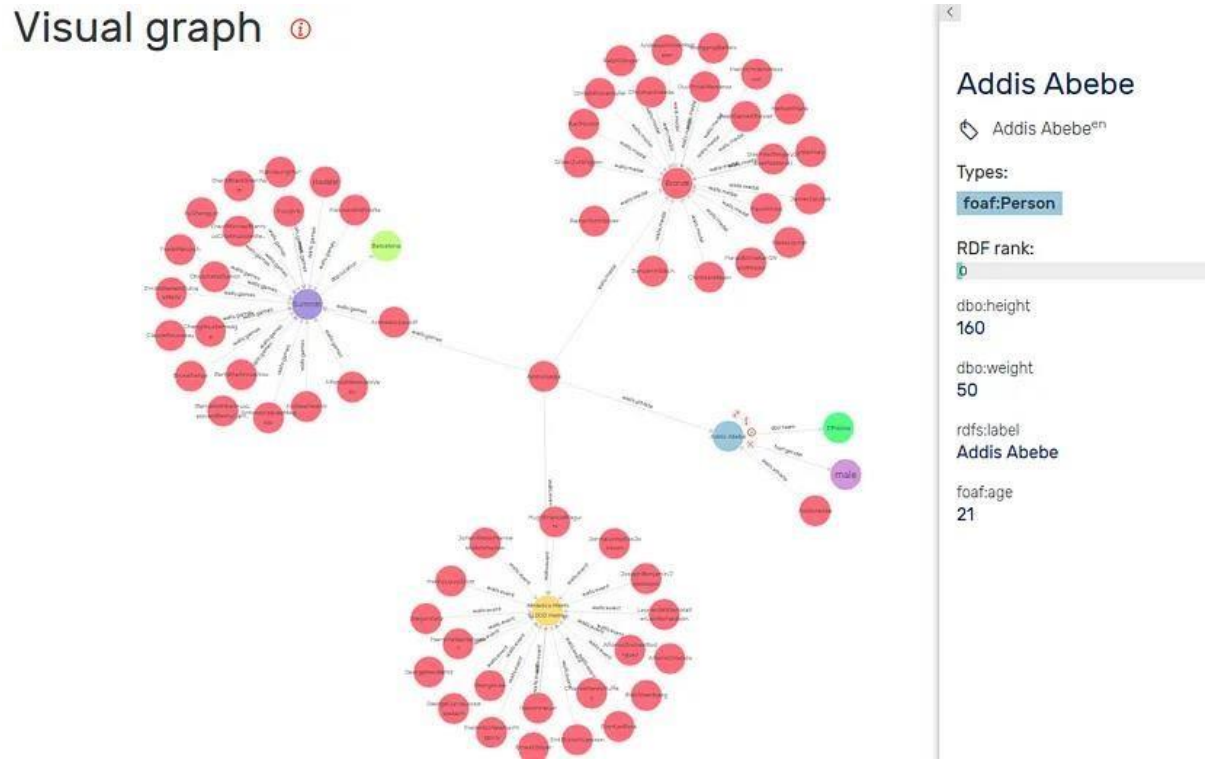
Query 2



Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

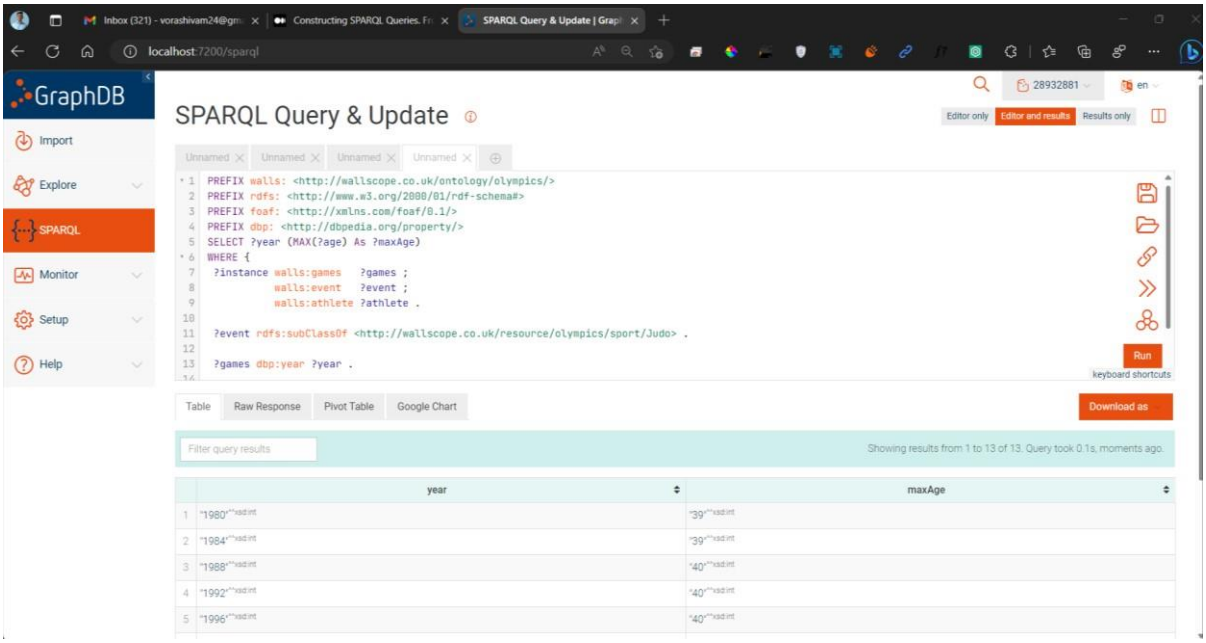
Query 3



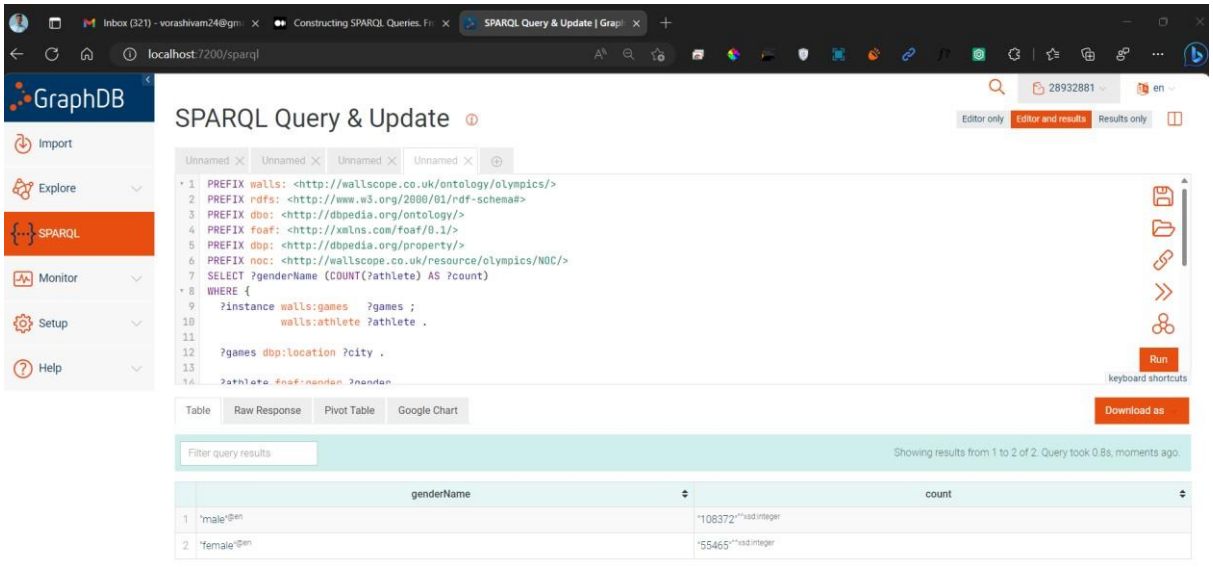


Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Query 4

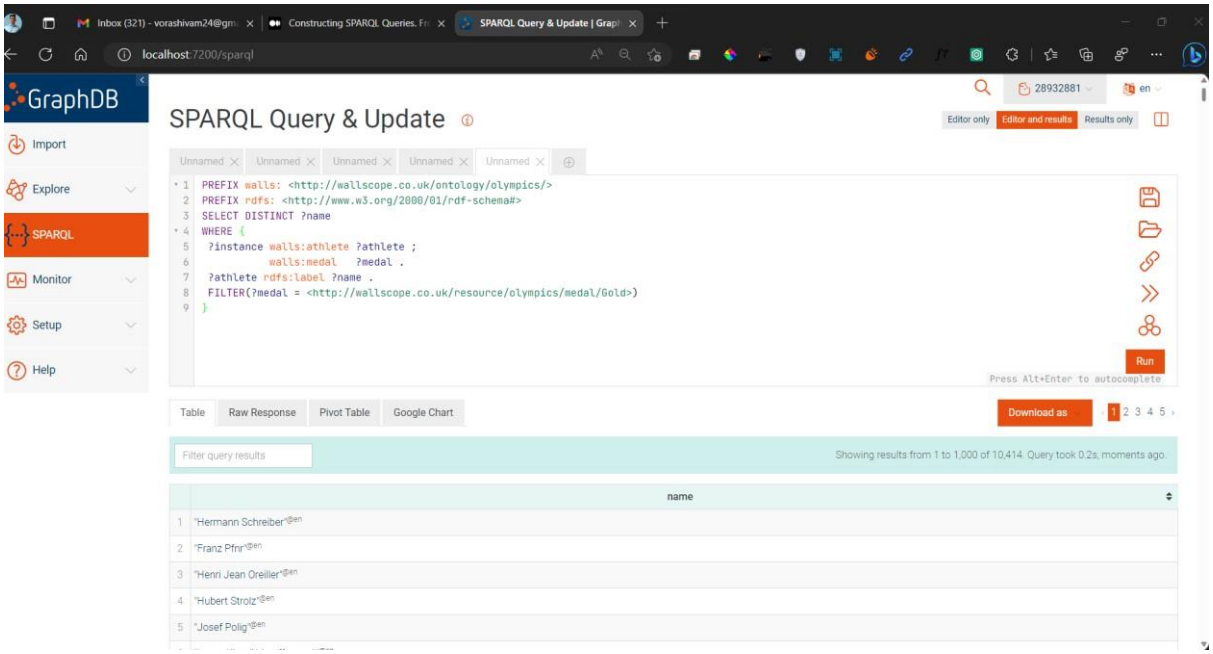


Query 5

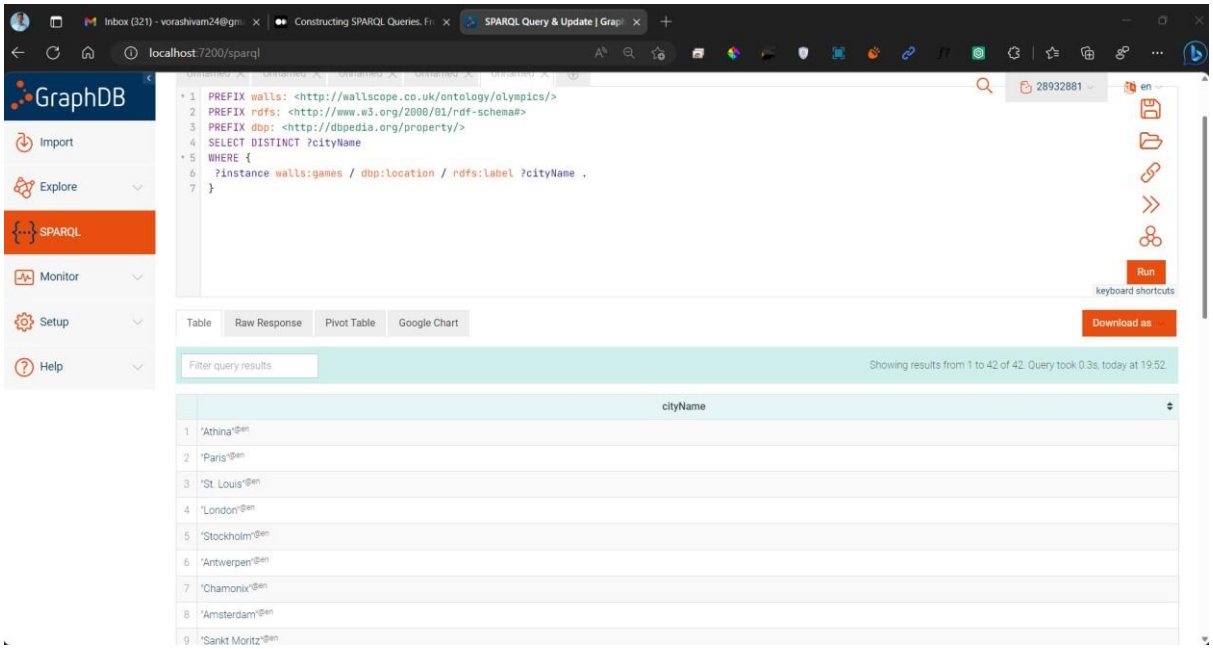


Shri Vile Parle Kelavani Mandal's
DWARKADAS J. SANGHVI COLLEGE OF ENGINEERING
(Autonomous College Affiliated to the University of Mumbai)
NAAC Accredited with "A" Grade (CGPA : 3.18)

Query 6



Query 7



CONCLUSION:

We have thus used GraphDB to execute SPARQL Queries on RDF Dataset

REFERENCES:

[1] https://medium.com/wallscope/constructing-sparql-queries-ca63b8b9ac02

[2] https://www.w3.org/TR/rdf-sparql-query/