GraphDB: Installation, Import, Queries, and Reasoning

Installation

To install and run GraphDB locally:

1. Download GraphDB from `https://graphdb.ontotext.com/` (In this project, GraphDB 11.0 for Windows was used.)
2. Use the GraphDB Desktop version.
3. Go to `Setup → License → Set new license`.
4. Paste the license key received via email and click Register.

Importing and Exporting Data

1. Create a repository by navigating to `Setup → Repositories → Create new repository`.
2. A Python script `demo.py` was developed to import:
   - A Turtle (.ttl) file with RDF triples
   - A CSV file from: `https://data.dws.informatik.uni-mannheim.de/structureddata/2024-12/quads/classspecific/AdministrativeArea/`
3. Both datasets are imported using HTTP POST requests to the GraphDB `/statements` endpoint.

SPARQL Queries

The Python program executes several SPARQL queries on the loaded data, including, retrieving all foaf:Person entities with names, extracting administrative areas with population greater than a threshold, verifying inferred triples using reasoning.
Results are exported as .csv files for further analysis..

Testing Reasoning

To evaluate GraphDB's reasoning capabilities:

1. Create a new repository with reasoning disabled (`ruleset:none`).
2. Add the following RDF triples to `example.ttl`:

```
ex:Human a rdfs:Class .
foaf:Person rdfs:subClassOf ex:Human .
ex:Alice a foaf:Person ;
     foaf:name "Alice" .
```

3. Execute the following SPARQL query:

```
PREFIX ex: <http://example.org/>
SELECT ?s
WHERE {
?s a ex:Human .
}
```

4. Compare the results from the repository with reasoning enabled vs. the one without. In the reasoning-enabled repository, `ex:Alice` is inferred to be of type `ex:Human`.

Benchmarking

Each import and query operation is benchmarked using the following metrics:

Metric	**Tool Used**
Execution Time	`time.perf_counter()`
CPU Usage	`psutil.cpu_percent()`
Memory Usage	`psutil.virtual_memory()`

Table 1: Benchmark Metrics and Tools Used

Benchmarks are recorded per phase and printed to the console for runtime evaluation.