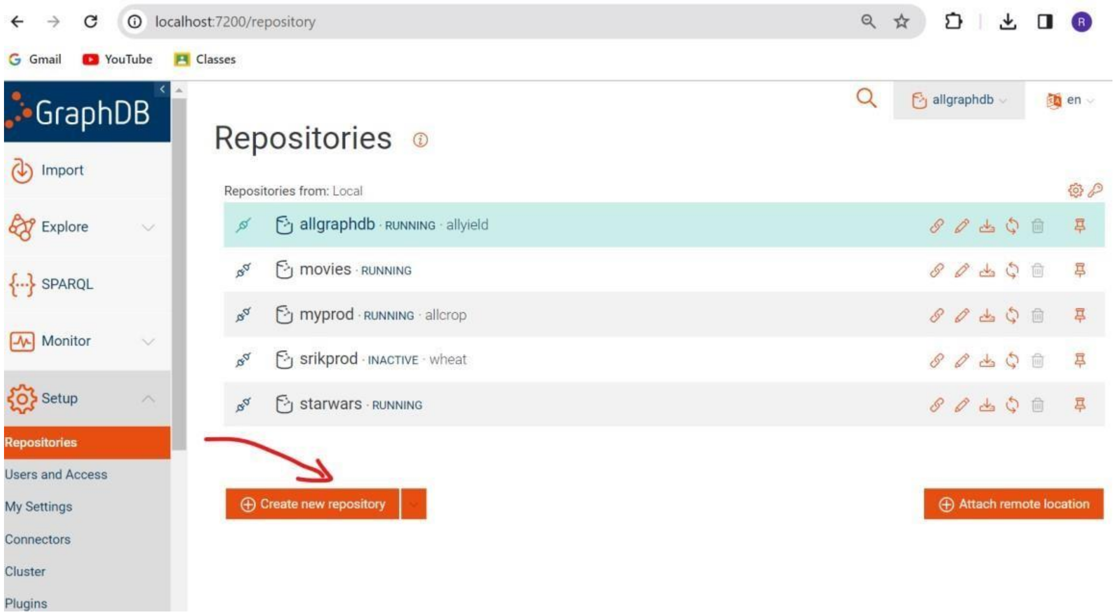
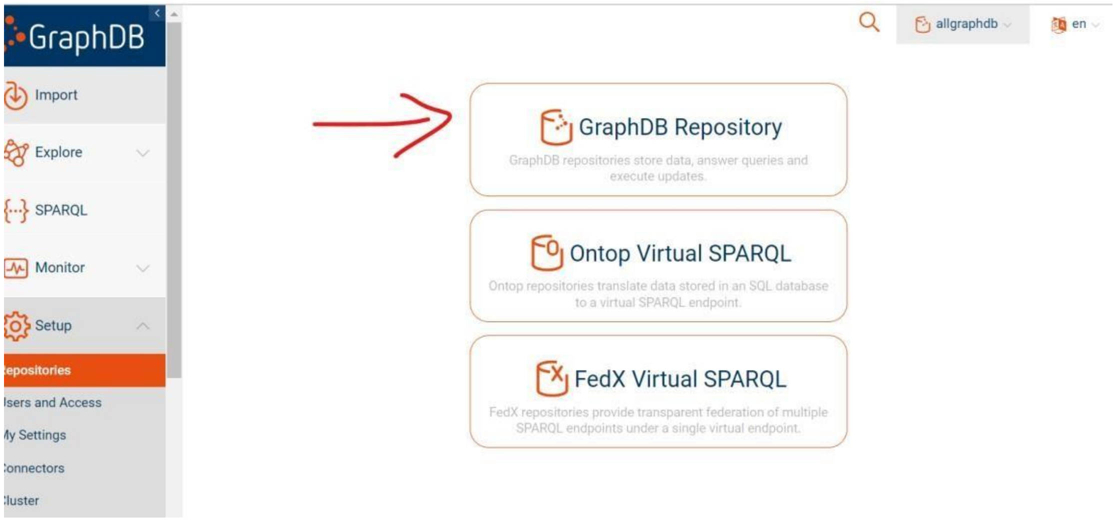
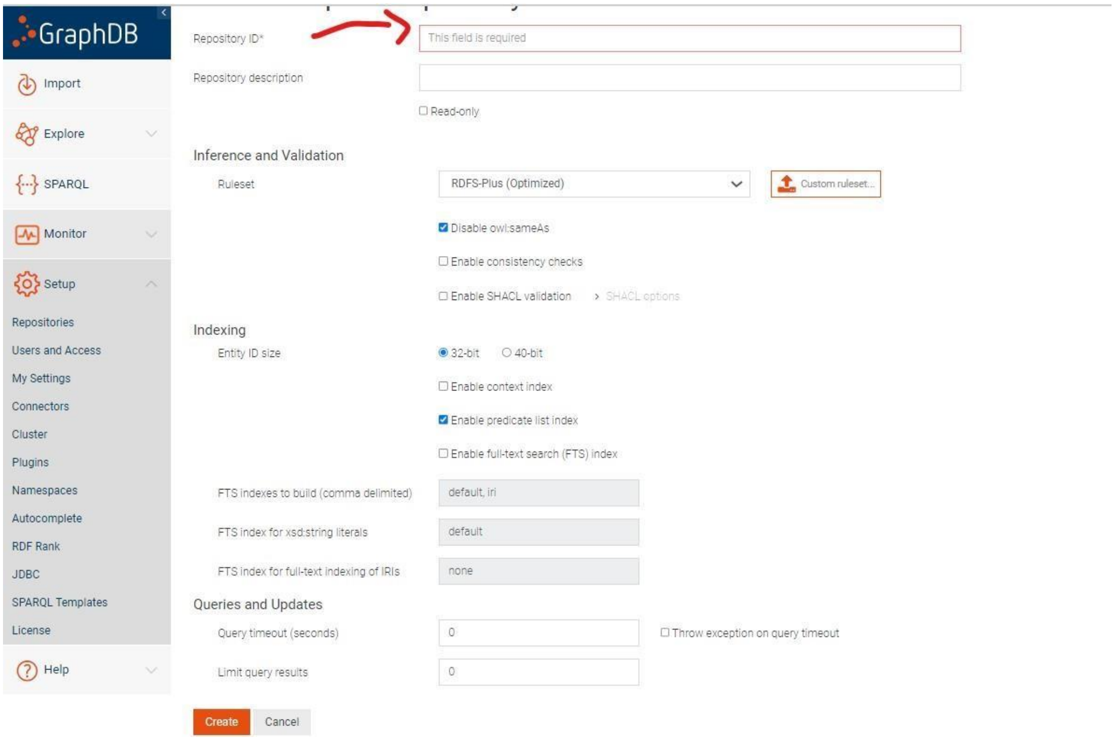
Instructions for running Graph DB instance:

1. https://download.ontotext.com/owlim/13956f1e-3766-11ee-910e-42843b1b6b38/graphdb-10.3.1-dist.zip
2. Download the above graphdb zip file and extract it .
3. Then install the graphdb desktop version and upon successful installation open it and then in the settings tab assign the port value to 7200.
4. Then open graphdb workbench so that it connects to local host server running on port 7200.
5. Convert all the csv files into one combined rdf file using python code. (make necessary changes to path in python code before running).

In Setup, create a new repository for our graph relations to show.




GraphDB
allgraphdb
en
Import
Explore
SPARQL
Monitor
Setup
epositories
lsers and Access
ly Settings
onnectors
luster
GraphDB Repository
GraphDB repositories store data, answer queries and execute updates.
Ontop Virtual SPARQL
Ontop repositories translate data stored in an SQL database to a virtual SPARQL endpoint.
FedX Virtual SPARQL
FedX repositories provide transparent federation of multiple SPARQL endpoints under a single virtual endpoint.



Repository ID* This field is required

Repository description

□ Read-only

Inference and Validation

Ruleset: RDFS-Plus (Optimized) | Custom ruleset...

☑ Disable owl:sameAs

□ Enable consistency checks

□ Enable SHACL validation › SHACL options

Indexing

Entity ID size: ● 32-bit ○ 40-bit

□ Enable context index

☑ Enable predicate list index

□ Enable full-text search (FTS) index

FTS indexes to build (comma delimited): default, iri

FTS index for xsd:string literals: default

FTS index for full-text indexing of IRIs: none

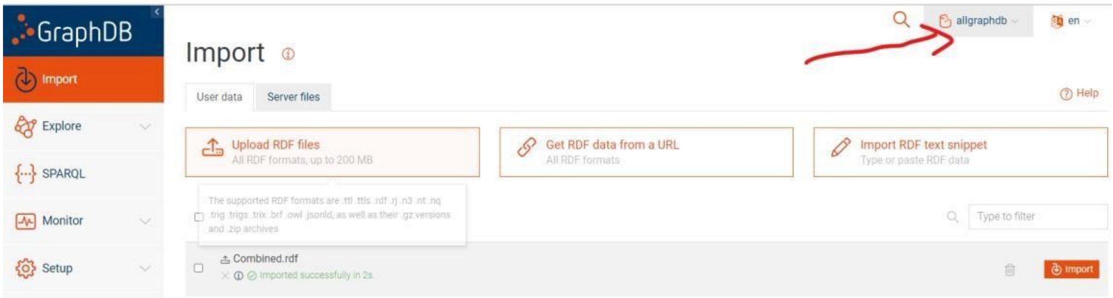
Queries and Updates

Query timeout (seconds): 0 □ Throw exception on query timeout

Limit query results: 0

Create Cancel

Open that repository on top right corner in dropdown.



Import

Upload RDF files
All RDF formats, up to 200 MB

Get RDF data from a URL
All RDF formats

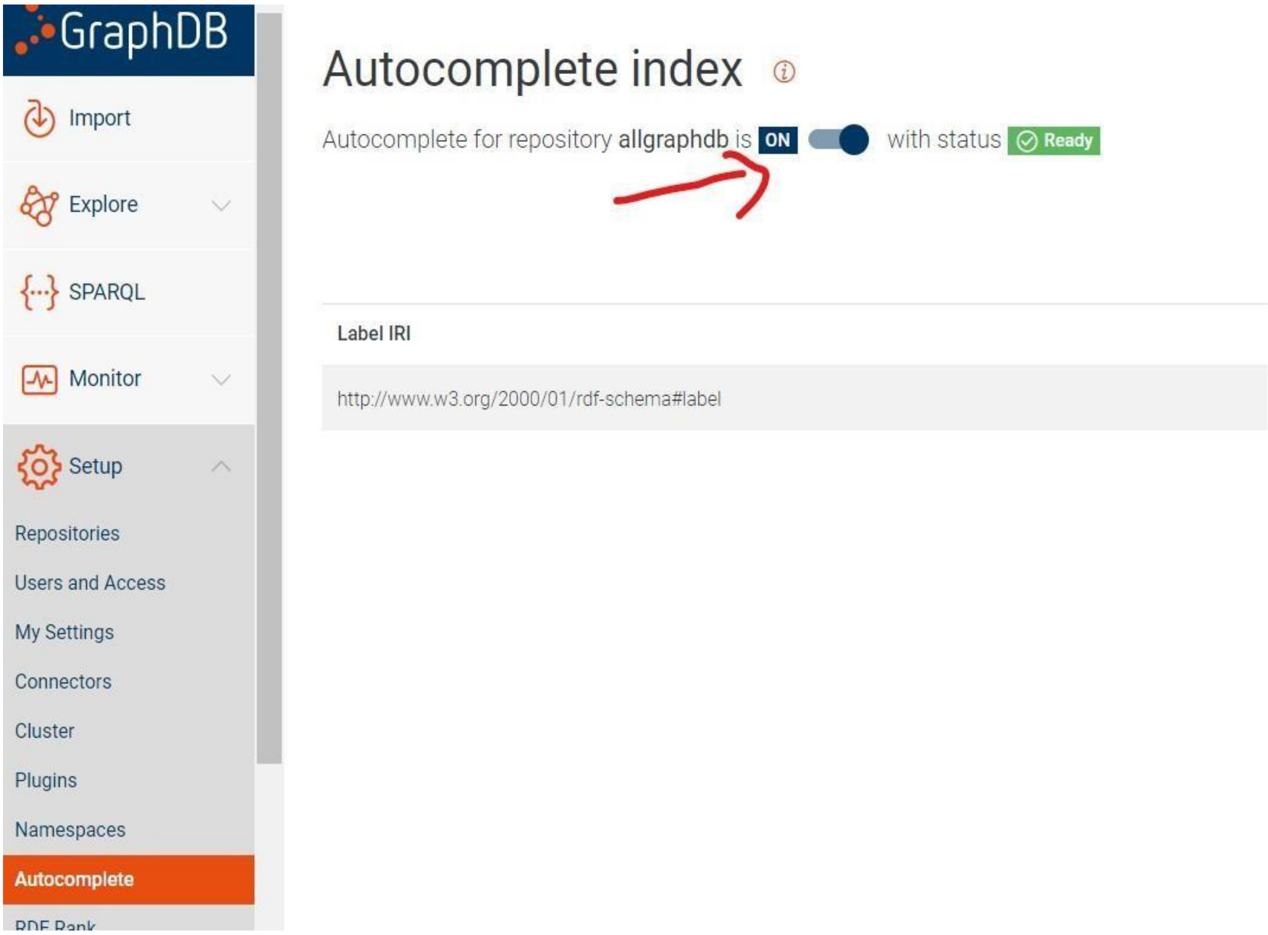
Import RDF text snippet
Type or paste RDF data

The supported RDF formats are .ttl .ttls .rdf .rj .n3 .nt .nq .trig .trigs .trix .brf .owl .jsonld, as well as their .gz versions and .zip archives

Type to filter

□ Combined.rdf
Imported successfully in 2s
Import

Make sure in setup that autocomplete is turned on.



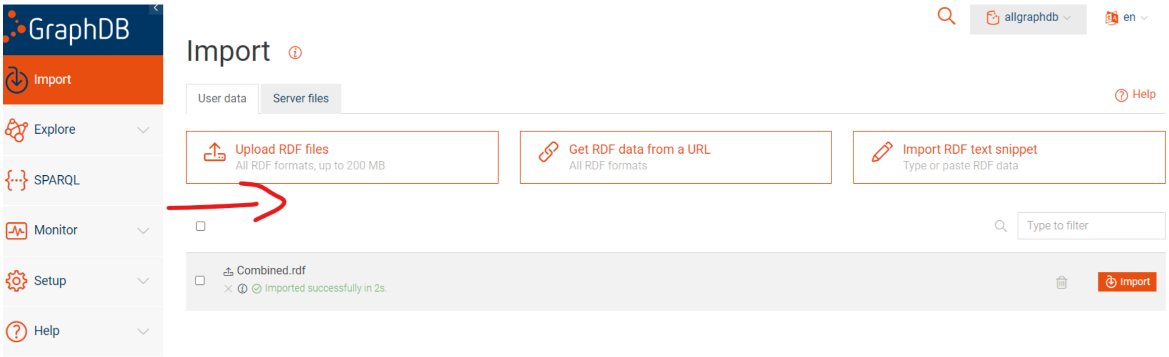
Autocomplete index

Autocomplete for repository **allgraphdb** is ON with status Ready

Label IRI

http://www.w3.org/2000/01/rdf-schema#label

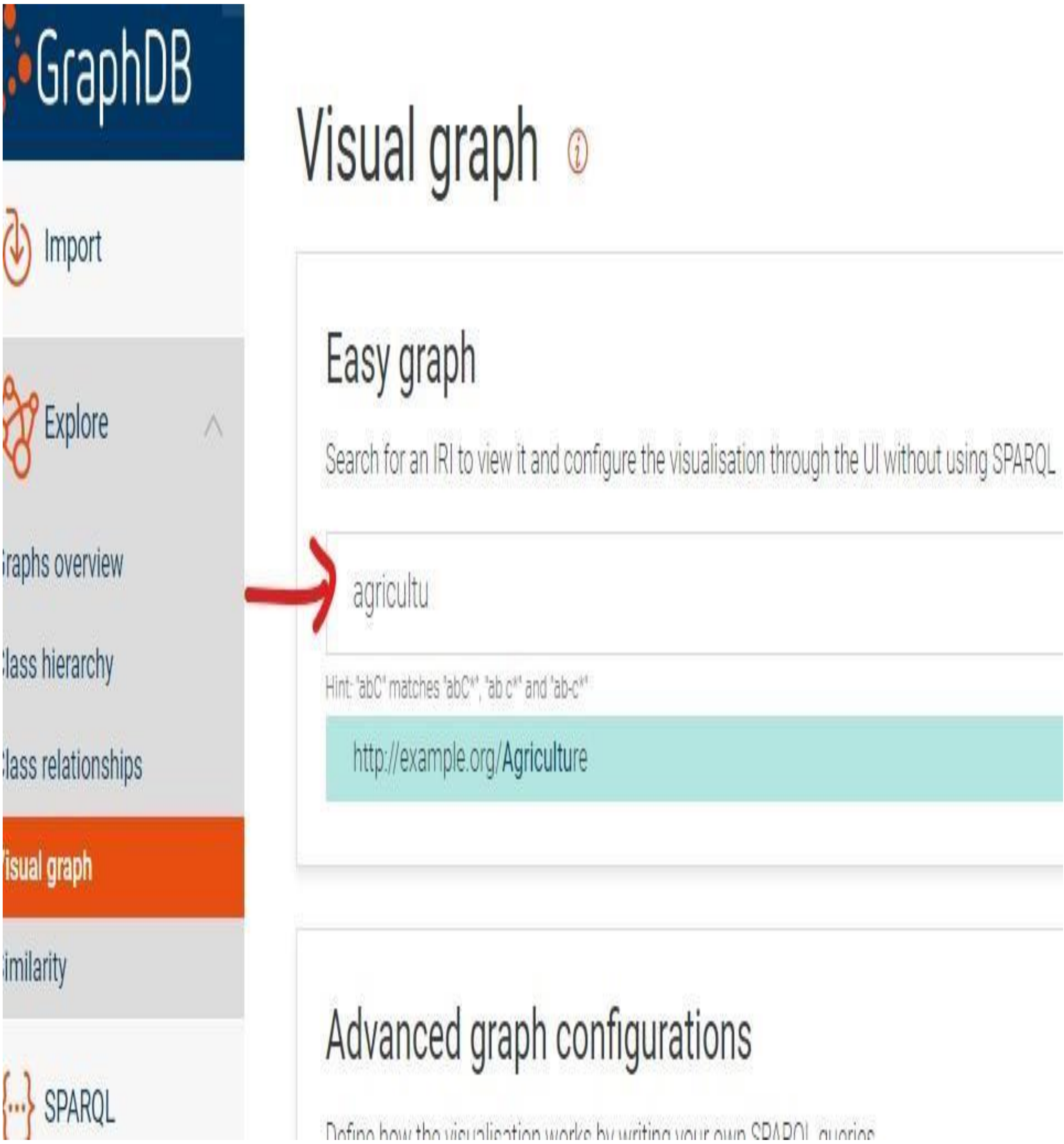
Go to import and upload the rdf file and then import.



Import

□ Combined.rdf
Imported successfully in 2s.

Open the visual graph and in easy graph search for agriculture and open it as it is the main node.



Visual graph

Easy graph

Search for an IRI to view it and configure the visualisation through the UI without using SPARQL

agricultu

Hint: 'abC' matches 'abC*', 'ab c*' and 'ab-c*'

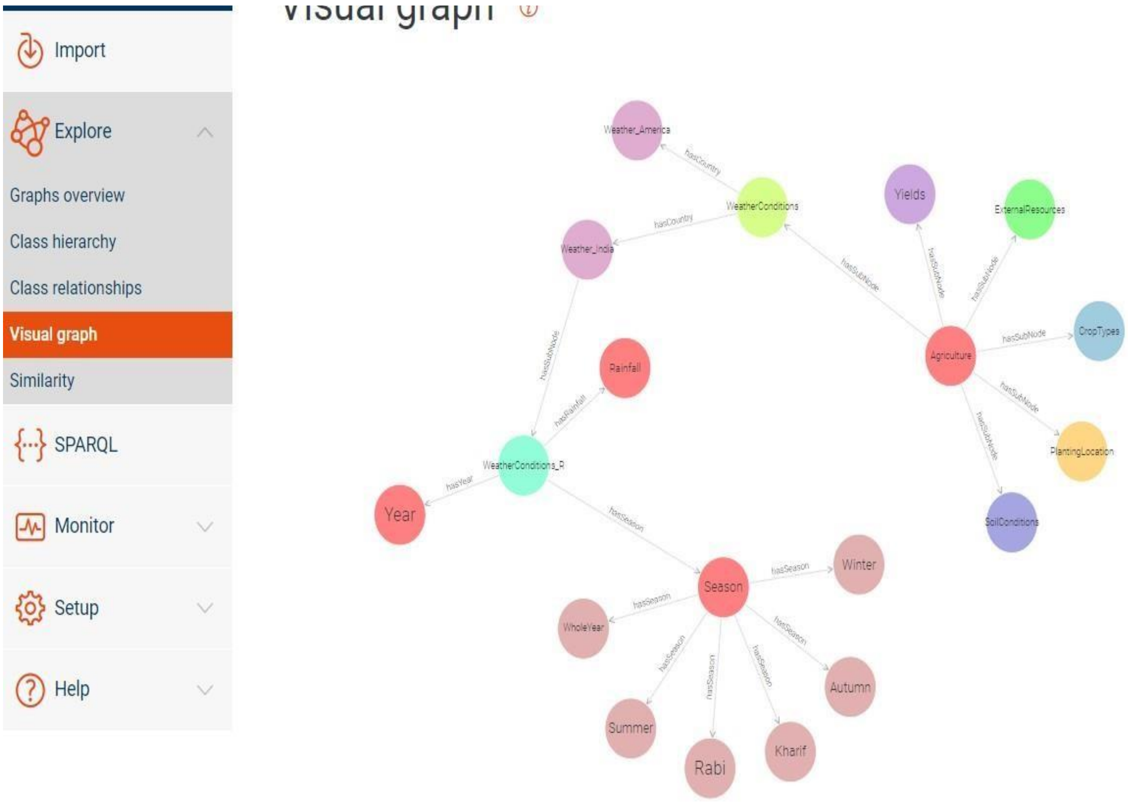
http://example.org/Agriculture

Advanced graph configurations

Define how the visualisation works by writing your own SPARQL queries

It consists of majorly 6 nodes. Then in each node there is Indian data, based on climate and soil characteristics (i.e, Revanth's data) and American data and again American data has 2 subnodes in which one belongs to Srikar data, based on weather and other belongs to Aryan data, based on amount of nitrogen.

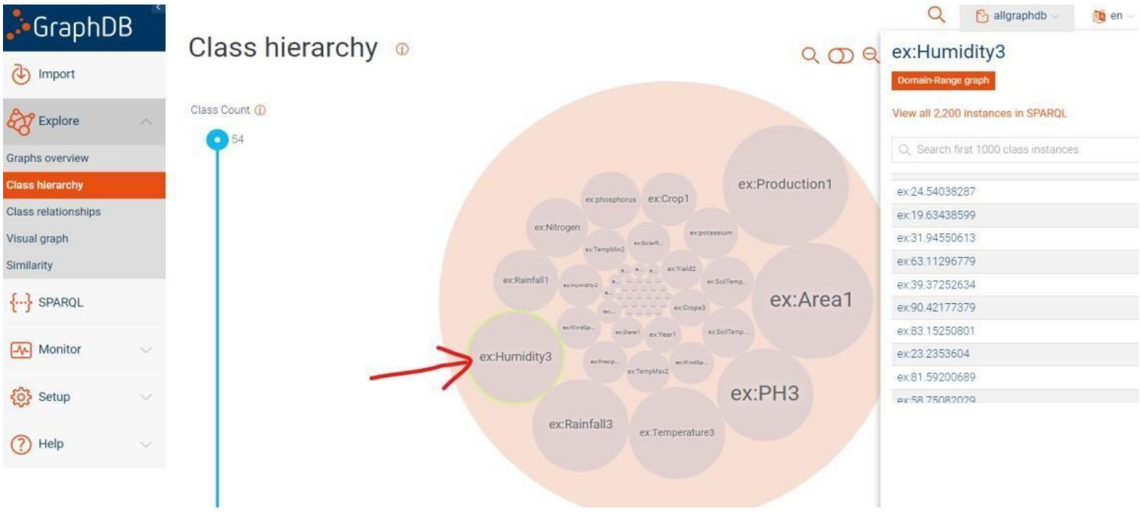
Example: WeatherConditions_R is Revanth data
WeatherConditions_S is Srikar data
WeatherConditions_A is Aryan data
Similarly for remaining fields it was mentioned like this.



- Explore all the nodes so that it contains all the data related to it.
- In easy graph as the limit is 1000, it shows only 1000 entries.
- Can check the count using countgraphdb python file. As graph db takes the unique data, it shows the data as one even when they are repeated.
- All 3 students data was taken and kept into related fields in graph db.

In explore, go to class hierarchy to check whether all the nodes and it’s data are present.

Numbered 1 like Area 1, Production 1… represents Revanth data.
Numbered 2 like Yield 2, Soil Temp min 2… represents Srikar data.
Numbered 3 like PH3, Humidity3…. represents Aryan data.



ex:24.54038287
ex:19.63438599
ex:31.94550613
ex:63.11296779
ex:39.37252634
ex:90.42177379
ex:83.15250801
ex:23.2353604
ex:81.59200689
ex:58.75082029