

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΗΜΜΥ

ΕΞΑΜΗΝΙΑΙΑ ΕΡΓΑΣΙΑ
ΣΤΟ ΜΑΘΗΜΑ
ΣΥΣΤΗΜΑΤΑ ΚΑΙ ΤΕΧΝΟΛΟΓΙΕΣ ΓΝΩΣΗΣ

ΚΟΝΤΟΓΙΑΝΝΗΣ ΑΝΔΡΕΑΣ
03115187

***ΑΝΑΠΤΥΞΗ ΒΑΣΗΣ ΓΝΩΣΗΣ ΜΕ ΤΗΝ ΧΡΗΣΗ ΤΟΥ VIRTUOSO ΚΑΙ
ΤΟΥ PROTEGE ΚΑΙ ΕΚΤΕΛΕΣΗ ΕΠΙΛΕΓΜΕΝΩΝ SPARQL
ΕΡΩΤΗΜΑΤΩΝ***

ΑΥΓΟΥΣΤΟΣ 2020

1.1 Περιεχόμενα

1.
 - 1.1 Περιεχόμενα
 - 1.2 Εισαγωγή
2.
 - 2.1 Ορισμός της οντολογίας OWL2
 - 2.1.1 Class hierarchy
 - 2.1.2 Object properties
 - 2.1.3 Data properties
 - 2.2 Παραδοχές
3.
 - 3.1 Εξόρυξη δεδομένων και κατασκευή των RDF τριάδων
 - 3.2 Περιγραφή RDF Τριάδων
 - 3.2.1 Περιγραφή των classes
 - 3.2.2 Περιγραφή των RDF Object properties
 - 3.2.3 Περιγραφή των RDF Data properties
4.
 - 4.1 Δημιουργία της βάσης γνώσης στο Virtuoso και εκτέλεση των SPARQL ερωτημάτων
 - 4.2 Ερωτήματα SPARQL για τα υπόλοιπα μέσα μεταφοράς
- 5
 - 5.1 Συζήτηση
 - 5.2 References

1.2 Εισαγωγή

Στην παρούσα εργασία θα αναπτύξουμε μία σημασιολογική βάση γνώσης με θέμα το συγκοινωνιακό δίκτυο μέσων μεταφοράς. Για την δημιουργία της βάσης, θα αναπτύξουμε μια οντολογία σε μορφή OWL2, σχεδιασμένη με τέτοιο τρόπο ώστε να καλύπτει τις ανάγκες ενός τέτοιου δικτύου. Στην συνέχεια, θα παράξουμε και θα εισάγουμε τριάδες RDF σε μορφή turtle, όπως θα δούμε αρκετά αναλυτικά σε επόμενες ενότητες, και θα αναπτύξουμε ερωτήματα SPARQL, προκειμένου να δώσουμε ορισμένες λειτουργικότητες στο σύστημα. Για την κατασκευή της οντολογίας, θα δώσουμε σημασία στην περιγραφή των μέσων μεταφοράς και των δρομολογίων τους, δίνοντας την δυνατότητα για ανάκτηση χρήσιμων πληροφοριών, όπως τις χρονικές στιγμές συγκεκριμένων δρομολογίων, τις ελάχιστες αποστάσεις από συγκεκριμένο προορισμό κ.ά.

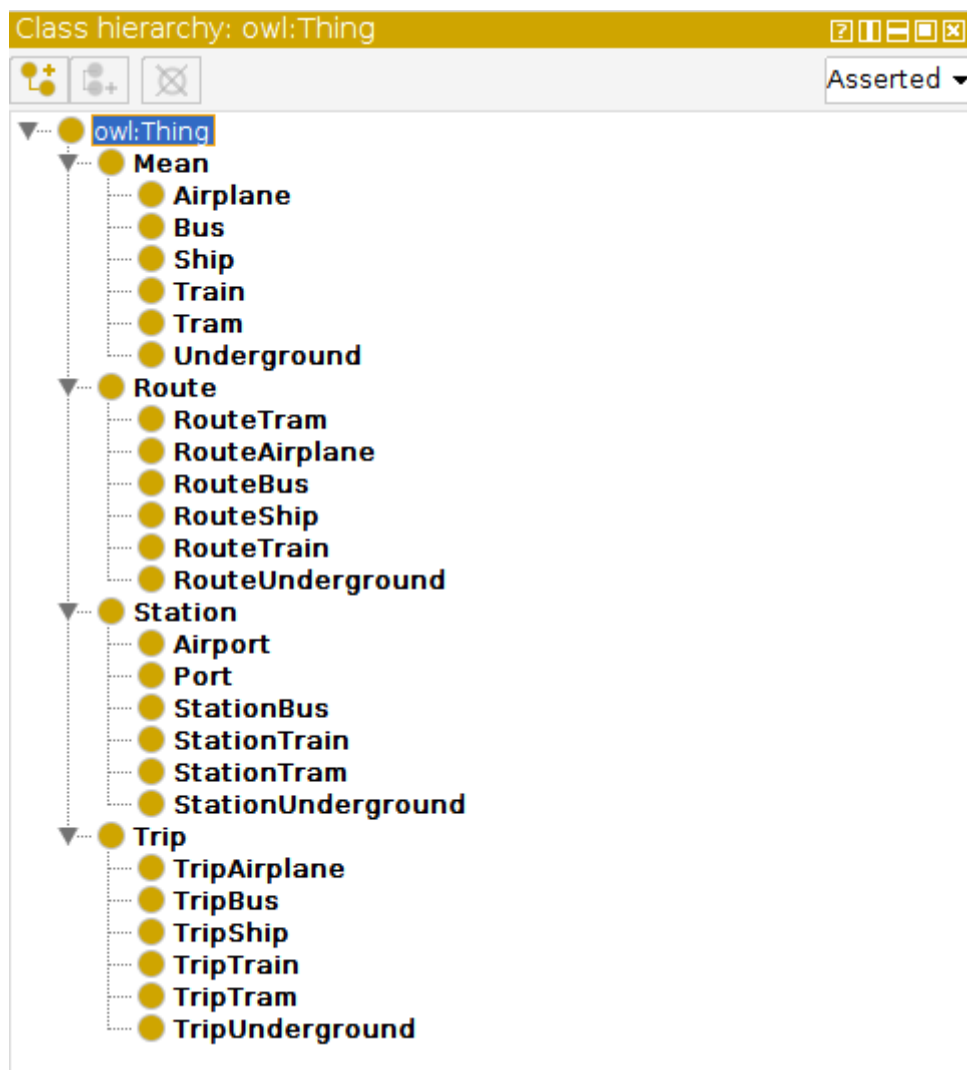
Θα χρησιμοποιήσουμε δεδομένα από το [Geodata OASA](#), τα οποία αφορούν αποκλειστικά αθηναϊκά λεωφορεία, με αποτέλεσμα το σύστημα που θα αναπτύξουμε να είναι λειτουργικό μόνο για το δίκτυο των λεωφορείων. Ωστόσο, θα δείξουμε παρακάτω και ορισμένα τετριμμένα ερωτήματα SPARQL και για άλλα μέσα μεταφοράς, όπως το τρένο, το αεροπλάνο, το πλοίο, το μετρό και το τραμ.

2.1 Ορισμός της οντολογίας OWL2

2.1.1 Class hierarchy

Στο σημείο αυτό κατασκευάζουμε την οντολογία OWL2 για το δίκτυο των μέσων μεταφοράς με την αναπαράσταση των κατάλληλων εννοιών, ρόλων και ιδιοτήτων των τύπων δεδομένων της οντολογίας. Παρακάτω, αναπαριστούμε την οντολογία όπως κατασκευάστηκε στο Protege.

Αρχικά, δείχνουμε την ιεραρχία των κλάσεων:

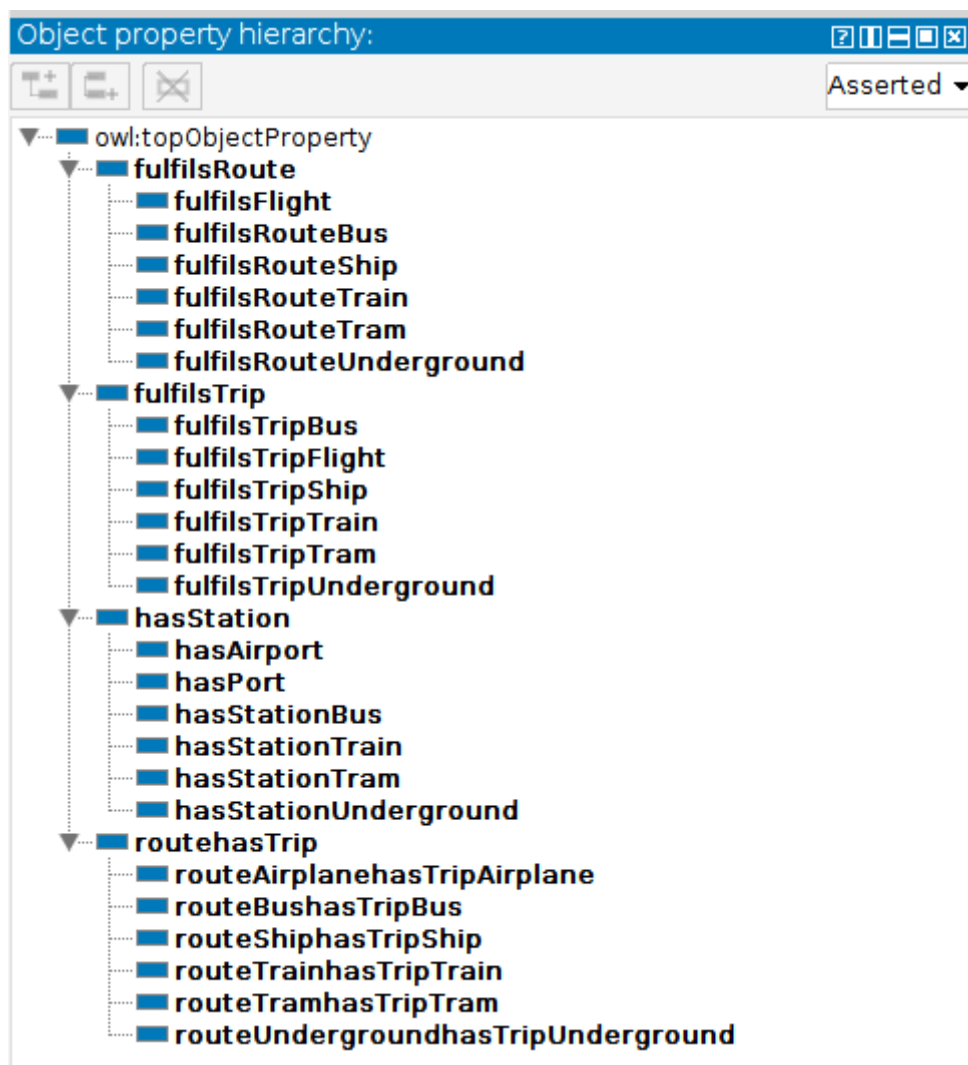


Η κλάση Mean περιέχει όλα τα μέσα μεταφοράς που αναφέρθηκαν στην Ενότητα 1.1, ενώ η κλασή Route τα ίδια τα μέσα μεταφοράς. Για παράδειγμα, η κλάση Bus περιέχει όλα τα

λεωφορεία, ενώ η Route περιέχει τα δρομολόγια. Για παράδειγμα το 410 είναι λεωφορείο (Bus), αλλά υπάρχει και το αντίστοιχο δρομολόγιο του (RouteBus). Έτσι, υπάρχει μια σαφής one-to-one αντιστοίχιση ανάμεσα σε ένα Mean και σε ένα Route. Ακόμα υπάρχει η κλάση Trip που αφορά τα επιμέρους ταξίδια που κάνει ένα δρομολόγιο, άρα και μέσο. Δηλαδή ένα Route έχει πολλά Trip - one-to-many αντιστοίχιση. Τέλος, η κλάση Station, όπως είναι εύκολο αντιληπτό, περιέχει τις στάσεις/σταθμούς των μέσων μεταφοράς. Για παράδειγμα, για το αεροπλάνο αφορά τα αεροδρόμια, για τα πλοία αφορά τα λιμάνια, ενώ για τα λεωφορεία αφορά τις στάσεις τους. Όπως θα δούμε παρακάτω, θα συνδέσουμε το Station με χρονικές στιγμές, καθώς και με γεωγραφικές συντεταγμένες, προκειμένου να είμαστε σε θέση να ανακτούμε χρήσιμες πληροφορίες που θα συζητήσουμε στα SPARQL ερωτήματα σε επόμενη ενότητα.

2.1.2 Object properties

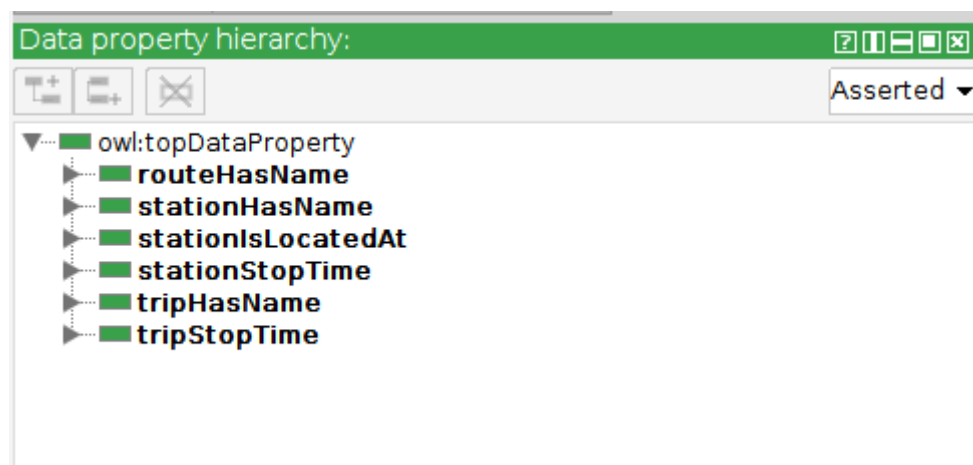
Στην συνέχεια, έχουμε τα object properties της οντολογίας μας, όπως απεικονίζονται από το Protege.



Το property `fulfilsRoute` με domain το `Mean` και range το `Route`, αποτελεί την one-to-one αντιστοίχιση των μέσων με τα δρομολόγια που αναφέραμε στην παραπάνω υποενότητα. Το property `fulfilsTrip` με domain το `Mean` και range το `Trip` είναι σχεδόν ταυτόσημο με το property `routeHasTrip` με domain το `Route` και range το `Trip`, και αφορούν την one-to-many αντιστοίχιση που αναφέραμε επίσης στην παραπάνω υποενότητα. Το property `hasStation` συνδέει ένα `Trip` με ένα `Station`.

2.1.3 Data properties

Στο σημείο αυτό θα δείξουμε τα data properties της OWL2 οντολογίας μας όπως φαίνονται στο Protege.



Σημειώνουμε πως για λόγους ευχρηστίας αναπαράστούμε τα γενικά properties, χωρίς τα subproperties τους που αφορούν τα επιμέρους μέσα μεταφοράς. Το property `routeHasName` συνδέει ένα `Route` με ένα literal string του ονόματος του. Ομοίως και το property `stationHasName` και `tripHasName`. Το property `stationIsLocatedAt` συνδέει ένα `station` με ένα `geometry`,

πχ. `"POINT(12.3830858 51.3465518)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>`
Τέλος, το property `stationStopTime` συνδέει ένα `Station` με ένα Literal string της ώρας, ενώ το `tripStopTime` ένα `Trip` με ένα Literal string της αντίστοιχης ώρας.

2.2 Παραδοχές

Αναλύοντας τα δεδομένα που είχαμε, θεωρήσαμε πως για όλα τα λεωφορεία υπάρχει η one-to-one αντιστοίχιση ανάμεσα στο `Bus` και στο `RouteBus`, με αποτέλεσμα να μας φανεί ιδανικός αυτός ο τρόπος αντίληψης του δικτύου και της ιεραρχίας των κλάσεων. Επιπλέον, θεωρήσαμε πως η στάσεις των λεωφορείων πρέπει να ανήκουν στην ίδια κλάση των `Station`

μαζί με τα αεροδρόμια ή τα λιμάνια, δεδομένου πως το σύστημα δεν θα αξιοποιεί δεδομένα εισιτηρίων ή οικονομικής φύσεως, παρά μόνο του χρόνου και των γεωγραφικών θέσεων. Δηλαδή, εάν σε κάθε αεροδρόμιο ή λιμάνι λαμβάναμε υπόψιν ότι εκεί ο πελάτης πρέπει να πάει πολύ νωρίτερα από την καθορισμένη ώρα για να βγάλει εισιτήριο, να ρυθμίσει την μετακίνησή του με βαλίτσες, άδειες, διαβατήρια κλπ. ενδεχομένως να είχαμε διαχωρήσει την υπερκλάση Station σε περισσότερες υπερκλάσεις. Τέλος, δεν λάβαμε υπόψιν τα δεδομένα για το calendar των δρομολογίων, γεγονός που αποδίδεται στο ότι αυτό δεν αποτελούσε υποχρεωτική απαίτηση για τις λειτουργικότητες του συστήματος.

Έτσι, πλέον είμαστε σε θέση να συνεχίσουμε στην δημιουργία των RDF τριάδων και στην δημιουργία της βάσης γνώσης, που θα συναντήσουμε στις αμέσως επόμενες ενότητες.

3.1 Εξόρυξη δεδομένων και κατασκευή των RDF τριάδων

Η εξόρυξη των δεδομένων γίνεται από τα αρχεία routes.txt, trips.txt, stops.txt και stop_times.txt που δίνονται από την ιστοσελίδα που αναφέραμε στην Εισαγωγή (Ενότητα 1.1). Χρησιμοποιούμε την pandas για να διαβάσουμε τα παραπάνω .csv αρχεία και την rdflib προκειμένου να κατασκευάσουμε τις RDF τριάδες. Το URI κάτω από το οποίο θα χρησιμοποιείται η βάση γνώσης μας θα είναι το <http://www.ex.org/ontologyEL15187>, το οποίο θα είναι και το Graph URI που θα χρησιμοποιούμε στο Virtuoso στην επόμενη ενότητα.

Παρακάτω, επισυνάπτουμε τον πηγαίο κώδικα σε python.

```
from rdflib import URIRef, Literal, Namespace, Graph
from rdflib.namespace import RDF

import pandas as pd

# Define the namespace for geo coordinates
GEO = Namespace("http://www.openlinksw.com/schemas/virttrdf#Geometry")

# Read csv files
dataset_routes = pd.read_csv("/home/ddaedalus/Downloads/gnws/routes.txt")
dataset_stops = pd.read_csv("stops.txt")
dataset_trips = pd.read_csv("trips.txt")
dataset_stop_times = pd.read_csv("stop_times.txt")

# Classes
bus = Namespace("http://www.ex.org/ontologyEL15187/Bus")
route_bus = Namespace("http://www.ex.org/ontologyEL15187/RouteBus")
station_bus = Namespace("http://www.ex.org/ontologyEL15187/StationBus")
trip_bus = Namespace("http://www.ex.org/ontologyEL15187/TripBus")

# Object properties
fulfils_route = Namespace("http://www.ex.org/ontologyEL15187/fulfilsRouteBus")
fulfils_trip = Namespace("http://www.ex.org/ontologyEL15187/fulfilsTripBus")
has_station = Namespace("http://www.ex.org/ontologyEL15187/hasStationBus")
route_has_trip = Namespace("http://www.ex.org/ontologyEL15187/routeBusHasTripBus")

# Data properties
station_located_at = Namespace("http://www.ex.org/ontologyEL15187/stationBusIsLocatedAt")
station_name = Namespace("http://www.ex.org/ontologyEL15187/stationBusHasName")
trip_stop_time = Namespace("http://www.ex.org/ontologyEL15187/tripBusHasStopTime")
station_stop_time = Namespace("http://www.ex.org/ontologyEL15187/stationBusStopTime")
route_name = Namespace("http://www.ex.org/ontologyEL15187/routeBusHasName")
trip_name = Namespace("http://www.ex.org/ontologyEL15187/tripBusHasName")
```

```

mygraph1 = Graph() # initialize an RDF graph for classes
mygraph2 = Graph() # initialize an RDF graph for object properties
mygraph3 = Graph() # initialize an RDF graph for data properties

# Fill the graphs
for _, row in dataset_routes.iterrows():
    mygraph1.add((URIRef(bus + "/" + row[1]), RDF.type, URIRef(bus)))
    mygraph1.add((URIRef(route_bus + "/" + row[0]), RDF.type, URIRef(route_bus)))
    mygraph2.add((URIRef(bus + "/" + row[1]), URIRef(fulfils_route), URIRef(route_bus + "/" + row[0])))
    mygraph3.add((URIRef(route_bus + "/" + row[0]), URIRef(route_name), Literal(row[2])))

for _, row in dataset_stops.iterrows():
    mygraph1.add((URIRef(station_bus + "/" + str(row[0])), RDF.type, URIRef(station_bus)))
    mygraph3.add((URIRef(station_bus + "/" + str(row[0])), URIRef(station_located_at),
        Literal("POINT("+str(row[4]) + " " + str(row[5]) + ")"), datatype = GEO)))
    mygraph3.add((URIRef(station_bus + "/" + str(row[0])), URIRef(station_name), Literal(row[2])))

for _, row in dataset_trips.iterrows():
    mygraph1.add((URIRef(trip_bus + "/" + str(row[2])), RDF.type, URIRef(trip_bus)))
    mygraph2.add(((URIRef(route_bus + "/" + row[0])), URIRef(route_has_trip), URIRef(trip_bus + "/" + row[2])))
    mygraph2.add((URIRef(bus + "/" + row[0].split("-")[0]), URIRef(fulfils_trip), URIRef(trip_bus + "/" + str(row[2]))))
    mygraph3.add((URIRef(trip_bus + "/" + row[2]), URIRef(trip_name), Literal(row[3])))

for _, row in dataset_stop_times.iterrows():
    mygraph2.add((URIRef(trip_bus + "/" + str(row[0])), URIRef(has_station), URIRef(station_bus + "/" + str(row[3]))))
    mygraph3.add((URIRef(trip_bus + "/" + row[0]), URIRef(trip_stop_time), Literal(row[1])))
    mygraph3.add((URIRef(station_bus + "/" + str(row[3])), URIRef(station_stop_time), Literal(row[1])))

mygraph1.serialize('rdf_classes.ttl', format='turtle')
mygraph2.serialize('rdf_object_properties.ttl', format='turtle')
mygraph3.serialize('rdf_data_properties.ttl', format='turtle')

```

Ο παραπάνω κώδικας παράγει τα αρχεία .ttl με τις τριάδες RDF που θα εισάγουμε στην βάση γνώσης στο Virtuoso και στην συνέχεια θα τρέξουμε τα SPARQL ερωτήματα. Θα παράξουμε τρία .ttl αρχεία, ένα για class, ένα για object properties και ένα για data properties. Αρχικά, δημιουργούμε τα namespaces των classes, object properties και data properties, καθώς και της γεωμετρίας, Geometry. Στην συνέχεια, αρχικοποιούμε τους τρεις γράφους έναν για κάθε αρχείο που θέλουμε να εξάγουμε και ακολουθούν επαναληπτικοί βρόχοι, ένας για κάθε .csv αρχείο που διαβάστηκε που δημιουργεί τις αντίστοιχες RDF τριάδες και τις αποθηκεύει στον αντίστοιχο γράφο με βάση αν είναι classes, object properties ή data properties. Στο τέλος, όταν έχει τελειώσει το γέμισμα των τριών γράφων, τους σειριοποιούμε και παράγουμε τα τρία επιθυμητά .ttl αρχεία σε μορφή turtle.

3.2 Περιγραφή RDF Τριάδων

Στο σημείο αυτό, αξίζει να δούμε την μορφή αυτών των αρχείων που παράξαμε.

3.2.1 Περιγραφή των Classes

Ξεκινάμε από το rdf_classes.ttl και περιμένουμε να δούμε τριάδες που περιέχουν μόνο το property type (a).


```

<http://www.ex.org/ontologyEL15187/Bus/021> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/022> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/024> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/025> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/026> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/027> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/032> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/035> a <http://www.ex.org/ontologyEL15187/Bus> .
<http://www.ex.org/ontologyEL15187/Bus/036> a <http://www.ex.org/ontologyEL15187/Bus> .

```

Πράγματι, το παραπάνω στιγμιότυπο περιέχει RDF τριάδες που δηλώνουν πως κάποια συγκεκριμένα λεωφορεία είναι τύπος Bus. Παρόμοια αποτελέσματα θα είχαμε και αν παίρναμε στιγμιότυπα με τα Routes, τα Stations ή τα Trips.

3.2.2 Περιγραφή των RDF Object properties

Για τις RDF Object properties τριάδες περιμένουμε να μην υπάρχουν πουθενά Literals, είτε αυτά είναι string για χρονικές στιγμές είτε για γεωγραφικές θέσεις. Τα object properties έχουν την παρακάτω μορφή:

a' uri hasProperty b' uri,
b' not a literal,
hasProperty not type(a)

Παρακάτω, δείχνουμε ένα στιγμιότυπο του αρχείου rfd_object_properties.ttl για να δούμε αν όντως ικανοποιείται η παραπάνω μορφή.

```

<http://www.ex.org/ontologyEL15187/TripBus/7463121-CALEND-832-833-Κυριακή-04> ns1:hasStationBus <http://www.ex.org/ontologyEL15187/StationBus/180015>,
<http://www.ex.org/ontologyEL15187/StationBus/180039>,
<http://www.ex.org/ontologyEL15187/StationBus/180042>,
<http://www.ex.org/ontologyEL15187/StationBus/180060>,
<http://www.ex.org/ontologyEL15187/StationBus/180061>,
<http://www.ex.org/ontologyEL15187/StationBus/180062>,
<http://www.ex.org/ontologyEL15187/StationBus/180063>,
<http://www.ex.org/ontologyEL15187/StationBus/180064>,
<http://www.ex.org/ontologyEL15187/StationBus/180065>,
<http://www.ex.org/ontologyEL15187/StationBus/180067>,
<http://www.ex.org/ontologyEL15187/StationBus/260062>,
<http://www.ex.org/ontologyEL15187/StationBus/260065>,
<http://www.ex.org/ontologyEL15187/StationBus/260066>,
<http://www.ex.org/ontologyEL15187/StationBus/260067>,
<http://www.ex.org/ontologyEL15187/StationBus/260069>,
<http://www.ex.org/ontologyEL15187/StationBus/260070>,
<http://www.ex.org/ontologyEL15187/StationBus/260071>,

```

Πράγματι, και πάλι βλέπουμε πως ικανοποιείται η επιθυμητή μορφή για το αρχείο αυτό. Σημειώνουμε πως προφανώς η ίδια μορφή ακολουθεί και καθόλη την έκταση του αρχείου.

3.2.3 Περιγραφή των RDF Data properties

Για τις RDF Data properties τριάδες περιμένουμε να υπάρχουν σε όλες Literals, είτε αυτά είναι string για χρονικές στιγμές είτε για γεωγραφικές θέσεις. Τα data properties έχουν την παρακάτω μορφή:

a' uri hasProperty b' literal

Παρακάτω, δείχνουμε ένα στιγμιότυπο του αρχείου rfd_data_properties.ttl για να δούμε αν όντως ικανοποιείται η παραπάνω μορφή.

```
<http://www.ex.org/ontologyEL15187/RouteBus/021-20> ns1:routeBusHasName "ΚΑΝΙΓΓΙΟΣ - ΓΚΥΖΗ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/022-20> ns1:routeBusHasName "Ν. ΚΥΨΕΛΗ - ΜΑΡΑΣΛΕΙΟΣ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/024-20> ns1:routeBusHasName "ΑΓ. ΑΝΑΓΕΥΡΟΙ - ΣΤ. ΑΤΤΙΚΗΣ - ΣΤ. ΛΑΡΙΣΗΣ (ΚΥΚΛΙΚΗ)" .  
<http://www.ex.org/ontologyEL15187/RouteBus/025-20> ns1:routeBusHasName "ΙΠΠΟΚΡΑΤΟΥΣ - ΠΡΟΦ. ΔΑΝΙΗΛ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/026-20> ns1:routeBusHasName "ΙΠΠΟΚΡΑΤΟΥΣ - ΒΟΤΑΝΙΚΟΣ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/027-20> ns1:routeBusHasName "ΙΠΠΟΚΡΑΤΟΥΣ - ΟΡΦΕΩΣ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/032-20> ns1:routeBusHasName "ΓΟΥΔΗ - ΜΑΡΑΣΛΕΙΟΣ (ΣΧΟΛΙΚΗ)" .  
<http://www.ex.org/ontologyEL15187/RouteBus/035-20> ns1:routeBusHasName "ΑΝΩ ΚΥΨΕΛΗ - ΠΕΤΡΑΛΩΝΑ - ΤΑΥΡΟΣ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/036-20> ns1:routeBusHasName "ΣΤ.ΚΑΤΕΧΑΚΗ - ΣΤ.ΠΑΝΟΡΜΟΥ - ΓΑΛΑΤΣΙ - ΚΥΨΕΛΗ (ΚΥΚ)" .  
<http://www.ex.org/ontologyEL15187/RouteBus/040-20> ns1:routeBusHasName "ΠΕΙΡΑΙΑΣ - ΣΥΝΤΑΓΜΑ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/046-20> ns1:routeBusHasName "ΜΟΥΣΕΙΟ - ΕΛΛΗΝΟΡΩΣΩΝ" .  
<http://www.ex.org/ontologyEL15187/RouteBus/049-20> ns1:routeBusHasName "ΠΕΙΡΑΙΑΣ - ΟΜΟΝΟΙΑ" .
```

Πράγματι, και πάλι βλέπουμε πως ικανοποιείται η επιθυμητή μορφή για το αρχείο αυτό. Σημειώνουμε πως προφανώς η ίδια μορφή ακολουθεί και καθόλη την έκταση του αρχείου.

Επομένως, σημειώνουμε πως έχουμε φτιάξει ορθώς τις RDF τριάδες στα αντίστοιχα αρχεία και το μόνο που μας μένει να τα ανεβάσουμε στην βάση και να υλοποιήσουμε τα SPARQL ερωτήματα, προκειμένου να δώσουμε στο σύστημα τις επιθυμητές λειτουργικότητες.

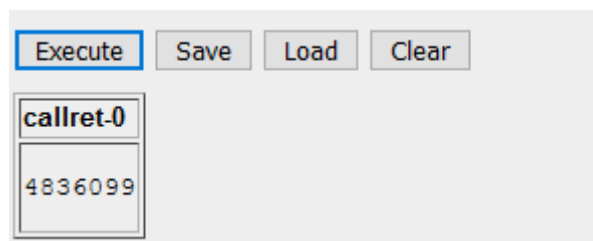
4.1 Δημιουργία της βάσης γνώσης στο Virtuoso και εκτέλεση των SPARQL ερωτημάτων

Τρέχουμε το Virtuoso στο localhost:8890/conductor και ανεβάζουμε τα παραγόμενα αρχεία με κοινό Graph URI το <http://www.ex.org/ontologyEL15187>, προκειμένου να δημιουργήσουμε την νέα μας βάση. Στο σημείο αυτό θα τρέξουμε ορισμένα ερωτήματα SPARQL και θα αναπαραστήσουμε τα αποτελέσματά τους. Τα ερωτήματα αυτά αποτελούν τις βασικές λειτουργικότητες του συστήματός μας.

Query 1

Φέρε μου τον αριθμό όλων των τριάδων που αποτελούν την βάση.

```
SELECT count(*)  
WHERE {  
    ?a    ?b    ?c  
}
```

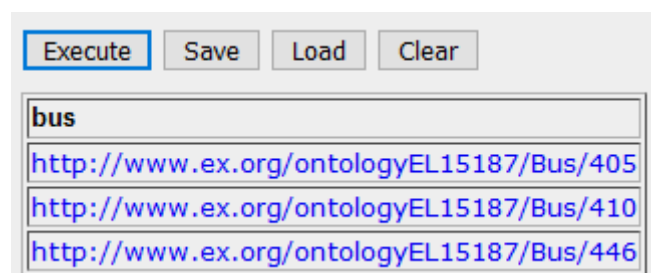


The screenshot shows the Virtuoso SPARQL query interface. At the top, there are four buttons: "Execute" (highlighted with a blue border), "Save", "Load", and "Clear". Below these buttons is a text area containing the query result. The text area has a header "callret-0" and a single row with the value "4836099".

Query 2

Φέρε μου ποια λεωφορεία περνάνε από την στάση με δοσμένο id πχ. 640046

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>  
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>  
  
SELECT distinct ?bus  
WHERE {  
    ?bus a ex:Bus.  
    ?bus ex:fulfilsTripBus ?trip.  
    ?trip ex:hasStationBus st:640046.  
}
```



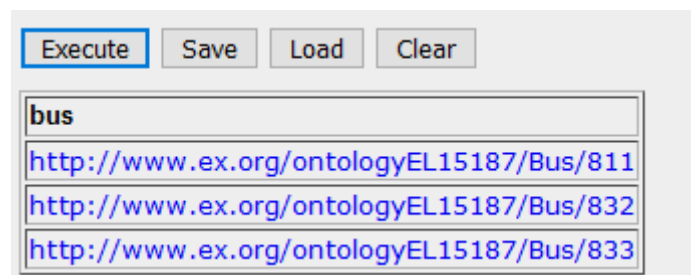
The screenshot shows the Virtuoso SPARQL query interface. At the top, there are four buttons: "Execute" (highlighted with a blue border), "Save", "Load", and "Clear". Below these buttons is a text area containing the query result. The text area has a header "bus" and three rows of results, each containing a URL: <http://www.ex.org/ontologyEL15187/Bus/405>, <http://www.ex.org/ontologyEL15187/Bus/410>, and <http://www.ex.org/ontologyEL15187/Bus/446>.

Query 3

Φέρε μου ποια λεωφορεία περνάνε από την στάση με δοσμένο όνομα πχ. Κομνηνών

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT distinct ?bus
WHERE {
    ?bus a ex:Bus.
    ?bus ex:fulfilsTripBus ?trip.
    ?trip ex:hasStationBus ?st.
    ?st ex:stationBusHasName "ΚΟΜΝΗΝΩΝ"
}
```



Query 4

Φέρε μου όλα τα λεωφορεία και τις ώρες που περνάνε από την στάση με δοσμένο id πχ. 100001

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>

SELECT ?bus ?time
WHERE {
    ?bus a ex:Bus.
    ?bus ex:fulfilsTripBus ?trip.
    ?trip ex:hasStationBus st:640046.
    ?trip ex:tripBusHasStopTime ?time.
    st:100001 ex:stationBusStopTime ?time.
}
```

Execute Save Load Clear

bus	time
http://www.ex.org/ontologyEL15187/Bus/446	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/446	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/405	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:44:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:46:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:46:00"
http://www.ex.org/ontologyEL15187/Bus/446	"05:46:00"
http://www.ex.org/ontologyEL15187/Bus/446	"05:46:00"
http://www.ex.org/ontologyEL15187/Bus/410	"05:46:00"

Προφανώς είναι πάρα πολλά, αλλά βλέπουμε πως λειτουργεί σωστά.

Query 5

Φέρε μου την τούπλα (id, geometry, distance) που αντιστοιχεί στην στάση με την μικρότερη απόσταση από μία στάση με δοσμένο το id πχ 100001

```
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?x ?point2 ?dist
WHERE {
    st:100001 ex:stationBusIsLocatedAt ?point.
    ?x ex:stationBusIsLocatedAt ?point2.
    FILTER(?x != st:100001)
    BIND(bif:st_distance(?point, ?point2) as ?dist).
}
ORDER BY ASC (?dist)
LIMIT 1
```

Execute

Save

Load

Clear

x	point2	dist
http://www.ex.org/ontologyEL15187/StationBus/250014	"POINT(38.06448458809 23.713464050503)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>	0.153003

Query 6

Φέρε όλα τα λεωφορεία που περνάνε από την στάση με τις συγκεκριμένες γεωγραφικές συντεταγμένες πχ. (38.070813012283, 23.710395367301)

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT distinct ?bus
WHERE {
    BIND("POINT(38.070813012283 23.710395367301)"^^<http://www.openlinksw.com/schemas/virttrdf#Geometry> as ?p).

    ?station ex:stationBusIsLocatedAt ?point.
    FILTER(bif:st_distance(?p, ?point) < 0.0001).

    ?tripBus ex:hasStationBus ?station.
    ?bus ex:fulfilsTripBus ?tripBus
}
```

Execute	Save	Load	Clear
bus			
http://www.ex.org/ontologyEL15187/Bus/B12			
http://www.ex.org/ontologyEL15187/Bus/713			

Query 7

Φέρε μου όλες τις στάσεις που απέχουν το πολύ 1km από την τοποθεσία μου πχ (38.070813012283, 23.710395367301), ταξινομημένες σε αύξουσα σειρά ως προς την απόσταση.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?station ?dist
WHERE {
    BIND("POINT(38.070813012283 23.710395367301)"^^<http://www.openlinksw.com/schemas/virttrdf#Geometry> as ?p)

    ?station ex:stationBusIsLocatedAt ?point.
    FILTER (bif:st_intersects (?p, ?point, 1)).
    BIND(bif:st_distance(?p, ?point) as ?dist).
}
ORDER BY ASC (?dist)
```

Execute Save Load Clear

station	dist
http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/StationBus/100010	0.0371649
http://www.ex.org/ontologyEL15187/StationBus/100149	0.0608393
http://www.ex.org/ontologyEL15187/StationBus/100083	0.213329
http://www.ex.org/ontologyEL15187/StationBus/100003	0.225152
http://www.ex.org/ontologyEL15187/StationBus/100177	0.225449
http://www.ex.org/ontologyEL15187/StationBus/100058	0.247605
http://www.ex.org/ontologyEL15187/StationBus/100140	0.373187
http://www.ex.org/ontologyEL15187/StationBus/100178	0.374513
http://www.ex.org/ontologyEL15187/StationBus/100059	0.452115
http://www.ex.org/ontologyEL15187/StationBus/100057	0.476436
http://www.ex.org/ontologyEL15187/StationBus/100259	0.542953
http://www.ex.org/ontologyEL15187/StationBus/100001	0.579401
http://www.ex.org/ontologyEL15187/StationBus/100060	0.64508

Query 8

Φέρε όλα τα λεωφορεία και τις αντίστοιχες στάσεις που απέχουν το πολύ 1km από την τοποθεσία μου πχ ίδια με πριν, ταξινομημένα με αύξουσα σειρά ως προς την απόσταση.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?bus ?station ?dist
WHERE {
  BIND("POINT(38.070813012283 23.710395367301)"^^<http://www.openlinksw.com/schemas/virttrdf#Geometry> as ?p)

  ?station ex:stationBusIsLocatedAt ?point.
  FILTER (bif:st_intersects (?p, ?point, 1)).
  BIND(bif:st_distance(?p, ?point) as ?dist).
  ?bus ex:hasStationBus ?station
}
ORDER BY ASC (?dist)
```

bus	station	dist
http://www.ex.org/ontologyEL15187/TripBus/9469203-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469204-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469205-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469206-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469207-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469208-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469209-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469210-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469211-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469212-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469213-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469214-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11
http://www.ex.org/ontologyEL15187/TripBus/9469215-CALEND-B12-Kaθnyερνv-12	http://www.ex.org/ontologyEL15187/StationBus/100002	3.46648e-11

Προφανώς το ερώτημα αυτό έχει ως αποτέλεσμα πάρα πολλούς συνδυασμούς.

Query 9

Με πόσους τρόπους μπορώ να πάρω ένα λεωφορείο για να πάω από την μία στάση στην άλλη, με δοσμένα τα ids τους.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>

SELECT distinct ?bus1
WHERE {
  ?bus1 a ex:Bus.
  ?bus1 ex:fulfilsTripBus ?trip1.
  ?trip1 ex:hasStationBus st:100003.
  ?trip1 ex:hasStationBus st:100001.
}
```


bus1
http://www.ex.org/ontologyEL15187/Bus/B12

Query 10

Με πόσους τρόπους μπορώ να πάρω δύο λεωφορεία για να πάω από την μία στάση στην άλλη, με δοσμένα τα ids τους.

```

PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>

SELECT distinct ?bus1 ?bus2
WHERE {
    ?bus1    a      ex:Bus.
    ?bus1    ex:fulfilsTripBus    ?trip1.
    ?trip1   ex:hasStationBus    st:100003.
    ?trip1   ex:hasStationBus    ?station.
    ?bus2    a      ex:Bus.
    ?bus2    ex:fulfilsTripBus    ?trip2.
    ?trip2   ex:hasStationBus    ?station.
    ?trip2   ex:hasStationBus    st:100001.
    FILTER(?bus1 != ?bus2)
}

```

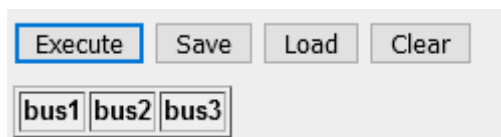
bus1	bus2
http://www.ex.org/ontologyEL15187/Bus/B12	http://www.ex.org/ontologyEL15187/Bus/711
http://www.ex.org/ontologyEL15187/Bus/713	http://www.ex.org/ontologyEL15187/Bus/B12
http://www.ex.org/ontologyEL15187/Bus/723	http://www.ex.org/ontologyEL15187/Bus/B12

Query 11

Με πόσους τρόπους μπορώ να πάρω τρία λεωφορεία για να πάω από την μία στάση στην άλλη, με δοσμένα τα ids τους.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>

SELECT distinct ?bus1 ?bus2 ?bus3
WHERE {
    ?bus1 a ex:Bus.
    ?bus1 ex:fulfilsTripBus ?trip1.
    ?trip1 ex:hasStationBus st:100003.
    ?trip1 ex:hasStationBus ?station.
    ?bus2 a ex:Bus.
    ?bus2 ex:fulfilsTripBus ?trip2.
    ?trip2 ex:hasStationBus ?station.
    ?trip2 ex:hasStationBus st:station2.
    ?bus3 a ex:Bus.
    ?bus3 ex:fulfilsTripBus ?trip3.
    ?trip3 ex:hasStationBus ?station2.
    ?trip3 ex:hasStationBus st:100010
}
```

A screenshot of a query execution interface. It features four buttons at the top: 'Execute' (highlighted with a blue border), 'Save', 'Load', and 'Clear'. Below these buttons is a row of three input fields labeled 'bus1', 'bus2', and 'bus3'.

Με κανέναν τρόπο.

Query 12

Φέρε μου ποια λεωφορεία μπορώ να πάρω για να πάω από μία στάση σε μία άλλη, με δοσμένα τα ids, και να φθάσω εκεί το χρονικό πλαίσιο 10.22-10.45, ταξινομήμενα σε αύξουσα σειρά ως προς την έγκυρη ώρα που θα φθάσω.

```

PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX st:<http://www.ex.org/ontologyEL15187/StationBus/>

SELECT distinct ?bus1 ?time
WHERE {
    ?bus1 a ex:Bus.
    ?bus1 ex:fulfilsTripBus ?trip1.
    ?trip1 ex:hasStationBus st:100003.
    ?trip1 ex:hasStationBus st:100001.
    ?trip1 ex:tripBusHasStopTime ?time.
    st:100001 ex:stationBusStopTime ?time.
    FILTER(?time > "10:22:00" )
    FILTER(?time < "10:45:00" )
}
ORDER BY ASC (?time)

```

Execute Save Load Clear

bus1	time
http://www.ex.org/ontologyEL15187/Bus/B12	"10:30:00"
http://www.ex.org/ontologyEL15187/Bus/B12	"10:31:00"
http://www.ex.org/ontologyEL15187/Bus/B12	"10:36:00"
http://www.ex.org/ontologyEL15187/Bus/B12	"10:41:00"

Query 13

Φέρε μου όλα τα ονόματα των στάσεων ενός λεωφορείου πχ του 410.

```

12.
// FERE OLA TA ONOMATA STASEWN TOU 410
PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX bus:<http://www.ex.org/ontologyEL15187/Bus/>

SELECT distinct ?station ?name
WHERE {
    bus:410 ex:fulfilsTripBus ?trip.
    ?trip ex:hasStationBus ?station.
    ?station ex:stationBusHasName ?name
}

```

Execute Save Load Clear

station	name
http://www.ex.org/ontologyEL15187/StationBus/640003	"ΠΑΛΑΝΤΟΡΟΣ"
http://www.ex.org/ontologyEL15187/StationBus/640007	"ΠΑΠΑΝΙΚΟΛΑΟΥ"
http://www.ex.org/ontologyEL15187/StationBus/640008	"ΠΛ. ΜΕΛΙΣΣΙΩΝ"
http://www.ex.org/ontologyEL15187/StationBus/640009	"ΔΙΑΤΙΤΑΡΧΕΩΝ"
http://www.ex.org/ontologyEL15187/StationBus/640010	"ΓΟΥΝΑΡΗ"
http://www.ex.org/ontologyEL15187/StationBus/640018	"ΜΠΟΝΟΠΟΛΙ"
http://www.ex.org/ontologyEL15187/StationBus/640019	"ΤΡΟΙΑΣ"
http://www.ex.org/ontologyEL15187/StationBus/640020	"ΦΟΥΡΟΣ"
http://www.ex.org/ontologyEL15187/StationBus/640024	"ΒΙΤΛΑΣ"
http://www.ex.org/ontologyEL15187/StationBus/640026	"ΠΑΛΛΙΝΟΜΕΡΕ"
http://www.ex.org/ontologyEL15187/StationBus/640027	"ΜΟΥΣΙΩΝ"

Καταλαβαίνουμε ότι είναι πάρα πολλά τα ονόματα των στάσεων ενός λεωφορείου για αυτό φέρνουμε ορισμένα για λόγους ευχρηστίας.

4.2 Ερωτήματα SPARQL για τα υπόλοιπα μέσα μεταφοράς

Όπως είχαμε πει και σε εισαγωγικές ενότητες, θα παρουσιάσουμε βασικά ερωτημάτα SPARQL ακόμα και για μέσα μεταφοράς που δεν έχουμε δεδομένα. Θα δείξουμε από ένα για κάθε άλλο μέσο.

Query 14: Αεροπλάνα

Φέρε μου το trip για να παώ από το ένα αεροδρόμιο στο άλλο, δεδομένων των ονομάτων τους

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>
PREFIX air:<http://www.ex.org/ontologyEL15187/Airport>

SELECT ?trip
WHERE {
    ?airport1    ex:airportHasName    "ΕΛ.ΒΕΝΙΖΕΛΟΣ".
    ?airport2    ex:airportHasName    "ΜΕΓΑΣ ΑΛΕΞΑΝΔΡΟΣ".
    ?airplane    ex:fulfillsTripFlight ?trip.
    ?trip        ex:hasAirport        ?airport1.
    ?trip        ex:hasAirport        ?airport2.
}
```

Query 15: Πλοία

Φέρε το trip για να πάω από το ένα λιμάνι στο άλλο δεδομένων των ονομάτων τους.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?trip
WHERE {
    ?port1    ex:portHasName    "ΠΕΙΡΑΙΑΣ".
    ?port2    ex:portHasName    "ΜΥΚΟΝΟΣ".
    ?ship     ex:fulfilsTripShip ?trip.
    ?trip     ex:hasAirport     ?port1.
    ?trip     ex:hasAirport     ?port2.
}
```

Query 15: Τρένα

Φέρε το trip για να πάω από έναν σταθμό σε έναν άλλον, δεδομένων των ονομάτων τους, αλλά με συγκεκριμένες χρονικές στιγμές.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?trip
WHERE {
    ?st1      ex:stationTrainHasName    "ΚΗΦΙΣΙΑ".
    ?st2      ex:stationTrainHasName    "ΝΕΟ ΗΡΑΚΛΕΙΟ".
    ?trip     ex:hasStationTrain    ?st1.
    ?trip     ex:hasStationTrain    ?st2.
    ?st2      ex:stationTrainStopTime    ?time.
    ?trip     ex:tripTrainStopTime    ?time.
    FILTER(?time > "9:32:00").
    FILTER(?time < "9:55:00").
}
```

Query 16: Τραμ

Φέρε μου τα τραμ που περνάνε από στάση με δοσμένες γεωγραφικές συντεταγμένες.

```
SELECT distinct ?tram
WHERE {
  BIND("POINT(38.070283 23.71039)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry> as ?p).

  ?station ex:stationTramIsLocatedAt ?point.
  FILTER(bif:st_distance(?p, ?point) < 0.0001).

  ?trip ex:hasStationTram ?station.
  ?tram ex:fulfilsTripTram ?trip.
}
```

Query 17: Μετρό

Φέρε το trip και το time που θα πάω από μια στάση σε μία άλλη, δεδομένων των ονομάτων τους, αλλά σε συγκεκριμένα χρονικά πλαίσια.

```
PREFIX ex:<http://www.ex.org/ontologyEL15187/>

SELECT ?trip ?time
WHERE {
  ?st1 ex:stationUndergroundHasName "ΔΟΥΚΙΣΣΗΣ ΠΛΑΚΕΝΤΙΑΣ".
  ?st2 ex:stationUndergroundHasName "ΚΑΤΕΧΑΚΗ".
  ?trip ex:hasStationUnderground ?st1.
  ?trip ex:hasStationUnderground ?st2.
  ?st2 ex:stationUndergroundStopTime ?time.
  ?trip ex:tripUndergroundStopTime ?time.
  FILTER(?time > "10:22:00").
  FILTER(?time < "10:45:00").
}
```

Σημειώνουμε πως τα παραπάνω ερωτήματα θα μπορούσαν να είναι αρκετά περισσότερα, αν είχαμε πρόσβαση σε αντίστοιχα δεδομένα. Αξίζει να τονίσουμε πως όλοι οι ρόλοι και οι ιδιότητες συμβαδίζουν με την οντολογία που δείξαμε στην αντίστοιχη ενότητα.

5.1 Συζήτηση

Η παραπάνω εργασία χτίστηκε πάνω στα δεδομένα που είχαμε για το δίκτυο των λεωφορείων της Αθήνας. Θεωρούμε πως θα είχε ιδιαίτερο ενδιαφέρον εάν είχαμε δεδομένα και για τα υπόλοιπα μέσα, προκειμένου να φτιάχναμε ένα σύστημα που να ανταποκρινόταν πλήρως στις απαιτήσεις του συγκοινωνιακού δικτύου της Ελλάδος. Επίσης, θα ήταν εξίσου ενδιαφέρον να ασχοληθούμε με την δημιουργία εφαρμογής που να τα είχε όλα αυτά μαζί και να συνδέαμε έτσι τις γνώσεις που λάβαμε από το μάθημα με τις γνώσεις από το μάθημα της Τεχνολογίας Λογισμικού.

5.2 References

<http://geodata.gov.gr/el/dataset/oasa>
<http://docs.openlinksw.com/virtuoso/rdfsparqlgeospat/>
<https://rdflib.readthedocs.io/en/stable/>
<https://pandas.pydata.org/>
<https://www.w3.org/TR/rdf-sparql-query/>
<https://protege.stanford.edu/>