

Εργαστηριακό θέμα στα Συστήματα και Τεχνολογίες Γνώσης

Δημήτρης Γαλάνης -03116088
el16088@central.ntua.gr

Αύγουστος 2020

1 Εισαγωγή

Σκοπός αυτής της εργασίας είναι η κατασκευή μιας σημασιολογικής βάσης γνώσης με δεδομένα σχετικά με τα μέσα μεταφοράς και της αντίστοιχης OWL2 οντολογίας που τα μοντελοποιεί. Τα δεδομένα που χρησιμοποιούνται αφορούν τα δρομολόγια των λεωφορείων του ΟΑΣΑ ¹, όμως η οντολογία που τα μοντελοποιεί είναι γενικότερη και μπορεί να μοντελοποιήσει δεδομένα από τον ευρύτερο χώρο των μέσων μεταφοράς.

Η κατασκευή και επαλήθευση της κατασκευασθείσας βάσης γνώσης έγκειται στα παρακάτω βήματα:

1. Κατασκευή της OWL2 οντολογίας.
2. Κατασκευή του A-BOX της οντολογίας, δηλαδή των RDF τριάδων που μοντελοποιούν τα δεδομένα βάσει της οντολογίας.
3. Εισαγωγή των δεδομένων σε μια αποθήκη τριάδων.
4. Επαλήθευση της βάσης και επίδειξη των εκφραστικών δυνατοτήτων της οντολογίας μέσω queries στην βάση.
5. Σχολιασμός προβλημάτων και περιορισμών του συστήματος που αναπτύχθηκε.

¹<http://geodata.gov.gr/el/dataset/oasa>

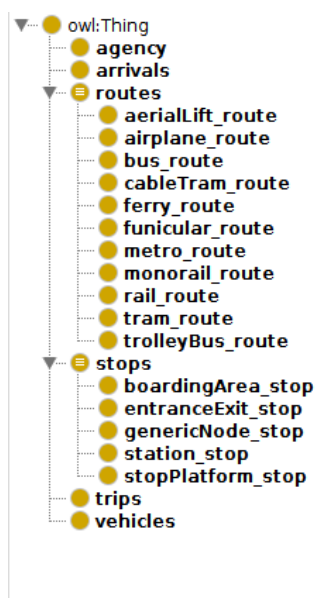
2 Οντολογία Μέσων Μεταφοράς

Η OWL2 οντολογία κατασκευάστηκε με την βοήθεια του open-source ontology editor: Protégé ² το οποίο προσφέρει ένα εύκολο στην χρήση UI που επιτρέπει την γρήγορη κατασκευή κλάσεων και ιδιοτήτων κλάσεων και δεδομένων καθώς και την εφαρμογή περιορισμών σε αυτές.

2.1 Συνοπτική παρουσίαση οντολογίας

Στην συνέχεια παραθέτουμε μια συνοπτική παρουσίαση της οντολογίας. Περισσότερες λεπτομέρειες παρέχονται στο ίδιο το αρχείο *owl* της οντολογίας.

Οι κλάσεις της κατασκευασθείσας οντολογίας δίνονται παρακάτω:



Σχήμα 1: Κλάσεις της κατασκευασθείσας οντολογίας.

Επιγραμματικά, έχουν οριστεί οι κλάσεις των δρομολογίων(*routes*), των ταξιδιών(*trips*) των στάσεων(*stops*), των αφίξεων των ταξιδιών σε στάσεις(*arrivals*), των πρακτορείων που πραγματοποιούν τα ταξίδια(*agency*) και των οχημάτων που ανήκουν στα *agencies* και μέσω των οποίων πραγματοποιούνται τα *trips*.

Πιο συγκεκριμένα, στις κλάσεις *routes* και *stops* έχουν οριστεί και υποκλάσεις η οποίες υπακούουν στο σχήμα δεδομένων του GTFS ³ και ορίζουν το είδος του μέσου που χρησιμοποιεί το δρομολόγιο και το είδος της στάσης (πχ επιβίβασης, αποβίβασης) αντίστοιχα.

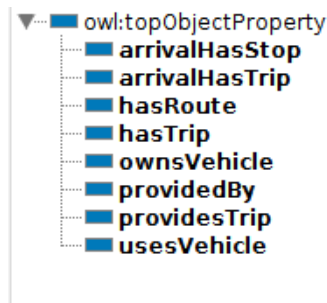
Οι ιδιότητες αντικειμένων που ορίζονται στην οντολογία είναι οι παρακάτω:

Όπως γίνεται εμφανές από τον ορισμό των instances της κλάσεως *arrivals* ένα *arrival* ορίζεται βάσει ενός instances της κλάσεως *stops* και ενός της κλάσεως *trips* και επομένως η οντολογία ορίζει τις αντίστοιχες ιδιότητες αντικειμένων *arrivalHasStop*, *arrivalHasTrip* που συνδέουν ένα instance της κλάσης *arrivals* με instances της κλάσης *stops* και *trips* αντίστοιχα.

Επίσης ορίζονται οι ιδιότητες αντικειμένων *ownsVehicle*, *providesTrip* που συνδέουν instances της κλάσεως *agency* με τα instances των κλάσεων *vehicle* που της ανήκουν και *trips* που παρέχουν, αντίστοιχα και οι ιδιότητες αντικειμένων *hasRoute*, *hasTrip* που συνδέουν instances της κλάσης *trips* με τα αντίστοιχα instances της κλάσης *routes* και αντίστροφα. Τέλος η ιδιότητα αντικειμένων *usesVehicle* συνδέει κάθε instance της κλάσης *trips* με το instance της κλάσης *vehicle* που χρησιμοποιεί το συγκεκριμένο *trip*.

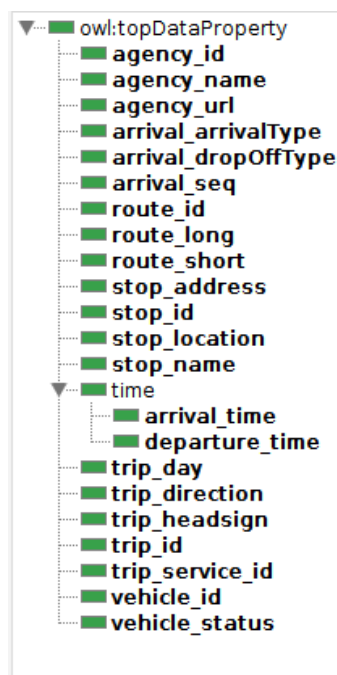
²<https://protege.stanford.edu/products.php>

³<https://gtfs.org/>



Σχήμα 2: Ιδιότητες αντικειμένων της κατασκευασθείσας οντολογίας.

Τέλος, ορίζουμε τις ιδιότητες δεδομένων της οντολογίας όπως φαίνονται στο παρακάτω σχήμα:



Σχήμα 3: Ιδιότητες αντικειμένων της κατασκευασθείσας οντολογίας.

Βλέπουμε πως για κάθε κλάση ορίζονται οι ιδιότητες δεδομένων που μοντελοποιούν επαρκώς τον χώρο των μέσων μεταφοράς. Ενδεικτικά, για τα instances της κλάσης *agency* ορίζονται οι ιδιότητες δεδομένων *agency_id*, *agency_name*, *agency_url* που συνδέουν instances της κλάσης *agency* με τα αντίστοιχα *Literals* που αφορούν τα *id*, το όνομα και το *url* του *agency*. Αντίστοιχες ιδιότητες δεδομένων συσχετίζουν στιγμιότυπα και των κλάσεων *route*, *arrival*, *stop*, *trip*, *vehicle*. Οι ορισμοί τους και οι περιορισμοί πάνω στο σύνολο ορισμού και τιμών που εφαρμόζονται σε αυτές βρίσκονται αναλυτικά στο *OWL* αρχείο της οντολογίας.

2.2 Σχολιασμός σχεδιαστικών αποφάσεων

Η οντολογία κατασκευάστηκε με στόχο να είναι αρκετά εκφραστική ώστε να μπορεί να περιγράψει τον τομέα των μέσων μεταφοράς αλλά και με στόχο να είναι όσο το δυνατόν πιο γενική, ώστε να μπορούν να μοντελοποιηθούν, μέσω αυτής, διαφορετικά μέσα μεταφοράς, είδη δρομολογίων ή ακόμη και τρόπος περιγραφής δρομολογίων. Πιο συγκεκριμένα, βάσει των παραπάνω, οδηγηθήκαμε στις παρακάτω σχεδιαστικές αποφάσεις:

- Κατασκευή υποκλάσεων για τις κλάσεις *stops* και *routes* βάσει του προτύπου δεδομένων *GTFS* καθώς είναι καθιερωμένο στις εφαρμογές σχετικές με τα μέσα μεταφοράς.
- Διαχωρισμός των δρομολογίων (*routes*) από τα ταξίδια (*trips*). Αυτό συμβαίνει διότι θέλουμε να αποδίδουμε περισσότερες και ενδεχομένες διαφορετικές πληροφορίες σε διαφορετικά στιγμιότυπα της κλάσης *trips*.
- Συσχέτιση των στάσεων των μέσων μεταφοράς(*stops*) με τα ταξίδια(*trips*) και όχι με τα δρομολόγια(*routes*), μέσω της *arrivalHasStop*, καθώς είναι επιθυμητό διαφορετικά ταξίδια του ίδιου δρομολογίου (π.χ τα νυκτερινά ταξίδια ενός λεωφορείου) να έχουν διαφορετικές στάσεις από ότι τα υπόλοιπα.
- Συσχέτιση των αφίξεων σε μια στάση με στιγμιότυπα της κλάσης *trip* μέσω της *arrivalHasTrip* αφού κάθε δρομολόγιο μπορεί να φτάνει πολλές φορές την ημέρα στην ίδια στάση, ή διαφορετικές μέρες να φτάνει σε διαφορετικές ώρες.
- Η ιδιότητα αντικειμένων *providesTrip* συνδέει στιγμιότυπα της κλάσης *agency* με στιγμιότυπα της κλάσης *trips* και όχι της *routes* καθώς το αυτό θα ανάγκαζε έναν πάροχο να εξυπηρετεί όλα τα ταξίδια ενός δρομολογίου.

3 Μετατροπή των δεδομένων σε RDF τριάδες.

Όπως έχει αναφερθεί και στην εισαγωγή, ως δεδομένα χρησιμοποιήθηκαν τα ανοικτά γεωχωρικά δεδομένα του ΟΑΣΑ που παρέχονται δωρεάν από το <http://geodata.gov.gr/> και μεταφορτώνονται στο τοπικό σύστημα αρχείων μέσω της συνταγής *download* του *Makefile* αρχείου του *project*. Για την κατασκευή των RDF τριάδων χρησιμοποιήθηκε η *Python* βιβλιοθήκη κατασκευής RDF τριάδων, *RDFLib* ⁴. Η επιλογή έγινε λόγω της πρότερης γνώσης της γλώσσας *Python* και στην απλό και άμεσο τρόπο με τον οποίο η *RDFLib* μπορεί να σειριοποιήσει RDF τριάδες.

3.1 Ενδεικτική διαδικασία κατασκευής RDF τριάδων.

Μια ενδεικτική διαδικασία κατασκευής RDF τριάδων που αφορούν της κλάση *routes* περιγράφεται παρακάτω:

- Αρχικά κατασκευάζεται το Namespace *MoPT* που αφορά όλα τα URIs που χρησιμοποιούνται από την βάση μας (δηλαδή είναι *prefix* τους). Βάσει αυτού, κατασκευάζουμε και τα URIs που αφορούν ιδιότητες αντικειμένων ή δεδομένων που συσχετίζονται με στιγμιότυπα της κλάσης *routes*. Ο αντίστοιχος κώδικας φαίνεται στο Σχήμα 4.
- Στην συνέχεια, αρχικοποιείται ο RDF γράφος και μέσω της *standard* βιβλιοθήκης της *Python* διαβάζονται τα δεδομένα του αρχείου *routes.txt* γραμμή προς γραμμή. Σε κάθε γραμμή, απομονώνονται τα δεδομένα που είναι χρήσιμα για την βάση δεδομένων μας, μέσω κατάλληλων εφαρμογών της εντολής *split*. Στο συγκεκριμένο παράδειγμα, πέραν των παραπάνω, παραμετροποιείται βάσει των δεδομένων εισόδου και το URI που μοντελοποιεί τον τύπο του *route*. Ο αντίστοιχος κώδικας φαίνεται στο Σχήμα 5.
- Έπειτα, βάσει του αναγνωριστικού του *route*, το οποίο έχουμε εξάγει όπως εξηγήσαμε παραπάνω, κατασκευάζουμε το URI που αντιστοιχεί στο τρέχον στιγμιότυπο της κλάσης *route* και το χρησιμοποιούμε μαζί με τα URIs που κατασκευάστηκαν σε προηγούμενο βήμα ως παραμέτρους της μεθόδου *add* της *RDFLib* για να εισαχθεί η RDF τριάδα στον RDF γράφο. Ο αντίστοιχος κώδικας δίνεται στο Σχήμα 6.

⁴<https://rdflib.readthedocs.io/en/stable/>

```

'''
This script produces the Ntriples containing basic information about
the (GTFS) route data class.
routes information should be in source/data/routes.txt
output file routes-basic.??? will be placed in source/Abox
'''

from rdflib import Graph, Literal, RDF, URIRef, Namespace, BNode
from rdflib.namespace import FOAF, XSD
import os.path

if __name__ == '__main__':

    #MoPT = URIRef("http://www.semanticweb.org/knowsys_project/ontologies/MoPT#")
    MoPT = Namespace("http://www.semanticweb.org/knowsys_project/ontologies/MoPT#")
    route_long_name = MoPT.route_long #URIRef("http://www.semanticweb.org/knowsys_project/ontologies/MoPT#route_long")
    route_id_URI = MoPT.route_id
    route_short_name = MoPT.route_short
    # Create RDF URI nodes for every route type to use as the subject for multiple triples

    routes = MoPT.routes
    aerialLift_route = MoPT.aerialLift_route
    bus_route = MoPT.bus_route
    cableTram_route = MoPT.cableTram_route
    ferry_route = MoPT.ferry_route
    funicular_route = MoPT.funicular_route
    metro_route = MoPT.metro_route
    monorail_route = MoPT.monorail_route
    rail_route = MoPT.rail_route
    tram_route = MoPT.tram_route
    trolleyBus_route = MoPT.trolleyBus_route
    #route type list (ordered following GTFS specification)
    routeType_list = [tram_route, metro_route, rail_route, bus_route, \
                      ferry_route, cableTram_route, aerialLift_route, funicular_route, \
                      trolleyBus_route, monorail_route]

```

Σχήμα 4: Κατασκευή των απαραίτητων URIs.

```

g = Graph() #RDF graph constructor
#read the data from routes.txt
f = open(os.path.dirname(__file__) + '/../data/routes.txt')
#consume first line
line = f.readline()
line=f.readline()
while line:
    #split line and get data
    line = line.split(',')
    route_id = line[0].split('-')[0]
    route_short = line[1]
    route_long=line[2][1:len(line[2])-1]
    route_type=line[4]

    #add data to RDF Graph
    #however the data do not follow exactly the GTFS route_type specifications
    #see source/data/dior8wseis.txt

    if route_type == '900':
        route_type_URI = routeType_list[2]
    elif route_type == '800':
        route_type_URI = routeType_list[0]
    elif int(route_type) < 8:
        route_type_URI = routeType_list[int(route_type)]
    else:
        route_type_URI = routeType_list[int(route_type) -3]]

```

Σχήμα 5: Εξαγωγή των χρήσιμων δεδομένων από το αρχείο εισόδου.

```

#name URI for this route using its short name (i.e buss with number 131 will have
#URI: http://www.semanticweb.org/knowsys_project/ontologies/MoPT#131)
this_URI = URIRef("http://www.semanticweb.org/knowsys_project/ontologies/MoPT#route_" + route_id)

g.add((this_URI, RDF.type, route_type_URI))
g.add((this_URI, route_long_name, Literal(route_long, datatype=XSD.string)))
g.add((this_URI, route_id_URI, Literal(route_id, datatype=XSD.string)))
g.add((this_URI, route_short_name, Literal(route_short, datatype=XSD.string)))
#since no Reasoning takes place we also have to assert that this particular route type route is a route
g.add((this_URI, RDF.type, routes))

#read next line
line=f.readline()

```

Σχήμα 6: Κατασκευή και προσθήκη των RDF τριάδων στον γράφο.

- Τέλος, ο ονοματόχωρος MoPT συνδέεται με το prefix MoPT ώστε το αρχείο που παράγεται να είναι πιο ευανάγνωστο και ο RDF γράφος σειριοποιείται σε ένα αρχείο και ακολουθώντας κάποιο συγκεκριμένο μορφότυπο (στο συγκεκριμένο παράδειγμα χρησιμοποιείται το μορφότυπο *Turtle*). Ακόμα δημιουργείται ένα αρχείο με κατάληξη “.graph” που περιέχει μόνο μια γραμμή με το URI που θα αντιστοιχεί στον γράφο δεδομένων όταν αυτά φορτωθούν στην αποθήκη τριάδων (δες και μέρος 4). Ο αντίστοιχος κώδικας δίνεται στο Σχήμα 7

```
#bind namespaces to prefixes for more readable output
g.bind("MoPT", MoPT)
f.close()
#serialize to file using preferred format
g.serialize(destination="source/Abox/routes-basic.txt", format="ttl")
ff = open(os.path.dirname(__file__) + '/../Abox/routes-basic.graph', "w+")
ff.write("http://localgraph.org/routes-basic")
ff.close()
```

Σχήμα 7: Σειριοποίηση του γράφου σε αρχείο.

Ανάλογη επεξεργασία γίνεται και κατά την κατασκευή των RDF τριάδων που αφορούν τις υπόλοιπες κλάσεις και ιδιότητες. Οι όποιες διαφοροποιήσεις εξηγούνται επαρκώς στα σχόλια που συνοδεύουν τον κώδικα.

3.2 Ενδεικτικά αποτελέσματα σειριοποίησης αποτελεσμάτων

Παρακάτω διαθέτουμε ενδεικτικά παραδείγματα από την σειριοποίηση του RDF γράφου για τα δεδομένα των κλάσεων της οντολογίας μας.

Για παράδειγμα, στα σειριοποιημένα δεδομένα της κλάσης *routes* (Σχήμα 8) βλέπουμε πως κάθε στιγμιότυπο είναι τύπου *routes* αλλά και τύπου της αντίστοιχης *GTFIS* υποκλάσης δρομολογίων η οποία βρίσκεται μέσω των δεδομένων εισόδου τα οποία αφορούν κυρίως δρομολόγια λεωφορείων αλλά και μερικά δρομολόγια τραμ.

```
@prefix MoPT: <http://www.semanticweb.org/knownsys/project/ontologies/MoPT#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

MoPT:route_021 a MoPT:bus_route,
    MoPT:routes ;
    MoPT:route_id "021"^^xsd:string ;
    MoPT:route_long "ΚΑΝΙΓΓΟΣ - ΓΚΥΖΗ"^^xsd:string ;
    MoPT:route_short "021"^^xsd:string .

MoPT:route_022 a MoPT:bus_route,
    MoPT:routes ;
    MoPT:route_id "022"^^xsd:string ;
    MoPT:route_long "Ν. ΚΥΨΕΛΗ - ΜΑΡΑΣΙΛΕΙΟΣ"^^xsd:string ;
    MoPT:route_short "022"^^xsd:string .

MoPT:route_024 a MoPT:bus_route,
    MoPT:routes ;
    MoPT:route_id "024"^^xsd:string ;
    MoPT:route_long "ΑΓ. ΑΝΑΓΥΡΟΙ - ΣΤ. ΑΤΤΙΚΗΣ - ΣΤ. ΛΑΡΙΣΗΣ (ΚΥΚΛΙΚΗ)"^^xsd:string ;
    MoPT:route_short "024"^^xsd:string .

MoPT:route_025 a MoPT:bus_route,
    MoPT:routes ;
    MoPT:route_id "025"^^xsd:string ;
    MoPT:route_long "ΙΠΠΟΚΡΑΤΟΥΣ - ΠΡΟΦ. ΔΑΝΙΗΛ"^^xsd:string ;
    MoPT:route_short "025"^^xsd:string .
```

Σχήμα 8: Ενδεικτικά σειριοποιημένα δεδομένα της κλάσης *routes*.

Στα σειριοποιημένα δεδομένα της κλάσης *stops* (Σχήμα 9) βλέπουμε πως κάθε στιγμιότυπο είναι τύπου *stops* αλλά και τύπου της αντίστοιχης *GTFIS* υποκλάσης στάσεων η οποία βρίσκεται μέσω των δεδομένων εισόδου τα οποία αφορούν αποκλειστικά στάσεις-πλατφόρμες.

```

@prefix LinkedGeoData: <http://www.openlinksw.com/schemas/virttrdf#> .
@prefix MoPT: <http://www.semanticweb.org/knownsys_project/ontologies/MoPT#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

MoPT:stop_010001 a MoPT:stopPlatform_stop,
    MoPT:stops ;
    MoPT:stop_address "Επί της ΕΑ.BENIZEΛΟΥ"^^xsd:string ;
    MoPT:stop_id "10001"^^xsd:int ;
    MoPT:stop_location "POINT(37.9986082641367 23.6649846246733)"^^LinkedGeoData:Geometry ;
    MoPT:stop_name "ΣΤΡΟΦΗ"^^xsd:string .

MoPT:stop_010002 a MoPT:stopPlatform_stop,
    MoPT:stops ;
    MoPT:stop_address "Επί της ΑΕΩ.ΕΑ.BENIZEΛΟΥ"^^xsd:string ;
    MoPT:stop_id "10002"^^xsd:int ;
    MoPT:stop_location "POINT(37.996719436997 23.6631823242898)"^^LinkedGeoData:Geometry ;
    MoPT:stop_name "ΚΟΛΩΝΕΣ"^^xsd:string .

MoPT:stop_010003 a MoPT:stopPlatform_stop,
    MoPT:stops ;
    MoPT:stop_address "Επί της ΑΕΩ.ΕΑ.BENIZEΛΟΥ"^^xsd:string ;
    MoPT:stop_id "10003"^^xsd:int ;
    MoPT:stop_location "POINT(37.9946947354979 23.6611529618238)"^^LinkedGeoData:Geometry ;
    MoPT:stop_name "ΑΓ.ΒΑΡΒΑΡΑ"^^xsd:string .

```

Σχήμα 9: Ενδεικτικά σειριοποιημένα δεδομένα της κλάσης *stops*.

Στα ενδεικτικά παραδείγματα σειριοποιημένων δεδομένων της κλάσης *trips* που υπάρχουν παρακάτω (Σχήμα 10) είναι εμφανής η σχεδιαστική επιλογή που αναφέρθηκε στο μέρος 2: τα πρακτορεία(*agencies*) να παρέχει ταξίδια που αντιστοιχούν σε δρομολόγια και όχι να παρέχει δρομολόγια (κάτι που θα συνεπάγονταν πως θα έπρεπε να παρέχει όλα τα ταξίδια ενός δρομολογίου).

Εδώ σημειώνουμε, πως η γεωγραφική τοποθεσία του κάθε στιγμιότυπου της κλάσεως *stops* με χρήση του τύπου δεδομένων `http://www.openlinksw.com/schemas/virttrdf#Geometry` που έχουν υιοθετήσει και τα *LinkedGeoData*⁵.

Τέλος, στα ενδεικτικά παραδείγματα σειριοποιημένων δεδομένων της κλάσης *arrivals* που υπάρχουν παρακάτω (Σχήμα 11) φαίνεται η σχεδιαστική επιλογή του να συσχετίσουμε τις αφίξεις με τα ταξίδια(*trips*) και τις αντίστοιχες στάσεις, ώστε εν τέλει να ισχύει η τρίτη και η τέταρτη παρατήρηση του μέρους 2.2 .

4 Εισαγωγή των RDF τριάδων στην αποθήκη τριάδων

Η αποθήκη τριάδων που χρησιμοποιούμε για την βάση δεδομένων είναι η ανοικτού κώδικα έκδοση 7.2.5 του Virtuoso⁶. Ο *server* εκκινείται μέσω της συνταγής *start server* του *Makefile* αρχείου του *project*. Τα δεδομένα εισάγονται στην αποθήκη τριάδων μέσω των εντολών

```

ld_dir('Path_to_project_ABOX', 'input_file.txt', 'graph_desired_URI');
rdf_loader_run();
checkpoint;

```

οι οποίες εισάγονται σε μια γραμμή εντολών *isql* η οποία 'βλέπει' τον τοπικό *server*.

⁵<http://linkedgeodata.org/About>

⁶<http://vos.openlinksw.com/owiki/wiki/VOS>


```

@prefix MoPT: <http://www.semanticweb.org/knownsys_project/ontologies/MoPT#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

MoPT:route_801 MoPT:hasTrip MoPT:trip_9761436 .

MoPT:route_836 MoPT:hasTrip MoPT:trip_9714451 .

MoPT:route_735 MoPT:hasTrip MoPT:trip_9443402,
  MoPT:trip_9443403 .

MoPT:route_865 MoPT:hasTrip MoPT:trip_9619584,
  MoPT:trip_9619585 .

MoPT:trip_5253235 a MoPT:trips ;
  MoPT:hasRoute MoPT:route_322 ;
  MoPT:providedBy MoPT:agency_20 ;
  MoPT:trip_day "Σάββατο"^^xsd:string ;
  MoPT:trip_direction 0 ;
  MoPT:trip_headsign "\"ΑΓ. ΝΙΚΟΛΑΟΣ - ΒΡΑΥΡΩΝΑ (ΚΥΚΛΙΚΗ)\"""^^xsd:string ;
  MoPT:trip_id 5253235 ;
  MoPT:trip_route_id "322"^^xsd:string ;
  MoPT:trip_service_id "CALEND-322-Σάββατο-02"^^xsd:string .

MoPT:trip_5253236 a MoPT:trips ;
  MoPT:hasRoute MoPT:route_322 ;
  MoPT:providedBy MoPT:agency_20 ;
  MoPT:trip_day "Σάββατο"^^xsd:string ;
  MoPT:trip_direction 0 ;
  MoPT:trip_headsign "\"ΑΓ. ΝΙΚΟΛΑΟΣ - ΒΡΑΥΡΩΝΑ (ΚΥΚΛΙΚΗ)\"""^^xsd:string ;
  MoPT:trip_id 5253236 ;
  MoPT:trip_route_id "322"^^xsd:string ;
  MoPT:trip_service_id "CALEND-322-Σάββατο-02"^^xsd:string .

MoPT:trip_5253237 a MoPT:trips ;
  MoPT:hasRoute MoPT:route_322 ;
  MoPT:providedBy MoPT:agency_20 ;
  MoPT:trip_day "Σάββατο"^^xsd:string ;
  MoPT:trip_direction 0 ;
  MoPT:trip_headsign "\"ΑΓ. ΝΙΚΟΛΑΟΣ - ΒΡΑΥΡΩΝΑ (ΚΥΚΛΙΚΗ)\"""^^xsd:string ;
  MoPT:trip_id 5253237 ;
  MoPT:trip_route_id "322"^^xsd:string ;
  MoPT:trip_service_id "CALEND-322-Σάββατο-02"^^xsd:string .

```

Σχήμα 10: Ενδεικτικά σειριοποιημένα δεδομένα της κλάσης *trips*.

5 Επαλήθευση βάσης και επίδειξη εκφραστικότητας της οντολογίας μέσω queries.

Στο μέρος αυτό δίνουμε και εκτελούμε ενδεικτικά SPARQL ερωτήματα(*queries*) στην βάση δεδομένων με σκοπό την επαλήθευση της ορθής λειτουργίας της βάσης καθώς και την επίδειξη των εκφραστικών δυνατοτήτων της οντολογίας που έχουμε κατασκευάσει. Τα ερωτήματα χωρίζονται σε 2 κατηγορίες:

1. Τα ερωτήματα για τα οποία υπάρχουν δεδομένα στην αποθήκη τριάδων.
2. Τα ερωτήματα για τα οποία δεν υπάρχουν δεδομένα στην αποθήκη τριάδων (αλλά προφανώς εξακολουθούν να είναι σημασιολογικά σωστά για την οντολογία που έχουμε κατασκευάσει).

Τα ερωτήματα υποβάλλονται μέσω του web-server του Virtuoso.

5.1 Ερωτήματα με διαθέσιμα δεδομένα απάντησης στην βάση δεδομένων.

5.1.1 5 κοντινότερες στάσεις λεωφορείου

Το ερώτημα αυτό επιστρέφει το όνομα και την διεύθυνση των 5 κοντινότερων (από την τρέχουσα τοποθεσία) στάσεων λεωφορείου. Το SPARQL ερώτημα έχει ως εξής:

```

@prefix MoPT: <http://www.semanticweb.org/knownsys_project/ontologies/MoPT#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .

MoPT:arrival_010001_9717200 a MoPT:arrivals ;
    MoPT:arrivalHasStop MoPT:stop_010001 ;
    MoPT:arrivalHasTrip MoPT:trip_9717200 ;
    MoPT:arrival_arrivalType "0"^^xsd:int ;
    MoPT:arrival_dropOffType "0"^^xsd:int ;
    MoPT:arrival_seq "17"^^xsd:nonNegativeInteger ;
    MoPT:arrival_time "05:39:00"^^xsd:time ;
    MoPT:departure_time "05:39:00"^^xsd:time .

MoPT:arrival_010001_9717201 a MoPT:arrivals ;
    MoPT:arrivalHasStop MoPT:stop_010001 ;
    MoPT:arrivalHasTrip MoPT:trip_9717201 ;
    MoPT:arrival_arrivalType "0"^^xsd:int ;
    MoPT:arrival_dropOffType "0"^^xsd:int ;
    MoPT:arrival_seq "17"^^xsd:nonNegativeInteger ;
    MoPT:arrival_time "06:04:00"^^xsd:time ;
    MoPT:departure_time "06:04:00"^^xsd:time .

MoPT:arrival_010001_9717202 a MoPT:arrivals ;
    MoPT:arrivalHasStop MoPT:stop_010001 ;
    MoPT:arrivalHasTrip MoPT:trip_9717202 ;
    MoPT:arrival_arrivalType "0"^^xsd:int ;
    MoPT:arrival_dropOffType "0"^^xsd:int ;
    MoPT:arrival_seq "17"^^xsd:nonNegativeInteger ;
    MoPT:arrival_time "06:42:00"^^xsd:time ;
    MoPT:departure_time "06:42:00"^^xsd:time .

```

Σχήμα 11: Ενδεικτικά σειριοποιημένα δεδομένα της κλάσης *arrivals*.

PREFIX LinkedGeoData:

<http://www.openlinksw.com/schemas/virttrdf#>

PREFIX MoPT: <http://www.semanticweb.org/knownsys_project/ontologies/MoPT#>

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

SELECT ?bstop_address ?bstop_name

WHERE { ?bstop rdf:type MoPT:stops .

?bstop MoPT:stop_address ?bstop_address .

?bstop MoPT:stop_location ?bstop_loc .

?bstop MoPT:stop_name ?bstop_name .

FILTER EXISTS {

?arr MoPT:arrivalHasStop ?bstop .

?arr MoPT:arrivalHasTrip ?trip .

?trip MoPT:hasRoute ?route .

?route rdf:type MoPT:bus_route

}

}

ORDER BY (bif:st_distance(bif:st_point(37.941749, 23.721553), ?bstop_loc))

LIMIT 5

Όπως φαίνεται στο ερώτημα βρίσκουμε όλα τα στιγμιότυπα της κλάσης *stops* και τα ταξινομούμε βάσει της απόστασής τους από το σημείο με γεωγραφικές συντεταγμένες (37.941749, 23.721553). Για να επιβάλουμε οι στάσεις που θα επιστραφούν να είναι στάσεις λεωφορείου (δηλαδή στάσεις από τις οποίες περνάει ένα τουλάχιστον δρομολόγιο λεωφορείου) χρησιμοποιούμε το φίλτρο να υπάρξει τουλάχιστον ένα δρομολόγιο που περνάει από την στάση το οποίο να είναι δρομολόγιο λεωφορείου. Τα αποτελέσματα του ερωτήματος δίνονται παρακάτω στο Σχήμα 12:

bstop_address	bstop_name
"Επί της ΑΓΧΙΑΔΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΛΟΥΤΡΑ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΔΡΙΑΝΟΥΠΟΛΕΩΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΠΑΙΔΙΚΗ ΧΑΡΑ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΔΡΙΑΝΟΥΠΟΛΕΩΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΛΟΥΤΡΑ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΥΣΕΙΝΟΥ ΠΟΝΤΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΣΧΟΛΕΙΑ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΔΑΡΔΑΝΕΛΙΩΝ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΕΘΝΙΚΗ ΣΤΕΓΗ"^^<http://www.w3.org/2001/XMLSchema#string>

Σχήμα 12: Αποτελέσματα του query1.txt .

5.1.2 Στάσεις στην περιοχή της Νέας Σμύρνης

Σε αυτό το ερώτημα χρησιμοποιούμε τις πράξεις και το λεξιλόγιο που μπορούν να εφαρμοστούν σε δεδομένα που ακολουθούν το πρότυπο *GeoSPARQL*, για να βρούμε όλες τις στάσεις μέσω μεταφοράς που βρίσκονται στην ευρύτερη περιοχή της Νέας Σμύρνης. Το ερώτημα είναι το παρακάτω

```

PREFIX LinkedGeoData:
<http://www.openlinksw.com/schemas/virttrdf#>
PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>

SELECT ?stop_address , ?stop_name
WHERE {
  ?stop rdf:type MoPT:stops .
  ?stop MoPT:stop_location ?stop_loc .
  ?stop MoPT:stop_address ?stop_address .
  ?stop MoPT:stop_name ?stop_name .
  FILTER( bif:st_intersects( ?stop_loc ,  bif:st_geomfromtext ( "POLYGON_((37.946108_23.703202,
37.955182_23.715021,_37.950518_23.719589,_37.943749_23.727669,
37.936893_23.718950,_37.939916_23.713019,_37.946108_23.703202))" ) ) ) )
}

```

Για τον αποκλεισμό στάσεων που δεν βρίσκονται στην περιοχή της Νέας Σμύρνης αποκλείουμε μέσω ενός φίλτρου όλα τις στάσεις με τοποθεσία που δεν είναι εντός ενός πολυγώνου που κατασκευάσαμε και προσεγγίζει τα όρια της περιοχής της Νέας Σμύρνης. Τα αποτελέσματα του ερωτήματος δίνονται παρακάτω στο σχήμα 13 .

5.1.3 Μεταμεσονύκτια λεωφορεία το Σάββατο-Κύριακο.

Το παρακάτω ερώτημα επιστρέφει τα *trips* λεωφορείων (μαζί με το όνομα των δρομολογίων αλλά και το μήνυμα πίνακα αφίξεως του *trip*) τα οποία φτάνουν στην στάση με αναγνωριστικό: 340027, μετά τις 12 το βράδυ και ημέρα Σάββατο ή Κυριακή.

```

PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```

SELECT ?trip_head ?route_name ?tim
WHERE {
  ?arr MoPT:arrivalHasStop MoPT:stop_340027 .
  ?arr MoPT:arrival_time ?tim .
}

```

stop_address	stop_name
"Επί της ΧΡ.ΣΜΥΡΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΣΤΡΟΦΗ Ν.ΣΜΥΡΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της Κ.ΠΑΛΑΙΟΔΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΔΗΜΑΡΧΕΙΟ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΜΕΓ.ΑΛΕΞΑΝΔΡΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΠΕΡΙΜΕΝΗ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΥΞΕΙΝΟΥ ΠΟΝΤΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΣΧΟΛΕΙΑ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΠΕΡΙΜΕΝΗ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΜΕΓ.ΑΛΕΞΑΝΔΡΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΛΕΩΦ.ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΓ.ΦΩΤΕΙΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΠΛ.Ν.ΣΜΥΡΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΓ.ΦΩΤΕΙΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΘΡΑΚΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΛΕΩΦ.ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΑΓΙΑΣ ΦΩΤΕΙΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΕΥΞΕΙΝΟΥ ΠΟΝΤΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΔΡΑΓΑΤΣΙΑΝΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"3η ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"4η ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"5η ΑΙΓΑΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>
"Επί της ΛΕΩΦ.ΕΛ.ΒΕΝΙΖΕΛΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"ΜΕΓΑΛΟΥ ΑΛΕΞΑΝΔΡΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>

Σχήμα 13: Μέρος αποτελεσμάτων του query2.txt .

```
?arr MoPT:arrivalHasTrip ?trip .
?trip MoPT:trip_day ?day .
?trip MoPT:trip_headsign ?trip_head .
?trip MoPT:hasRoute ?route .
?route MoPT:route_short ?route_name .
FILTER (?tim >= "24:00:00"^^xsd:time &&
(?day = "Savvato"^^xsd:string || ?day = "Kyriaki"^^xsd:string) )
}
```

Σχεδιαγραμματικά, στο ερώτημα αυτό βρίσκουμε όλες τις αφίξεις στην στάση με το δουέν αναγνωριστικό και φροντίζουμε οι αφίξεις να αντιστοιχούν σε ταξίδι που γίνεται το Σαββατοκύριακο και η άφιξη να γίνεται μετά τα μεσάνυχτα. Ακόμη μέσω του ταξιδιού, βρίσκουμε το αντίστοιχο δρομολόγιο και από εκεί το όνομα του δρομολογίου. Τα αποτελέσματα του ερωτήματος βρίσκονται στο Σχήμα 14

trip_head	route_name	tim
"ΣΤ. ΔΑΦΝΗΣ - ΠΕΙΡΑΙΑΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"218"^^<http://www.w3.org/2001/XMLSchema#string>	"24:02:00"^^<http://www.w3.org/2001/XMLSchema#time>
"ΣΤ. ΔΑΦΝΗΣ - ΠΕΙΡΑΙΑΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"218"^^<http://www.w3.org/2001/XMLSchema#string>	"24:02:00"^^<http://www.w3.org/2001/XMLSchema#time>

Σχήμα 14: Αποτελέσματα του query3.txt .

5.1.4 Προσβάσιμες στάσεις χωρίς καμία αλλαγή μέσου.

Αυτό το ερώτημα βρίσκει τις προσβάσιμες - από την στάση με αναγνωριστικό 340027 - στάσεις όταν δεν γίνεται αλλαγή μέσου, δηλαδή μόνο μέσω ενός στιγμιότυπου της κλάσης *trips*.

PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>

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

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

SELECT SAMPLE(?stop_name) SAMPLE(?stop_location)
WHERE {
  ?arr MoPT:arrivalHasStop MoPT:stop_340027 .
  ?arr MoPT:arrivalHasTrip ?trip .
  ?arr1 MoPT:arrivalHasTrip ?trip .
  ?arr1 MoPT:arrivalHasStop ?stop .
  ?stop MoPT:stop_id ?stop_id .
  ?stop MoPT:stop_name ?stop_name .
  ?stop MoPT:stop_location ?stop_location .
FILTER (?stop_id != "340027"^^xsd:int)

}
GROUP BY ?stop_id

```

Το αποτέλεσμα του ερωτήματος δίνεται στο Σχήμα 15

"4η ΑΙΓΑΙΟΥ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.947942682362 23.722168986674)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"7η ΑΙΓΑΙΟΥ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.94387263994 23.727647244211)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΜΙΑΔΥΝΗ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.955693433843 23.682823156854)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΗΣΑΠ Ν.ΦΑΛΗΡΟΥ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.944788190102 23.66665176887)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΙΚΑ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.95157435189 23.707885149132)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΠΑΘΜΑΡΙΟΥ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.957545555502 23.691284301927)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΑΓ. ΣΟΕΤΗΣ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.95609213823 23.715376794925)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΠΑΛ. ΤΕΡΜΑ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.941864760543 23.717036366599)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΔΕΚΑΕΠΤΑ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.946147437738 23.659759853414)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΑΝΟΙΣΗ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.93542635649 23.715718231328)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΤΑΣΙΑΡΧΩΝ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.956220889878 23.687954624823)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΚΑΤΡΙΒΑΝΟΥ^^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.949665563207 23.730528273015)^^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>

Σχήμα 15: Μέρος αποτελεσμάτων του query4.txt .

5.1.5 Προσβάσιμες στάσεις με μία αλλαγή μέσου.

Αυτό το ερώτημα βρίσκει τις προσβάσιμες - από την στάση με αναγνωριστικό 340027 - στάσεις όταν γίνεται μια ακριβώς αλλαγή μέσου, δηλαδή μόνο μέσω 2 στιγμιοτύπων της κλάσης *trips*.

```

PREFIX MoPT: <http://www.semanticweb.org/knowsproject/ontologies/MoPT#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```

SELECT SAMPLE(?stop_name) SAMPLE(?stop_location)
WHERE {
  { SELECT SAMPLE(?stop) AS ?first_stop
    WHERE {
      ?arr MoPT:arrivalHasStop MoPT:stop_340027 .
      ?arr MoPT:arrivalHasTrip ?trip .
      ?arr1 MoPT:arrivalHasTrip ?trip .
      ?arr1 MoPT:arrivalHasStop ?stop .
      ?stop MoPT:stop_id ?stop_id .
      ?stop MoPT:stop_name ?stop_name .
      ?stop MoPT:stop_location ?stop_location .
    }
  }
}

```

```

FILTER (?stop_id != "340027"^^xsd:int)

}
GROUP BY ?stop_id }

?arr MoPT:arrivalHasStop ?first_stop .
?arr MoPT:arrivalHasTrip ?trip .
?arr1 MoPT:arrivalHasTrip ?trip .
?arr1 MoPT:arrivalHasStop ?stop .
?stop MoPT:stop_id ?stop_id .
?stop MoPT:stop_name ?stop_name .
?stop MoPT:stop_location ?stop_location .
FILTER (?stop_id != "340027"^^xsd:int )

}
GROUP BY ?stop_id

```

Το ερώτημα βρίσκει πρώτα τις στάσεις από το πρώτο μέσο και μετά κάνει την ίδια διαδικασία με πριν ώστε να βρει τις στάσεις που προκύπτουν μέσω ταξιδιού με το δεύτερο μέσο από τις στάσεις που έχουν προκύψει με το πρώτο μέσο. Τα αποτελέσματα βρίσκονται στο Σχήματα 16 και προφανώς είναι αρκετά περισσότερα από αυτά του προηγούμενου ερωτήματος.

"ΕΧΘΑΕΙΟ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.955738489212 23.686180928059)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΣΙΔΕΡΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(38.017843427025 23.79542143528)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΖΗΝΩΝΟΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.885621241197 23.877928884226)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΚΟΥΝΤΟΥΡΙΟΥ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.931941089629 23.720624738639)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΕΚΚΡΑΤΟΥΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.848088123521 23.764156617995)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΕΧΘΑΕΙΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.974424026 23.625221476435)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΑΓ. ΙΩΑΝΝΗΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.841305002089 23.765462246574)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΣΙΡΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.933067004 23.629038806433)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΕΧΘΑΕΙΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.952731094611 23.727113714455)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"Κ.Ε.Α.Τ. ΠΡ.ΟΙΚΟΣ ΤΥΦΛΩΝ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.952277237847 23.696371517674)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΖΕΦΥΡΟΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(38.048386313055 23.761190210495)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΑΘΗΝΑΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.945980624825 23.701223155221)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΣΧΟΛΗ ΚΑΘΗΓΗΤΩΝ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.962375153291 23.690785959057)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΝΕΡΑΙΔΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.917021801739 23.704685154995)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΠΑΝΑΓΙΤΣΑ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.965045061325 23.698538904635)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>
"ΠΗΛΕΟΣ"^^<http://www.w3.org/2001/XMLSchema#string>	"POINT(37.938435483844 23.730535137246)"^^<http://www.openlinksw.com/schemas/virtrdf#Geometry>

Σχήμα 16: Μέρος αποτελεσμάτων του query5.txt .

Στην περίπτωση που θέλουμε να βρούμε τις προσβάσιμες στάσεις όταν έχουμε το πολύ μία αλλαγή μέσου αρκεί να κάνουμε ένα *UNION* των αποτελεσμάτων των ερωτημάτων query4, query5

5.2 Ερωτήματα χωρίς διαθέσιμα δεδομένα απάντησης στην βάση δεδομένων

5.2.1 Πτήσεις Αθήνα-Σκόπια

Αυτό το ερώτημα επιστρέφει όλες τις πτήσεις (μαζί με τις ώρες απογείωσης και προγείωσης) που ξεκινούν από την Αθήνα, Ελλάδα και καταλήγουν στα Σκόπια, Βόρεια Μακεδονία. Το ερώτημα βρίσκει αρχικά όλες τις στάσεις στις ευρύτερες περιοχές της Αθήνας και των Σκοπίων και τα ταξίδια που συνδέουν τις στάσεις αυτές. Στην συνέχεια, για να αναγκάσουμε οι πτήσεις να είναι από Αθήνα σε Σκόπια φιλτράρουμε εκτός

όλες τα ταξίδια όπου η σειρά άφιξης στην Αθήνα είναι μεγαλύτερη από την σειρά άφιξης στα Σκόπια. Το ερώτημα δίνεται παρακάτω:

PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>

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

PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```
SELECT ?trip ?trip_head ?arr_time ?dep_time
WHERE {
```

```
  { #find all stops in Athens
```

```
    SELECT ?stop AS ?stop_ath
```

```
    WHERE {
```

```
      ?stop rdf:type MoPT:stops .
```

```
      ?stop MoPT:stop_location ?stop_loc .
```

```
      ?stop MoPT:stop_address ?stop_address .
```

```
      ?stop MoPT:stop_name ?stop_name .
```

```
      FILTER(bif:st_intersects(?stop_loc , bif:st_geomfromtext ("POLYGON ((38.075609 23.512203,
38.126833 23.756285, 38.153746 23.877452,
38.041231 23.999741, 37.813518 24.051387,
37.802982 23.773815, 37.939056 23.622694,
38.075609 23.512203)))"))))
    }
```

```
  }
```

```
  { #find all stops in Skopje
```

```
    SELECT ?stop AS ?stop_skopj
```

```
    WHERE {
```

```
      ?stop rdf:type MoPT:stops .
```

```
      ?stop MoPT:stop_location ?stop_loc .
```

```
      ?stop MoPT:stop_address ?stop_address .
```

```
      ?stop MoPT:stop_name ?stop_name .
```

```
      FILTER(bif:st_intersects(?stop_loc , bif:st_geomfromtext (
"POLYGON ((42.053499 21.340780, 42.041020 21.449740,
41.980602 21.653567, 41.946793 21.614481,
41.931839 21.529338, 42.004092 21.318612,
42.053499 21.340780 )))"))))
    }
```

```
  }
```

```
#find corresponding trip between athens and skopje
```

```
?arr1 MoPT:arrivalHasStop ?stop_ath .
```

```
?arr2 MoPT:arrivalHasStop ?stop_skopj .
```

```
?arr1 MoPT:arrivalHasTrip ?trip_ath_skopj .
```

```
?arr2 MoPT:arrivalHasTrip ?trip_ath_skopj .
```

```
#find corresponding trips route
```

```
#and assert it is a airplane route
```

```
?trip_ath_skopj MoPT:hasRoute ?route_ath_skopj .
```

```
?route_ath_skopj rdf:type MoPT:airplane_route .
```

```
?trip MoPT:trip_head ?trip_head
```

```
#find sequence of arrivals
```

```

?arr1 MoPT:arrival_seq ?seq_ath .
?arr2 MoPT:arrival_seq ?seq-skopj .

#find departure and arrival time
?arr1 MoPT:departure_time ?dep_time .
?arr2 MoPT:arrival_time ?arr_time .

#the flight must be from Athens to Skopje
FILTER (?seq-skopj > ?seq-ath)

}

```

Εφόσον δεν διαθέτουμε στην βάση δεδομένα για αεροπορικές πτήσεις, το ερώτημα παρατίθεται χωρίς τα αποτελέσματά του.

5.2.2 RedCube ferries to Σάββατο

Αυτό το ερώτημα επιστρέφει όλα τα ταξίδια με πλοίο τα οποία παρέχονται από το πρακτορείο RedCube και ξεκινούν μέρα Σάββατο. Ο κώδικας του ερωτήματος δίνεται παρακάτω:

```

PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```

SELECT ?trip_head AS ?trip_description
WHERE {

    ?trip rdf:type MoPT:trips .
    ?trip MoPT:hasRoute ?route .
    ?route rdf:type MoPT:ferry_route .

    ?trip MoPT:trip_day ?day .
    ?trip MoPT:provideBy ?agency .
    ?agency MoPT:agency_name ?ag_nam .
    FILTER(?day = "Savvato"^^xsd:string && regex(?ag_nam, "RedCube"))
}

```

Επειδή δεν διαθέτουμε στην βάση δεδομένα για δρομολόγια πλοίων, το ερώτημα παρατίθεται χωρίς τα αποτελέσματά του.

5.2.3 Κατεστραμμένα λεωφορεία

Αυτό το ερώτημα επιστρέφει το πλήθος των λεωφορείων που ανήκουν στον πάροχο με αναγνωριστικό: 42 και που χρησιμοποιούνται σε δρομολόγια λεωφορείων και τα οποία είναι εκτός υπηρεσίας λόγω μόνιμης/σοβαρής βλάβης. Το ερώτημα δίνεται παρακάτω:

```

PREFIX MoPT: <http://www.semanticweb.org/knowsys_project/ontologies/MoPT#>
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>

```

```

SELECT COUNT(*)
WHERE {

    ?veh rdf:type MoPT:vehicles .
    ?veh MoPT:vehicle_status ?veh_stat .

    ?ag MoPT:ownsVehicle ?veh .

```



```

?ag MoPT:agency_id ?ag_id .

?trip MoPT:usesVehicle ?veh .
?trip MoPT:hasRoute ?route .
?route rdf:type MoPT:bus_route

FILTER(?veh_stat = 2 && ag_id = 42 )
}

```

6 Σχολιασμός συστήματος και περιορισμών του

Η οντολογία που αναπτύξαμε φαίνεται να έχει αρκετή εκφραστικότητα για την περιγραφή δεδομένων που αφορούν τα δρομολόγια λεωφορείων στην Αθήνα αλλά επίσης φαίνεται να μπορεί να γενικευτεί ικανοποιητικά και σε άλλα μέσα μεταφοράς. Όμως, η σχεδιαστική απόφαση να διαχωριστούν τα δρομολόγια από τα ταξίδια οδηγεί σε αρκετά μεγαλύτερα αρχεία τριάδων RDF που πιθανόν να οδηγεί σε μη ικανοποιητικές αποδόσεις όταν έχουμε μεγαλύτερη κλίμακα δεδομένων. Σε αυτές τις περιπτώσεις μια διαφορετική αρχιτεκτονική της οντολογίας ενδεχομένως να ήταν προτιμητέα. Σημειώνουμε πως για τον παραπάνω λόγο και επιπλέον λόγω του ορίου μεγέθους αρχείου που επιβάλλει η πλατφόρμα *mycourses.ntua* το παραδοθέν αρχείο θα περιέχει υποσύνολα των αρχείων τριάδων που αφορούν τις κλάσεις *trips* και *arrivals*. Για να αναπαραχθούν τα *ABOXes* βάσει των οποίων προκύψαν τα αποτελέσματα του μέρους 5 αρκεί να εκτελεστούν οι συνταγές

```

make download
make Abox

```

του αρχείου *Makefile*.