

Συστήματα και Τεχνολογίες Γνώσης

Θέμα

Βεκράκης Εμμανουήλ

03116068



Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών
και Μηχανικών Υπολογιστών
Ιούλιος 2020



Γενικές πληροφορίες για το θέμα

Στο θέμα του μαθήματος κληθήκαμε να σχεδιάσουμε και να κατασκευάσουμε μία σημασιολογική βάση γνώσης, η οποία θα αφορά σε δεδομένα μετακίνησης με μέσα μεταφοράς. Συγκεκριμένα η βάση γνώσης χρειάζεται να καλύπτει πλήρως τις ανάγκες περιγραφής όλων των μέσων μεταφοράς και των πληροφοριών που αυτά απαιτούν, όπως αστικά μέσα μεταφοράς, πλοία, αεροπλάνα, τραίνα, δρομολόγια, χρόνους άφιξης στους σταθμούς και άλλα. Η οντολογία θα πρέπει να είναι σε μορφή OWL2 και το λεξιλόγιο της να καλύπτει τις απαιτούμενες εκφραστικές δυνατότητες. Μετά την ανάπτυξη της, κληθήκαμε να την εμπλουτίσουμε με δεδομένα, τα οποία μετατρέψαμε σε τριάδες RDF και χάριν ευκολίας περιορίστηκαν στις αστικές συγκοινωνίες της Αθήνας, καθώς και να επιδείξουμε τις δυνατότητες της μέσω μερικών χρήσιμων ερωτημάτων SPARQL. Όλα τα παραπάνω παρουσιάζονται αναλυτικότερα στη συνέχεια, μαζί με τυχούσες προκλήσεις που αντιμετωπίσαμε στο κάθε μέρος.

Κατασκευή οντολογίας μέσων μεταφοράς

Όπως αναφέρθηκε προηγουμένως, η οντολογία που θα κατασκευάσουμε πρέπει να καλύπτει ολόκληρο το φάσμα των μέσων μεταφοράς. Έτσι, θα πρέπει σίγουρα να συμπεριλάβουμε μία κλάση που θα έχει ως υποκλάσεις όλα τα μέσα μεταφοράς. Αυτή την κλάση την ονομάσαμε Means, με τις υποκλάσεις Airplane, Bus, IntercityBus, Metro, Ship, Tram και Trolley. Όλα αυτά τα μέσα προφανώς, κάνουν κάποιες στάσεις (Stops). Ανάλογα με το μέσο όμως, η στάση μπορεί να είναι αεροδρόμιο (Airport), στάση λεωφορείου (BusStop), μετρό (MetroStation) ή λιμάνι (Port). Δημιουργήσαμε λοιπόν την κλάση Stop με υποκλάσεις όλες τις παραπάνω κατηγορίες στάσεων. Τέλος, για όλα τα μέσα μεταφοράς έχουμε κάποιες πληροφορίες (Info). Αυτές συγκεκριμένα είναι το δρομολόγιο που ακολουθούν (Route), η διαδρομή που ακολουθεί αυτό το δρομολόγιο (Trip) και η υπηρεσία που εκτελούν (Service). Σε αυτό το σημείο εντοπίσαμε το πρώτο πρόβλημα, το οποίο έχει να κάνει με το ότι ένα μεταφορικό μέσο, φτάνει σε μία συγκεκριμένη στάση μία συγκεκριμένη ώρα όταν εκτελεί μία συγκεκριμένη διαδρομή. Σε μία άλλη διαδρομή δηλαδή του ίδιου δρομολογίου η ώρα που θα φτάσει στην ίδια στάση θα είναι διαφορετική, ενώ μπορεί οι στάσεις που κάνει να μεταβάλλονται. Για να αποτυπώσουμε αυτή την τριπλή σχέση, χρειάστηκε να κατασκευάσουμε μία επιπλέον κλάση, την οποία ονομάσαμε Arrival και αντιπροσωπεύει την άφιξη του μέσου μεταφοράς σε μία στάση κάποια συγκεκριμένη χρονική στιγμή. Έτσι προέκυψε η ιεραρχία κλάσεων που φαίνεται στην παρακάτω εικόνα. Προφανώς, όλες οι υποκλάσεις είναι ξένες (disjoint) με τις υπόλοιπες.

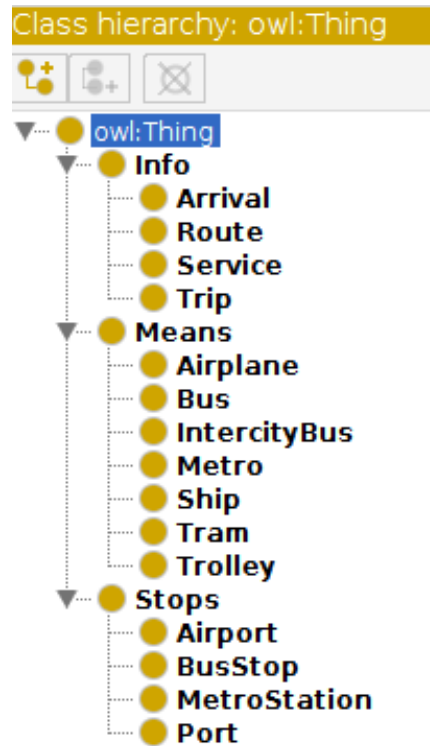


Image 1: Class Hierarchy

Σχετικά με τις ιδιότητες των αντικειμένων, μπορούμε εύκολα από τον τρόπο με τον οποίο χωρίσαμε τις κλάσεις να δημιουργήσουμε τις ιδιότητες που χρειάζεται η οντολογία μας, καθώς και το πεδίο ορισμού (Domain) και το σύνολο τιμών (Range) αυτών. Αρχικά, ένα μέσο μεταφοράς εκτελεί ένα δρομολόγιο, επομένως τα αντικείμενα των κλάσεων **Mean** και **Route** συνδέονται μέσω της ιδιότητας **hasRoute**. Το κάθε δρομολόγιο συνδέεται με τη σειρά του μέσω των **hasTrip** και **hasService** με τη διαδρομή και την υπηρεσία που εκτελεί. Μία διαδρομή με τη σειρά της, πραγματοποιεί κάποια άφιξη, οπότε συνδέεται με ένα αντικείμενο της **Arrival** μέσω της **ArrivesAt**, το οποίο με τη σειρά του συνδέεται με τη στάση (Stop) στην οποία φτάνει, μέσω της **arrivalPlace**.

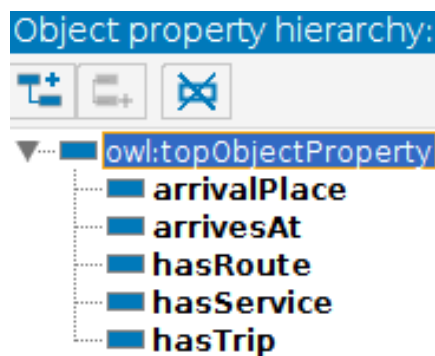


Image 2: Object Properties Hierarchy

Τέλος, για την κατασκευή της οντολογίας χρειάστηκε να ορίσουμε τις ιδιότητες τύπου δεδομένων, που αφορούν ουσιαστικά τις πληροφορίες των μέσων μεταφοράς. Συνεχίζοντας με την ίδια σειρά με πριν, ένα δρομολόγιο (Route) χαρακτηρίζεται από το όνομα του, δηλαδή ένα xsd:string με το οποίο συνδέεται μέσω της ιδιότητας routeName. Αντίστοιχα και η διαδρομή (Trip) έχει ένα όνομα στην πινακίδα του μέσου μεταφοράς, με το οποίο συνδέεται μέσω της hasHeadsign. Η υπηρεσία που εκτελεί κάποιο μέσο μεταφοράς, μπορεί να λειτουργεί συγκεκριμένες μόνο μέρες. Έτσι, δημιουργήσαμε μία ιδιότητα worksOnXday για κάθε μέρα της εβδομάδας, με πεδίο ορισμού ένα Service και σύνολο τιμών xsd:boolean, που είναι true αν εκτελείται εκείνη τη μέρα. Χάριν ευκολίας στην παρουσίαση, τις παραπάνω ιδιότητες τις ορίσαμε ως υποιδιότητες της ιδιότητας worksOn. Η άφιξη (Arrival) ενός μέσου σε μία στάση χαρακτηρίζεται από τον χρόνο άφιξης, αναχώρησης και τη σειρά της στάσης αυτής στο δρομολόγιο του μέσου. Έτσι, δημιουργήσαμε τις ιδιότητες arrivalTime, departureTime και arrivalSequence που συνδέουν ένα αντικείμενο Arrival με τα αντίστοιχα δεδομένα. Σημειώνεται ότι για τους χρόνους δεν προσδιορίσαμε σύνολο τιμών καθώς κανένα από τα διαθέσιμα δεν μας κάλυπτε, ενώ για το arrivalSequence το σύνολο τιμών είναι προφανώς xsd:integer. Τέλος, κάθε στάση μέσου μεταφοράς χαρακτηρίζεται από το όνομα της, μία περιγραφή για αυτή και την γεωγραφική τοποθεσία της. Έτσι δημιουργήσαμε τις ιδιότητες stopName, stopDescription (και τα δύο με σύνολο τιμών xsd:string) και stopLocation. Για την τελευταία ιδιότητα δεν προσδιορίσαμε κάποιο σύνολο τιμών, καθώς για τις ανάγκες προσδιορισμού των γεωγραφικών συντεταγμένων της κάθε στάσης θα χρησιμοποιήσουμε το πρότυπο GeoSPARQL και συγκεκριμένα τον τύπο δεδομένων:

<http://www.openlinksw.com/schemas/virttrdf#Geometry>, ο οποίος δεν ήταν διαθέσιμος στο Protege.

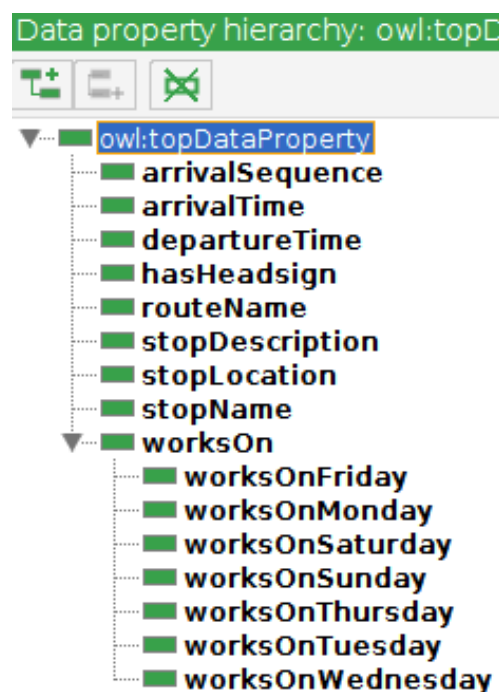


Image 3: Data Properties Hierarchy

Μετατροπή δεδομένων σε τριάδες RDF

Για τον εμπλουτισμό της οντολογίας με δεδομένα (ABox), περιοριστήκαμε σε δεδομένα για τα αστικά μέσα μεταφοράς στην Αθήνα, τα οποία ήταν διαθέσιμα σε μορφή πίνακα. Επομένως, για την μετατροπή των δεδομένων αυτών σε τριάδες RDF χρειάστηκε να γράψουμε κώδικα, ο οποίος θα παράξει τις τριάδες σε σύνταξη Turtle. Για τη μετατροπή αυτή χρησιμοποιήθηκε η Java και η βιβλιοθήκη Apache Jena. Έτσι, για κάθε σειρά από τα δεδομένα μας, δημιουργούσαμε με τα κατάλληλα πεδία τα επιθυμητά στιγμιότυπα κλάσεων και ιδιοτήτων και εξάγαμε το προκύπτον μοντέλο σε ένα ttl αρχείο (ξεχωριστό για κάθε αρχείο δεδομένων). Από τα διαθέσιμα δεδομένα, χρησιμοποιήσαμε όσα αφορούσαν την οντολογία που δημιουργήσαμε προηγουμένως. Δηλαδή, από το `calendar.txt` λάβαμε πληροφορίες για το ποιες μέρες εκτελείται η αντίστοιχη υπηρεσία, από το `routes.txt` πληροφορίες για τις διαδρομές, το όνομα τους και τον τύπο τους (900 για Tram, 1 για Metro, 800 για Trolley και 3 για Bus) και από το `stops.txt` λάβαμε πληροφορίες για τις στάσεις, το όνομα, την περιγραφή και την γεωγραφική τους θέση. Για την αναπαράσταση της γεωγραφικής θέσης των στάσεων χρησιμοποιήθηκε το πρότυπο GeoSPARQL όπως φαίνεται και στο αντίστοιχο δείγμα δεδομένων παρακάτω. Τέλος, από το `stop_times.txt` λάβαμε πληροφορίες για την ώρα άφιξης και αναχώρησης του κάθε Trip από τις στάσεις του και τη σειρά που κατέχει η στάση αυτή στη διαδρομή, ενώ με το `trips.txt` καταφέραμε να συνδέσουμε το κάθε Route, με το Service και το Trip που εκτελεί. Σημειώνεται ότι τα Arrivals ονομάστηκαν σύμφωνα με το πρότυπο "TripId"_ "StopId", καθώς κάθε Arrival χαρακτηρίζεται μοναδικά από τη διαδρομή και τη στάση στην οποία καταφθάνει το μέσο. Επίσης, πριν από τη δημιουργία των RDF τριάδων, μετατρέψαμε την ταυτότητα των Services των Σαββατοκύριακων από Σαβ/Κυρ σε Σαβ-Κυρ, προκειμένου να μην υπάρξει σύγχυση με το / των URIs. Ένα πρόβλημα που αντιμετωπίσαμε ήταν ότι το πολύ μεγάλο μέγεθος του αρχείου `stop_times.txt` ήγειρε σφάλμα κατά τη μετατροπή του σε RDF στο περιβάλλον Eclipse, οπότε χρειάστηκε να το χωρίσουμε σε 4 μικρότερα αρχεία, παράγοντας αντίστοιχα 4 αρχεία με τριάδες RDF. Παρακάτω φαίνονται ενδεικτικά αποσπάσματα από τα αρχεία που παράχθηκαν, ενώ οι κλάσεις Java περιέχονται στα παραδοτέα της άσκησης (λόγω του απαγορευτικά μεγάλου μεγέθους των αρχείων που αφορούν στο `stop_times.txt`, συμπεριλήφθηκε μόνο το τελευταίο και μικρότερο κομμάτι του ttl αρχείου που παράχθηκε).

```

1  <http://example.com/transportation#CALEND-036-Καθημερινή-12>
2      <http://example.com/transportation#worksOnFriday>
3          true ;
4      <http://example.com/transportation#worksOnMonday>
5          true ;
6      <http://example.com/transportation#worksOnSaturday>
7          false ;
8      <http://example.com/transportation#worksOnSunday>
9          false ;
10     <http://example.com/transportation#worksOnThursday>
11         true ;
12     <http://example.com/transportation#worksOnTuesday>
13         true ;
14     <http://example.com/transportation#worksOnWednesday>
15         true .
16
17     <http://example.com/transportation#CALEND-906-Κυριακή-05>
18         <http://example.com/transportation#worksOnFriday>
19             false ;
20         <http://example.com/transportation#worksOnMonday>
21             false ;
22         <http://example.com/transportation#worksOnSaturday>
23             false ;
24         <http://example.com/transportation#worksOnSunday>
25             true ;
26         <http://example.com/transportation#worksOnThursday>
27             false ;
28         <http://example.com/transportation#worksOnTuesday>
29             false ;
30         <http://example.com/transportation#worksOnWednesday>
31             false .

```

Turtle sample 1: Calendar

```

<http://example.com/transportation#420140Point>
  <http://www.w3.org/2003/01/geo/wgs84_pos#geometry>
    "point(38.0048782822674 23.686825901861)"^^<http://www.openlinksw.com/schemas/virttrdf#Geometry> .

<http://example.com/transportation#061321>
  a      "http://example.com/transportation#Stop" ;
  <http://example.com/transportation#stopDescription>
    "Επί της ΠΡΙΓΚΙΠΟΝΝΗΣΩΝ" ;
  <http://example.com/transportation#stopLocation>
    <http://example.com/transportation#061321Point> ;
  <http://example.com/transportation#stopName>
    "ΑΦΕΘΡΙΑ" .

```

Turtle sample 2: Stops

```

<http://example.com/transportation#15>
  a      "http://example.com/transportation#Trolley" ;
  <http://example.com/transportation#hasRoute>
    <http://example.com/transportation#15-20> .

<http://example.com/transportation#831>
  a      "http://example.com/transportation#Bus" ;
  <http://example.com/transportation#hasRoute>
    <http://example.com/transportation#831-20> .

<http://example.com/transportation#040-20>
  <http://example.com/transportation#routeName>
    "ΠΕΙΡΑΙΑΣ - ΣΥΝΤΑΓΜΑ" .

<http://example.com/transportation#M2-20>
  <http://example.com/transportation#routeName>
    "ΑΝΘΟΥΠΟΛΗ - ΣΤ.ΕΛΛΗΝΙΚΟ" .

<http://example.com/transportation#M1>
  a      "http://example.com/transportation#Metro" ;
  <http://example.com/transportation#hasRoute>
    <http://example.com/transportation#M1-20> .

```

Turtle sample 3: Routes

```

<http://example.com/transportation#9787552-CALEND-A7-Κυριακή-05_690015>
  a      "http://example.com/transportation#Arrival" ;
  <http://example.com/transportation#arrivalPlace>
    <http://example.com/transportation#690015> ;
  <http://example.com/transportation#arrivalSequence>
    "18"^^<http://www.w3.org/2001/XMLSchema#int> ;
  <http://example.com/transportation#arrivalTime>
    "10:27:00" ;
  <http://example.com/transportation#departureTime>
    "10:27:00" .

<http://example.com/transportation#7050513-CALEND-B9-Σάββατο-03_270111>
  a      "http://example.com/transportation#Arrival" ;
  <http://example.com/transportation#arrivalPlace>
    <http://example.com/transportation#270111> ;
  <http://example.com/transportation#arrivalSequence>
    "9"^^<http://www.w3.org/2001/XMLSchema#int> ;
  <http://example.com/transportation#arrivalTime>
    "08:03:00" ;
  <http://example.com/transportation#departureTime>
    "08:03:00" .

```

Turtle sample 4: Stop times


```

<http://example.com/transportation#9259430-CALEND-B18-Γ18-Καθημερινή-11>
  a      "http://example.com/transportation#Trip" ;
  <http://example.com/transportation#hasHeadsign>
    "ΟΜΟΝΟΙΑ - ΠΕΡΑΜΑ (Μέσω Π. ΡΑΛΛΗ)" .

<http://example.com/transportation#9606187-ΧΗΛΠΑΠ-2-4-Κυριακή-06>
  a      "http://example.com/transportation#Trip" ;
  <http://example.com/transportation#hasHeadsign>
    " ΣΤ. ΜΕΤΡΟ ΑΓ. ΙΩΑΝΝΗΣ - ΑΓ. ΑΡΤΕΜΙΟΣ - ΑΝΩ ΚΥΨΕΛΗ " .

<http://example.com/transportation#9758916-ΧΗΛΠΑΠ-24-Σάββατο-05>
  a      "http://example.com/transportation#Trip" ;
  <http://example.com/transportation#hasHeadsign>
    "ΠΕΤΡΟΥΠΟΛΗ - ΙΛΙΟΝ - ΣΤ. ΑΓ. ΑΝΤΩΝΙΟΣ (ΚΥΚΛΙΚΗ)" .

```

Turtle sample 5: Trips

Κατασκευή και εκτέλεση ερωτημάτων SPARQL

Η οντολογία από το Protege και τα αρχεία που παράχθηκαν παραπάνω, φορτώθηκαν στην αποθήκη τριάδων του Virtuoso μέσω του Web UI, με στόχο την εκτέλεση ερωτημάτων SPARQL επί των διαθέσιμων δεδομένων. Ο σκοπός των ερωτημάτων είναι η επίδειξη των δυνατοτήτων της βάσης γνώσης, οπότε για τον σχεδιασμό τους σκεφτήκαμε όπως ένας χρήστης μιας εφαρμογής που θα αξιοποιούσε τα δεδομένα αυτά. Αρχικά, ο χρήστης μπορεί να αναζητήσει τις στάσεις που κάνει το δρομολόγιο ενός συγκεκριμένου μέσου μεταφοράς. Στο παράδειγμα αναζητούμε στοιχεία για το λεωφορείο 608 και όπως φαίνεται, εμφανίζουμε για τις διαδρομές που εκτελεί, τις στάσεις, καθώς και την σειρά της κάθε στάσης στην εκάστοτε διαδρομή. Παρακάτω φαίνεται το query και μέρος του αποτελέσματός του για τις δύο κατευθύνσεις του δρομολογίου του.

PREFIX ex: <http://example.com/transportation#>

```

SELECT DISTINCT ?headsign (xsd:integer(?seq) AS ?Sequence) ?stopName WHERE {
  <http://example.com/transportation#608> ex:hasRoute ?route .
  ?route ex:routeName ?name .
  ?route ex:hasTrip ?trip .
  ?trip ex:arrivesAt ?arrival .
  ?trip ex:hasHeadsign ?headsign .
  ?arrival ex:arrivalPlace ?stop .
  ?arrival ex:arrivalSequence ?seq .
  ?stop ex:stopName ?stopName .
}
ORDER BY ?trip (?seq)

```


Query

Saved Queries

Default Graph IRI
http://example.com/transportation

Query

```

PREFIX ex: <http://example.com/transportation#>

SELECT DISTINCT ?headsign (xsd:integer(?seq) AS ?Sequence) ?stopName WHERE
{
  <http://example.com/transportation#608> ex:hasRoute ?route .
  ?route ex:routeName ?name .
  ?route ex:hasTrip ?trip .
  ?trip ex:arrivesAt ?arrival .
  ?trip ex:hasHeadsign ?headsign .
  ?arrival ex:arrivalPlace ?stop .
  ?arrival ex:arrivalSequence ?seq .
  ?stop ex:stopName ?stopName .
}
ORDER BY ?trip (?seq)

```

Execute

Save

Load

Clear

headsign	Sequence	stopName
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	1	"ΑΦΕΤΗΡΙΑ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	2	"ΑΓ.ΕΙΡΗΝΗ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	3	"ΧΡΙΣΤΙΑΝΟΥΠΟΛΕΩΣ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	4	"ΠΥΘΙΑΣ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	5	"ΕΣΠΕΡΙΔΩΝ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	6	"ΜΑΥΡΟΓΕΝΟΥΣ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	7	"ΠΑΛ.ΤΕΡΜΑ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	8	"ΣΥΚΙΑ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	9	"ΛΙΝΑΡΑ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	10	"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	11	"ΧΑΤΖΗΔΑΚΗ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	12	"ΚΩΣΤΗ ΠΑΛΑΜΑ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	13	"ΚΛΩΝΑΡΙΔΟΥ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	14	"ΑΓ.ΛΟΥΚΑΣ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΣΩΓΡΑΦΟΥ"	15	"ΠΛΑΤΕΙΑ ΚΟΛΙΑΤΣΟΥ"

Query 1: 608 stops (1)

"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΖΩΓΡΑΦΟΥ"	38	"7η ΖΩΓΡΑΦΟΥ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΖΩΓΡΑΦΟΥ"	39	"8η ΖΩΓΡΑΦΟΥ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΖΩΓΡΑΦΟΥ"	40	"9η ΖΩΓΡΑΦΟΥ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΖΩΓΡΑΦΟΥ"	41	"10η ΖΩΓΡΑΦΟΥ"
"ΓΑΛΑΤΣΙ - ΑΚΑΔΗΜΙΑ - ΝΕΚΡ. ΖΩΓΡΑΦΟΥ"	42	"ΤΕΡΜΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	1	"ΤΕΡΜΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	2	"10η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	3	"9η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	4	"8η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	5	"7η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	6	"6η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	7	"ΤΣΙΤΟΥΡΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	8	"ΚΟΤΟΠΟΥΛΗ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	9	"2η ΖΩΓΡΑΦΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	10	"ΙΚΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	11	"ΑΛΕΞ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	12	"ΚΑΦΕΝΕΙΟ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	13	"ΙΛΙΣΙΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	14	"ΝΟΣΟΚ.ΕΥΑΓΓΕΛΙΣΜΟΣ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	15	"ΡΗΓΙΛΛΗΣ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	16	"ΣΥΝΤΑΓΜΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	17	"ΑΚΑΔΗΜΙΑ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	18	"ΡΕΞ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	19	"ΠΛ.ΛΑΥΡΙΟΥ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	20	"ΠΟΛΥΤΕΧΝΕΙΟ"
" ΝΕΚΡ. ΖΩΓΡΑΦΟΥ - ΑΚΑΔΗΜΙΑ - ΓΑΛΑΤΣΙ "	21	"ΜΟΥΣΕΙΟ"

Query 1: 608 stops (2)

Στη συνέχεια ο χρήστης αυτός φτάνει σε κάποια από τις στάσεις και θέλει να δει τις ώρες στις οποίες περνάνε τα μέσα σε εκείνη τη στάση. Έτσι με παράδειγμα τη στάση ΑΝΟΙΞΗ, παρουσιάζουμε το όνομα της στάσης, την περιγραφή της και ποια μέσα και τι ώρες διέρχονται από τη στάση αυτή.

```
PREFIX ex: <http://example.com/transportation#>
```

```
SELECT DISTINCT ?name ?description ?bus ?time WHERE {
  ?stop ex:stopName "ΑΝΟΙΞΗ" .
  ?stop ex:stopName ?name .
  ?stop ex:stopDescription ?description .
  ?route ex:hasTrip ?trip .
  ?bus ex:hasRoute ?route .
  ?trip ex:arrivesAt ?arrival .
  ?arrival ex:arrivalPlace ?stop .
  ?arrival ex:arrivalTime ?time .
}
ORDER BY ?stop (?bus) (?time)
```

Query		Saved Queries	
Default Graph IRI		<input type="text" value="http://example.com/transportation"/>	
Query		<pre>PREFIX ex: <http://example.com/transportation#> SELECT DISTINCT ?name ?description ?bus ?time WHERE { ?stop ex:stopName "ΑΝΟΙΞΗ" . ?stop ex:stopName ?name . ?stop ex:stopDescription ?description . ?route ex:hasTrip ?trip . ?bus ex:hasRoute ?route . ?trip ex:arrivesAt ?arrival . ?arrival ex:arrivalPlace ?stop . ?arrival ex:arrivalTime ?time . } ORDER BY ?stop (?bus) (?time)</pre>	
Execute		Save Load Clear	
name	description	bus	time
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"05:43:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"05:46:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:08:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:11:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:32:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:33:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:40:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"06:52:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:01:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:16:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:17:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:18:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:33:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"07:37:00"

Query 2: ΑΝΟΙΞΗ times (1)

"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#235	"24:06:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"05:41:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"05:48:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"05:51:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:05:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:06:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:11:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:21:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:36:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:37:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:41:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"06:51:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:05:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:06:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:16:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:18:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:27:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:31:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:32:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:36:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:45:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:52:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:56:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"07:57:00"
"ΑΝΟΙΞΗ"	"Επί της ΛΕΩΦ.ΣΤΡΧΟΥ ΠΑΠΑΓΟΥ"	http://example.com/transportation#608	"08:01:00"

Query 2: ΑΝΟΙΞΗ times (2)

Μία πιο προχωρημένη αναζήτηση την οποία μπορεί να πραγματοποιήσει ο χρήστης της βάσης γνώσης είναι το πώς μπορεί να μεταβεί από μία στάση σε μία άλλη χρησιμοποιώντας ένα συγκεκριμένο αριθμό από μέσα. Στο παράδειγμα μας αναζητήσαμε τρόπους μετάβασης από τη στάση ΒΟΡΕΙΟΣ ΠΟΛΟΣ μέχρι τη στάση ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ με την αλλαγή δύο μέσων (καθώς τα περισσότερα αποτελούν γενίκευση του ερωτήματος των δύο μέσων). Έτσι αναζητούμε δύο μέσα που να κάνουν και τα δύο στάση σε μία κοινή ενδιάμεση στάση και το ένα να σταματάει στην αρχική, ενώ το άλλο στην τελική στάση. Αποφασίσαμε να παρουσιάζουμε σε κάθε περίπτωση τα δύο μέσα που χρειάζεται να χρησιμοποιήσει ο χρήστης, καθώς και την αρχική, τελική και ενδιάμεση στάση.

PREFIX ex: <http://example.com/transportation#>

```
SELECT DISTINCT (?stopName1 AS ?start) ?bus1 (?stopName2 AS ?change) ?bus2 (?stopName3 AS ?destination)
WHERE {
  ?bus1 ex:hasRoute ?route1 .
  ?route1 ex:hasTrip ?trip1 .
  ?trip1 ex:arrivesAt ?arrival1 .
  ?arrival1 ex:arrivalPlace ?stop1 .
  ?stop1 ex:stopName ?stopName1 .
  ?stop1 ex:stopName "ΒΟΡΕΙΟΣ ΠΟΛΟΣ" .

  ?bus2 ex:hasRoute ?route2 .
  ?route2 ex:hasTrip ?trip2 .
  ?trip2 ex:arrivesAt ?arrival4 .
  ?arrival4 ex:arrivalPlace ?stop3 .
  ?stop3 ex:stopName ?stopName3 .
  ?stop3 ex:stopName "ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ" .

  ?trip1 ex:arrivesAt ?arrival2 .
  ?arrival2 ex:arrivalPlace ?stop2 .
  ?stop2 ex:stopName ?stopName2 .
  ?trip2 ex:arrivesAt ?arrival3 .
  ?arrival3 ex:arrivalPlace ?stop2 .

  FILTER (?bus1 != ?bus2) .
}
```

Query				
Default Graph IRI				
Query				
PREFIX ex: <http://example.com/transportation#> SELECT DISTINCT (?stopName1 AS ?start) ?bus1 (?stopName2 AS ?change) ?bus2 (?stopName3 AS ?destination) WHERE { ?bus1 ex:hasRoute ?route1 . ?route1 ex:hasTrip ?trip1 . ?trip1 ex:arrivesAt ?arrival1 . ?arrival1 ex:arrivalPlace ?stop1 . ?stop1 ex:stopName ?stopName1 . ?stop1 ex:stopName "ΒΟΡΕΙΟΣ ΠΟΛΟΣ" . ?bus2 ex:hasRoute ?route2 . ?route2 ex:hasTrip ?trip2 . ?trip2 ex:arrivesAt ?arrival4 . ?arrival4 ex:arrivalPlace ?stop3 .				
Execute Save Load Clear				
start	bus1	change	bus2	destination
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"1η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"9η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"8η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"7η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"6η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"ΤΣΙΤΟΥΡΑ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"ΚΟΤΟΠΟΥΛΗ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"
"ΒΟΡΕΙΟΣ ΠΟΛΟΣ"	http://example.com/transportation#608	"2η ΖΩΓΡΑΦΟΥ"	http://example.com/transportation#230	"ΛΑΙΚΟ ΝΟΣΟΚΟΜΕΙΟ"

Query 3: Αλλαγή 2 μέσων

Μία άλλη δυνατότητα της βάσης που αξίζει να προβληθεί είναι η γεωγραφική αναπαράσταση της τοποθεσίας των στάσεων. Όπως αναφέρθηκε προηγουμένως, χρησιμοποιήθηκε το πρότυπο [GeoSPARQL](#) και το λεξιλόγιο [WGS84 Geo Positioning](#) προκειμένου να αντιστοιχίσουμε τις γεωγραφικές συντεταγμένες των στάσεων σε γεωγραφικά σημεία. Έτσι δημιουργήσαμε ένα ερώτημα στο οποίο ο χρήστης αναζητεί τις στάσεις που απέχουν από την θέση του μέχρι και 500 μέτρα. Στο αποτέλεσμα παρουσιάζουμε κατά αύξουσα σειρά απόστασης το όνομα της στάσης, την περιγραφή της καθώς και την απόσταση από τον χρήστη.

PREFIX ex: <http://example.com/transportation#>

```
SELECT DISTINCT ?name ?desc (bif:st_distance (?geo, bif:st_point (37.966311, 23.729053)) as ?distance) where {
  ?stop ex:stopLocation ?m .
  ?stop ex:stopName ?name .
  ?stop ex:stopDescription ?desc .
  ?m geo:geometry ?geo .
```

```
  FILTER (bif:st_intersects (?geo, bif:st_point (37.966311, 23.729053), 0.5))
} ORDER BY bif:st_distance (?geo, bif:st_point (37.966311, 23.729053))
```

Query Saved Queries		
Default Graph IRI <input type="text" value="http://example.com/transportation"/>		
<p>Query</p> <pre>PREFIX ex: <http://example.com/transportation#> SELECT DISTINCT ?name ?desc (bif:st_distance (?geo, bif:st_point (37.966311, 23.729053)) as ?distance) WHERE { ?stop ex:stopLocation ?m . ?stop ex:stopName ?name . ?stop ex:stopDescription ?desc . ?m geo:geometry ?geo . FILTER (bif:st_intersects (?geo, bif:st_point (37.966311, 23.729053), 0.5)) } ORDER BY bif:st_distance (?geo, bif:st_point (37.966311, 23.729053))</pre>		
Execute Save Load Clear		
name	desc	distance
"ΧΑΤΖΗΧΡΗΣΤΟΥ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.118424
"ΑΓ.ΙΩΑΝΝΗΣ"	"Επί της ΒΕΙΚΟΥ"	0.146643
"ΑΓ.ΙΩΑΝΝΗΣ"	"Επί της ΔΗΜΗΤΡΑΚΟΠΟΥΛΟΥ"	0.183584
"ΑΓ.ΠΑΝΤΕΛΕΗΜΩΝ"	"Επί της ΚΑΛΛΙΠΡΟΗΣ"	0.203621
"ΜΟΥΣΕΙΟ ΑΚΡΟΠΟΛΗΣ"	"Επί της ΧΑΤΖΗΧΡΗΣΤΟΥ"	0.22173
"ΦΙΞ"	"Επί της ΚΑΛΛΙΠΡΟΗΣ"	0.25359
"ΣΤ.ΑΚΡΟΠΟΛΗ"	"Επί της ΣΤΡΓΟΥ ΜΑΚΡΥΓΙΑΝΝΗ"	0.254981
"ΦΙΞ"	"Επί της ΚΑΛΛΙΠΡΟΗΣ"	0.262501
"ΜΟΥΣΕΙΟ ΑΚΡΟΠΟΛΗΣ"	"Επί της ΜΙΣΑΡΑΛΙΝΤΟΥ"	0.282987
"ΣΤ.ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.295609
"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.303498
"ΛΕΩΦ.ΒΟΥΛΙΑΓΜΕΝΗΣ"	"Επί της ΚΑΛΛΙΠΡΟΗΣ"	0.306468
"ΜΕΤΡΟ ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.307086
"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.313851

Query 4: Αναζήτηση κοντινών στάσεων (1)

"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.303498
"ΛΕΩΦ.ΒΟΥΛΙΑΓΜΕΝΗΣ"	"Επί της ΚΑΛΛΙΡΡΟΗΣ"	0.306468
"ΜΕΤΡΟ ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.307086
"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.313851
"ΣΤ.ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.315715
"ΣΤ.ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.32137
"ΜΕΤΡΟ ΣΥΓΓΡΟΥ-ΦΙΞ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.325049
"ΑΦΕΤΗΡΙΑ"	"Επί της ΔΙΑΜ.ΧΩΡΟΥ ΣΤ.ΣΥΓΓΡΟΥ ΦΙΞ"	0.385071
"ΣΥΓΓΡΟΥ"	"Επί της ΑΘ.ΔΙΑΚΟΥ"	0.399538
"ΑΓΙΑ ΦΩΤΕΙΝΗ"	"Επί της ΚΑΛΛΙΡΡΟΗΣ"	0.408832
"ΑΓ.ΦΩΤΕΙΝΗΣ"	"Επί της ΚΑΛΛΙΡΡΟΗΣ"	0.425074
"ΑΚΡΟΠΟΛΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.425369
"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΑΘ.ΔΙΑΚΟΥ"	0.433764
"ΑΚΡΟΠΟΛΗ"	"Επί της ΛΕΩΦ.ΣΥΓΓΡΟΥ"	0.448025
"ΓΑΡΓΑΡΕΤΤΑ"	"Επί της ΒΕΙΚΟΥ"	0.449937
"ΓΑΡΓΑΡΕΤΤΑ"	"Επί της ΔΗΜΗΤΡΑΚΟΠΟΥΛΟΥ"	0.451633
"ΜΑΚΡΥΓΙΑΝΝΗ"	"Επί της ΛΕΩΦ.ΒΟΥΛΙΑΓΜΕΝΗΣ"	0.455454
"ΕΡΕΧΘΕΙΟΥ"	"Επί της ΡΟΒΕΡΤΟΥ ΓΚΑΛΛΙ"	0.461952
"ΕΡΕΧΘΕΙΟΥ"	"Επί της ΚΑΒΑΛΛΟΤΙ"	0.484654

Query 4: Αναζήτηση κοντινών στάσεων (2)

Τέλος, επειδή η βάση γνώσης δεν περιορίζεται μόνο στα αστικά μέσα μεταφοράς της Αθήνας, αλλά επεκτείνεται και σε όλα τα μέσα μεταφοράς, κρίθηκε αναγκαία η περίληψη ενδιαφερόντων ερωτημάτων που θα αξιοποιούν τη μεγάλη αυτή ποικιλία γνώσης. Επειδή όμως δεν υπήρχαν διαθέσιμα δεδομένα για όλες τις μεταφορές, περιοριστήκαμε απλά στη σύνταξη των ερωτημάτων. Αρχικά, ένα σχετικά απλό ερώτημα είναι η εμφάνιση των αεροδρομίων που φτάνουν απευθείας πτήσεις με αφετηρία το αεροδρόμιο του Ηρακλείου.

Το τελευταίο ερώτημα επιστρέφει τα μέσα τα οποία συνδέουν απευθείας ένα αεροδρόμιο με ένα λιμάνι, καθώς και το εν λόγω αεροδρόμιο και λιμάνι. Αυτό το επιτυγχάνουμε αναζητώντας μέσα που κάνουν στάση σε απόσταση μικρότερη του ενός χιλιομέτρου από ένα λιμάνι και από ένα αεροδρόμιο.

PREFIX ex: <http://example.com/transportation#>

```
SELECT DISTINCT ?airportName WHERE {
  ?plane a <http://example.com/transportation#Airplane> .
  ?plane ex:hasRoute ?route .
  ?route ex:hasTrip ?trip .
  ?trip ex:arrivesAt ?arrival .
  ?arrival ex:arrivalPlace "KAHK" .
  ?arrival ex:arrivalPlace ?airport .
  ?airport ex:stopName ?airportName .

  FILTER (?airportName != "KAHK") .
}
```

PREFIX ex: <http://example.com/transportation#>

```
SELECT DISTINCT ?mean ?airportName ?portName WHERE {
  ?mean ex:hasRoute ?route .
  ?route ex:hasTrip ?trip .
  ?trip ex:arrivesAt ?arrival1 .
  ?arrival1 ex:arrivalPlace ?stop1 .
  ?stop1 ex:stopLocation ?loc1 .
  ?loc1 geo:geometry ?geo1 .
  ?trip ex:arrivesAt ?arrival2 .
  ?arrival2 ex:arrivalPlace ?stop2 .
  ?stop2 ex:stopLocation ?loc2 .
  ?loc2 geo:geometry ?geo2 .

  ?airport a <http://example.com/transportation#Airport> .
  ?airport ex:stopLocation ?airLoc .
  ?airport ex:stopName ?airportName .
  ?airLoc geo:geometry ?airGeo .
  ?port a <http://example.com/transportation#Port> .
  ?port ex:stopLocation ?portLoc .
  ?port ex:stopName ?portName .
  ?portLoc geo:geometry ?portGeo .

  FILTER (bif:st_intersects (?geo1, ?airGeo, 1) &&
    bif:st_intersects (?geo2, ?portGeo, 1)) .
}
```