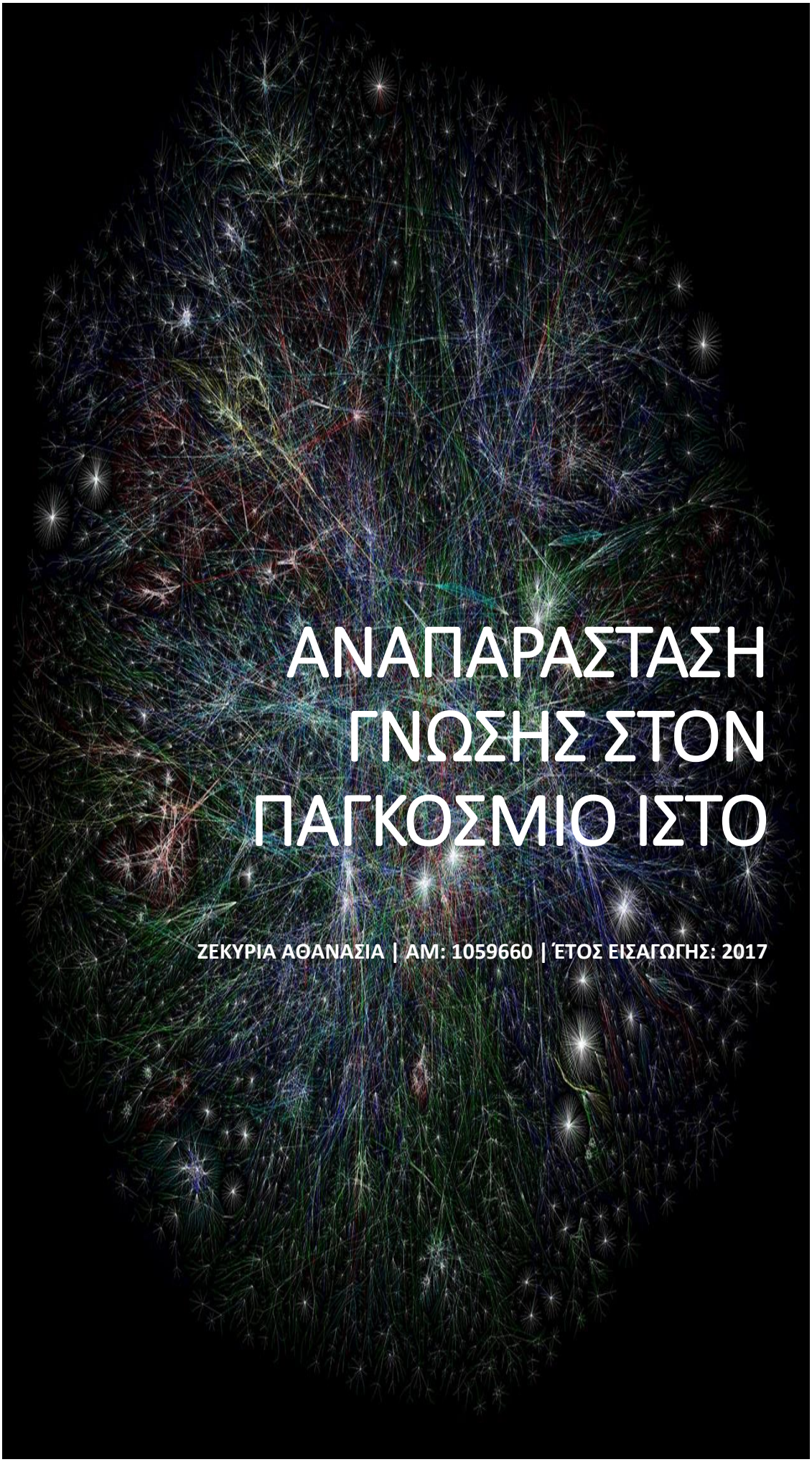


ΑΝΑΠΑΡΑΣΤΑΣΗ ΓΝΩΣΗΣ ΣΤΟΝ ΠΑΓΚΟΣΜΙΟ ΙΣΤΟ

ΖΕΚΥΡΙΑ ΑΘΑΝΑΣΙΑ | ΑΜ: 1059660 | ΈΤΟΣ ΕΙΣΑΓΩΓΗΣ: 2017

Άσκηση2 GitHub

Περιεχόμενα

Ερώτημα 1:

“Η Ιλιάδα συνθέθηκε από ποιητή που έζησε τον 8ο αιώνα π.Χ. στην Ιωνία της Μικράς Ασίας.”

1.a.

Ο RDF κώδικας:

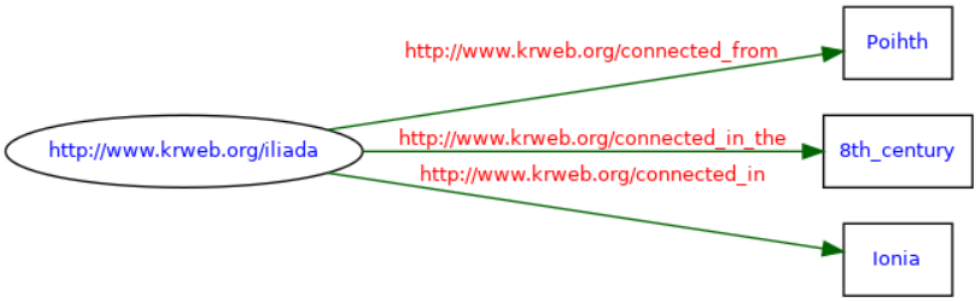
```
<?xml version="1.0"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:krweb="http://www.krweb.org/">

    <rdf:Description rdf:about="http://www.krweb.org/iliada">
        <krweb:connected_from>Poihth</krweb:connected_from>
        <krweb:connected_in_the>8th_century</krweb:connected_in_the>
        <krweb:connected_in>Ionia</krweb:connected_in>
    </rdf:Description>

</rdf:RDF>
```

Τα αποτελέσματα του validator (στα display result options επιλέγουμε Triples and Graph, όπως ζητείται στην εκφώνηση):

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.krweb.org/iliada	http://www.krweb.org/connected_from	"Poihth"
2	http://www.krweb.org/iliada	http://www.krweb.org/connected_in_the	"8th_century"
3	http://www.krweb.org/iliada	http://www.krweb.org/connected_in	"Ionia"

Graph of the data model



1.b.

Ο RDF κώδικας:

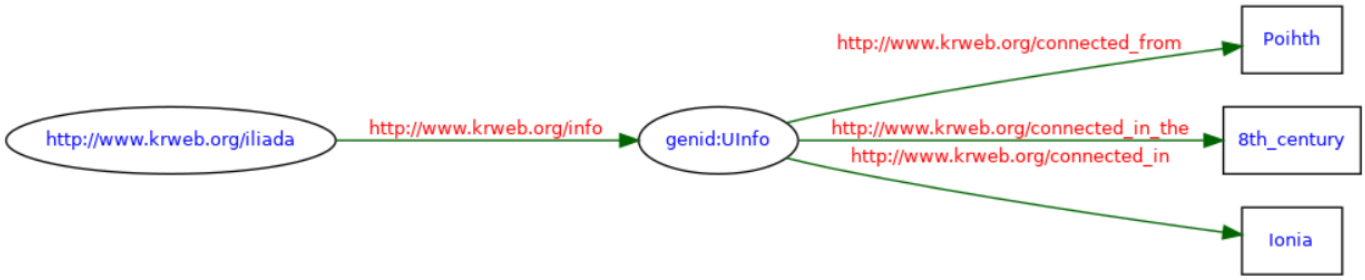
```
<?xml version="1.0"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:krweb="http://www.krweb.org/">

    <rdf:Description rdf:about="http://www.krweb.org/iliada">
        <krweb:info rdf:nodeID="Info"/>
    </rdf:Description>
    <rdf:Description rdf:nodeID="Info">
        <krweb:connected_from>Poihth</krweb:connected_from>
        <krweb:connected_in_the>8th_century</krweb:connected_in_the>
        <krweb:connected_in>Ionia</krweb:connected_in>
    </rdf:Description>

</rdf:RDF>
```

Τα αποτελέσματα του validator:

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.krweb.org/iliada	http://www.krweb.org/info	genid:UInfo
2	genid:UInfo	http://www.krweb.org/connected_from	"Poihth"
3	genid:UInfo	http://www.krweb.org/connected_in_the	"8th_century"
4	genid:UInfo	http://www.krweb.org/connected_in	"Ionia"

Graph of the data model



Ερώτημα 2:

“Η google αναφέρει ότι το Τμήμα Η/Υ & Πληροφορικής βρίσκεται στο Ρίο.”

Ο RDF κώδικας:

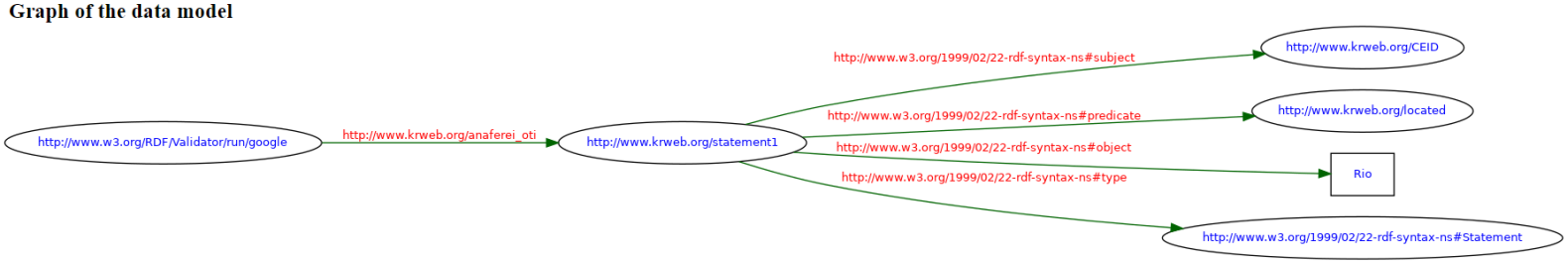
```
<?xml version="1.0"?>

<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:krweb="http://www.krweb.org/">

    <rdf:Description rdf:about="http://www.krweb.org/statement1">
        <rdf:subject rdf:resource="http://www.krweb.org/CEID" />
        <rdf:predicate rdf:resource="http://www.krweb.org/located" />
        <rdf:object>Rio</rdf:object>
        <rdf:type rdf:resource="http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement"/>
    </rdf:Description>

    <rdf:Description rdf:about="google">
        <krweb:anaferei_oti rdf:resource="http://www.krweb.org/statement1"/>
    </rdf:Description>
</rdf:RDF>
```

Τα αποτελέσματα του validator:

Triples of the Data Model

Number	Subject	Predicate	Object
1	http://www.krweb.org/statement1	http://www.w3.org/1999/02/22-rdf-syntax-ns#subject	http://www.krweb.org/CEID
2	http://www.krweb.org/statement1	http://www.w3.org/1999/02/22-rdf-syntax-ns#predicate	http://www.krweb.org/located
3	http://www.krweb.org/statement1	http://www.w3.org/1999/02/22-rdf-syntax-ns#object	"Rio"
4	http://www.krweb.org/statement1	http://www.w3.org/1999/02/22-rdf-syntax-ns#type	http://www.w3.org/1999/02/22-rdf-syntax-ns#Statement
5	http://www.w3.org/RDF/Validator/run/google	http://www.krweb.org/anaferei_oti	http://www.krweb.org/statement1

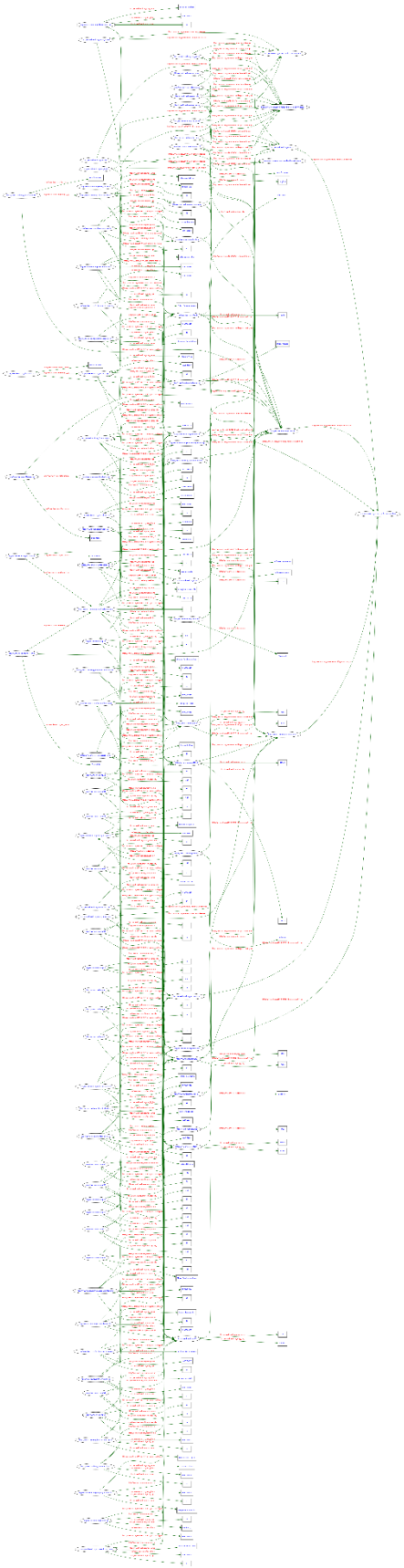
Graph of the data model



Ερώτημα 3:

Ο RDF κώδικας υπάρχει στο τελικό αρχείο rar με όνομα “3.rdf”.

Τα αποτελέσματα του validator:



Για την πλήρη εικόνα, πατήστε:

Εδώ

Βάση του xml: xml:base="http://www.krweb.org/">
Αυτό το κάνουμε για να κάνουμε απευθείας χρήση των ιδιοτήτων του rdfs χωρίς να χρειαζόμαστε περιγραφή rdf.

Οι 6 κλάσεις δηλώνονται <rdfs:Class rdf:ID= "όνομα της κλάσης"> και
<rdfs:subClassOf rdf:resource= #όνομα της υπερκλάσης"/> για την περαιτέρω δήλωση υποκλάσης.

Οι ιδιότητες των κλάσεων δηλώνονται
<rdf:Property rdf:about="http://www.krweb.org/όνομα ιδιότητας">.

Το <rdfs:domain rdf:resource="http://www.krweb.org/# όνομα κλάσης"> δηλώνει σε ποια κλάση ανήκει η ιδιότητα.

Το <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Literal"/> δηλώνει ότι πρόκειται για αλφαριθμητικό.

Το <rdfs:range rdf:resource="http://www.w3.org/2000/01/rdf-schema#Integer"/> δηλώνει ότι πρόκειται για ακέραιο.

Το <rdfs:range rdf:resource="http://www.krweb.org/#όνομα κλάσης"/> δηλώνει την συσχέτιση της ιδιότητας με κάποια άλλη κλάση. Για παράδειγμα, η ιδιότητα taught_by ανήκει στην κλάση Lesson και συσχετίζεται με την κλάση Professor.

Για τη δημιουργία των τμημάτων, καθηγητών, φοιτητών, μαθημάτων και αιθουσών:
<rdf:Description rdf:about="http://www.krweb.org/όνομα">, για το όνομα τους.
<rdf:type rdf:resource="http://www.krweb.org/#όνομα κλάσης"/> για την κλάση στην οποία ανήκουν.
<krweb:όνομα ιδιότητας>αλφαριθμητικό ή ακέραιος</krweb:όνομα ιδιότητας> αναλόγως την ιδιότητα και το πως έχει δηλωθεί το range της ιδιότητας.
<krweb:όνομα ιδιότητας rdf:resource="http://www.krweb.org/συσχετιζόμενο όνομα"/>, αυτό το «συσχετιζόμενο» όνομα, π.χ. μπορεί να είναι το όνομα του τμήματος που διδάσκει ο εκάστοτε καθηγητής για την ιδιότητα member_of.

Ερώτημα 4:

Ακολούθησα αναλυτικά τις οδηγίες εγκατάστασης των jdk-11 και jena. Ανέβασα το rdf της προηγούμενης άσκησης και στην συνέχεια, κατασκεύασα το ζητούμενο SPARQL Query.

4.i.

Μας ζητείται να επιστρέφονται τα τηλέφωνα: SELECT ?phone
όλων των καθηγητών: ?professor rdf:type <http://www.krweb.org/#Professor>
μέσω της ιδιότητας has_phone που έχει η κλάση καθηγητής: ?professor krweb:has_phone ?phone.

Το ζητούμενο 4i.rq αρχείο για την εμφάνιση των τηλεφώνων των καθηγητών:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?phone
WHERE
{
    ?professor rdf:type  <http://www.krweb.org/#Professor>.
    ?professor krweb:has_phone ?phone.
}
```

Ο ζητούμενος πίνακας με τα τηλέφωνα των καθηγητών:

	phone
1	6901234567
2	691234567
3	692345678
4	693456789
5	694567891
6	695678912
7	696789123
8	697891234
9	698912345
10	699123456

4.ii.

Με τον ίδιο τρόπο όπως και στο 4.i. επιλέγουμε τα τηλέφωνα των φοιτητών που έχουν την ιδιότητα has_phone και στην συνέχεια, όταν ο φοιτητής έχει την ιδιότητα has_age: (?student krweb:has_age ?age.),
κάνουμε χρήση της: FILTER (?age>"23") για να ορίσουμε περιορισμό η ηλικία να είναι άνω των 23.

Το ζητούμενο 4ii.rq αρχείο για την εμφάνιση των τηλεφώνων των μαθητών με ηλικία μεγαλύτερη των 23:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?phone
WHERE
{
    ?student rdf:type  <http://www.krweb.org/#Student>.
    ?student krweb:has_phone ?phone.
    ?student krweb:has_age ?age.
    FILTER (?age>"23")
}
```

Ο ζητούμενος πίνακας με τα τηλέφωνα των μαθητών με ηλικία μεγαλύτερη των 23:

	phone
1	6900000003
2	6900000005
3	6900000007
4	6900000012
5	6900000013
6	6900000017

4.iii.

Αυτήν την φορά πρέπει να εμφανίσουμε τα ονόματα των ατόμων: SELECT ?name
Το άτομο πρέπει να είναι μέλος: ?person krweb:member_of ?dep. ,
τμήματος που βρίσκεται στην Πάτρα: ?dep krweb:dep_city "Patra".
και προφανώς, φέρει την ιδιότητα: ?person krweb:has_name ?name.

Το ζητούμενο 4iii.rq αρχείο για την εμφάνιση των ονοματεπωνύμων των ατόμων που είναι μέλη σε τμήμα που βρίσκεται στην Πάτρα:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?name
WHERE
{
    ?person rdf:type  <http://www.krweb.org/#Person>.
    ?person krweb:member_of ?dep.
    ?dep krweb:dep_city "Patra".
    ?person krweb:has_name ?name.
}
```

Τα αποτελέσματα δεν είναι τα αναμενόμενα:

name
No data available in table

Ωστόσο, αλλάζοντας τις παραμέτρους σε professor και student, οι ζητούμενοι πίνακες με τα ονόματα των ατόμων που διδάσκουν ή φοιτούν είναι σωστοί:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?name
WHERE
{
    ?professor rdf:type  <http://www.krweb.org/#Professor>.
    ?professor krweb:member_of ?dep.
    ?dep krweb:dep_city "Patra".
    ?professor krweb:has_name ?name.
}
```

	name
1	Athanasia Zekyria
2	Besiana Agko
3	Aristea Korba
4	Ellada Nikolaou
5	Fotini Mortzou
6	Maria Drebela

Showing 1 to 6 of 6 entries

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?name
WHERE
{
    ?student rdf:type  <http://www.krweb.org/#Student>.
    ?student krweb:member_of ?dep.
    ?dep krweb:dep_city "Patra".
    ?student krweb:has_name ?name.
}
```

	name
1	Nikolaos Loukisas
2	Konstantinos Daglas
3	Dimitrios Altanis
4	Giorgos Giannopoulos
5	Klodi Filaj
6	Konstantinos Akrivos
7	Mixail Vardalis
8	Orfeas Nonis
9	Giorgos Papakonstantinou
10	Panagiotis Mpogas
11	Panagiotis Pipis

Showing 1 to 11 of 11 entries

Συμπεραίνουμε με αυτόν τον τρόπο ότι η SPARQL λειτουργεί για βεβαιωμένες δηλώσεις και όχι για συμπερασματικές, δηλαδή **δεν χρησιμοποιεί κάποιο Μηχανισμό Συμπερασμού [Inference Engine].**

“Δημιουργήστε τουλάχιστον 10 καθηγητές και 20 φοιτητές συνολικά και ορίστε τιμές για τις ιδιότητες τους. Στην δήλωση τους θα αναφέρετε την κλάση που ανήκουν [καθηγητής ή φοιτητής] ωστόσο ΔΕΝ πρέπει να αναφέρετε ότι ανήκουν στην κλάση Person.” Στην εκφώνηση μας ζητείται ρητά στο ερώτημα 3 κατά την συγγραφή της RDFS οντολογίας, να μην αναφέρουμε την υπερκλάση person, ακριβώς για να καταλήξουμε στο παραπάνω συμπέρασμα.

4.iv.

Επιλέγουμε τα ονόματα: SELECT ?room_name
των αιθουσών: ?classroom rdf:type <http://www.krweb.org/#Classroom>.
που βρίσκονται σε τμήμα: ?classroom krweb:room_department ?dep.
που βρίσκεται στην Πάτρα: ?dep krweb:dep_city "Patra".
και φέρει τις ιδιότητες room_name: ?classroom krweb:room_name ?room_name.
και room_capacity, αναφερόμενοι σε αυτό cap: ?classroom krweb:room_capacity ?cap.
Με την χρήση της: FILTER (?cap>"150"), περιορίζουμε την εμφάνιση των ονομάτων σε αίθουσες με χωρητικότητα άνω του 150.

Το ζητούμενο 4iv.rq αρχείο για την εμφάνιση των ονομάτων των αιθουσών που έχουν μεγαλύτερη χωρητικότητα από 150 και βρίσκονται στην Πάτρα:

```
PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX rdfs:<http://www.w3.org/2000/01/rdf-schema#>
PREFIX krweb:<http://www.krweb.org/>
SELECT ?room_name
WHERE
{
    ?classroom rdf:type  <http://www.krweb.org/#Classroom>.
    ?classroom krweb:room_department ?dep.
    ?dep krweb:dep_city "Patra".
    ?classroom krweb:room_name ?room_name.
    ?classroom krweb:room_capacity ?cap.
    FILTER (?cap>"150")
}

```

Ο ζητούμενος πίνακας με τα ονόματα των αιθουσών που έχουν μεγαλύτερη χωρητικότητα από 150 και βρίσκονται στην Πάτρα:

	room_name
1	A
2	A1
3	A2

Showing 1 to 3 of 3 entries

*Τα ονόματα των αιθουσών για κάθε τμήμα είναι: Ai, Bi, Ci, (με 1≤i≤5), με διαφορά το τμήμα CEID που έχει αίθουσες A, B, C.

Ερώτημα 5:

Στον φάκελο ‘’5’’ του .rar υπάρχει ο σχετικός κώδικας.

Για την κατασκευή του προγράμματος, χρειάστηκαν 6 κλάσεις:

Main.java: κάνει print το menu:

```
private static void printMenu()
    {
        System.out.println("*************************************************************
***");
        System.out.println("* Please select one of the following:");
        System.out.println("* 1. Show Staff by Department");
        System.out.println("* 2. Show Students by Department");
        System.out.println("* 3. Show Classrooms by Department");
        System.out.println("* 4. Show Lessons by Department");
        System.out.println("* 4. Add Data");
        System.out.println("* 6. Show Statements by URI");
        System.out.println("* 7. Exit");
        System.out.println("*************************************************************
***");
        System.out.print("Choice: ");
    }
```

τα departments, ρωτάει τον χρήστη να επιλέξει department και στην συνέχεια τυπώνει τους professors, students, classrooms και lessons του επιλεγμένου department:

```
private static void showStaffByDepartment()
    {
        // Creating List of Departments
        ArrayList<String> departments = RDFHandler.findDepartments();

        // Printing Available Departments
        System.out.println("Departments: ");
        for (String department : departments)
        {
            System.out.println((departments.indexOf(department)+1) + ". " + department);
        }

        // Ask user to select Department
        System.out.print("Select Department: ");
        int departmentIndex = scanner.nextInt();
        departmentIndex--;
        String department = departments.get(departmentIndex);
        System.out.println("You Selected: " + department);

        // Creating lists of staff, students, classrooms, lessons of selected Department
        ArrayList<Professor> professors = RDFHandler.findStaffByDepartment(department);
        ArrayList<Student> students = RDFHandler.findStudentsByDepartment(department);
```

```
        ArrayList<Classroom> classrooms =  RDFHandler.findClassroomsbyDepartment(department);
        ArrayList<Lesson> lessons = RDFHandler.findLessonsByDepartment(department);

        // Printing Staff of selected Department
        System.out.println(String.format("%30s %30s %30s","Name", "Age", "Phone"));
        for (Professor professor:professors)
        {
            professor.printProfessor();
        }

        // Printing Students of selected Department
        System.out.println(String.format("%30s %30s %30s","Name", "Age", "Phone"));
        for (Student student:student)
        {
            student.printStudent();
        }

        // Printing Classrooms of selected Department
        System.out.println(String.format("%30s %30s","Room_Name", "Capacity"));
        for (Classroom classroom:classroom)
        {
            classroom.printClassroom();
        }

        // Printing Lessons of selected Department
        System.out.println(String.format("%30s %30s","Lesson_Name", "Teacher"));
        for (Lesson lesson:lesson)
        {
            lesson.printLesson();
        }
    }
```

Δείχνει το data menu και ζητάει από τον χρήστη την επιλογή του:

```
private static void printAddDataMenu()
    {
        System.out.println("*****************************************************************
");
        System.out.println("* Please select one of the following:");
        System.out.println("* 1. Add Professor");
        System.out.println("* 2. Add Student");
        System.out.println("* 3. Add Department");
        System.out.println("* 4. Add Lesson");
        System.out.println("* 5. Add Classroom");
        System.out.println("* 6. Cancel");
        System.out.println("*****************************************************************
");
        System.out.print("Choice: ");
    }
```

Ζητάει τα δεδομένα του καθηγητή, φοιτητή, τμήματος, μαθήματος ή τάξης για προσθήκη:

```
private static void addProfessor()
    {
        System.out.print("Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("Phone: ");
        String phone = scanner.next();
        System.out.print("Age: ");
        String age = scanner.next();
        System.out.print("Department: ");
        String department = scanner.next();
        System.out.print("Lesson: ");
        String lesson = scanner.next();

        RDFHandler.addProfesor(name, phone, age, department, lesson);
    }


    private static void addStudent()
    {
        System.out.print("Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("Phone: ");
        String phone = scanner.next();
        System.out.print("Age: ");
        String age = scanner.next();
        System.out.print("Department: ");
        String department = scanner.next();

        RDFHandler.addStudent(name, phone, age, department);
    }


    private static void addDepartment()
    {
        System.out.print("Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("City: ");
        String city = scanner.next();

        RDFHandler.addDepartment(name, city);
    }


    private static void addLesson()
```

```
    {
        System.out.print("Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("Teacher: ");
        scanner.nextLine();
        String teacher = scanner.nextLine();

        RDFHandler.addLesson(name, teacher);
    }


    private static void addClassroom()
    {
        System.out.print("Name: ");
        scanner.nextLine();
        String name = scanner.nextLine();
        System.out.print("Capacity: ");
        scanner.nextLine();
        String teacher = scanner.nextLine();

        RDFHandler.addClassroom(name, capacity);
    }
```

Και τέλος, για το ερώτημα Γ ζητάει το URI και τυπώνει τα statements του δεδομένου URI:

```
private static void showDataByURI()
    {
        System.out.print("URI: ");
        String uri = scanner.next();

        String uriInfo = RDFHandler.selectDataByUri(uri);

        System.out.println(uriInfo);
    }
```

Professor.java: κάνει store τα δεδομένα του καθηγητή

```java
public class Professor
{
    private String name;
    private String age;
    private String phone;

    public Professor(String name, String age, String phone)
    {
        this.name = name;
        this.age = age;
        this.phone = phone;
    }

    void printProfessor()
    {
        System.out.println(String.format("%30s %30s %30s",name, age, phone));
    }
}
```

Student.java: κάνει store τα δεδομένα του φοιτητή

```java
public class Student
{
    private String name;
    private String age;
    private String phone;

    public Student(String name, String age, String phone)
    {
        this.name = name;
        this.age = age;
        this.phone = phone;
    }

    void printStudent()
    {
        System.out.println(String.format("%30s %30s %30s",name, age, phone));
    }
}
```

Lesson.java: κάνει store τα δεδομένα του μαθήματος

```java
public class Lesson
{
    private String les_name;
    private String taught_by;

    public Lesson(String les_name, String taught_by)
    {
        this.les_name = les_name;
        this.taught_by = taught_by;
    }

    void printLesson()
    {
        System.out.println(String.format("%30s %30s","Lesson_Name", "Teacher"));
    }
}
```

Classroom.java: κάνει store τα δεδομένα της αίθουσας

```java
public class Classroom
{
    private String room_name;
    private String room_capacity;

    public Classroom(String room_name, String room_apacity)
    {
        this.room_name = room_name;
        this.room_capacity = room_capacity;
    }

    void printClassroom()
    {
        System.out.println(String.format("%30s %30s","Room_Name", "Capacity"));
    }
}
```

RDFHandler.java: κάνει store το main rdf model:

```
private static Model model;
```

κάνει load το rdf file:

```
static void loadRDF(String fileName)
    {
        // create an empty model
        model = ModelFactory.createDefaultModel();

        // use the FileManager to find the input file
        file = fileName;
        InputStream file = FileManager.get().open(fileName);
        if (file == null) {
            throw new IllegalArgumentException( "File: " + fileName + " not found");
        }

        // read the RDF/XML file
        model = model.read( file, null);
    }
```

αναζητάει departments στο rdf file:

```
private static SimpleSelector selectDepartments()
    {
        // Selects all the resources with a dep_name property
        Model m = ModelFactory.createDefaultModel();
        Property depName = m.createProperty(krweb,"dep_name" );

        return new SimpleSelector(null, depName,(RDFNode) null);
    }
```

και επιστρέφει την λίστα τους:

```
 static ArrayList<String> findDepartments()
    {
        SimpleSelector departmentsSelector = selectDepartments();

        // Adding departments to list
        ArrayList<String> departments = new ArrayList<String>();
        StmtIterator iter = model.listStatements(departmentsSelector);
        while (iter.hasNext())
        {
            departments.add(iter.nextStatement().getString());
        }

        return departments;
    }
```

Και το hashmap department-URI:

```
 private static HashMap<String,String> findDepartmentURI()
    {
        SimpleSelector departmentsSelector = selectDepartments();

        // Adding Departments and URIs to hashmap
        HashMap<String,String> departments = new HashMap<String,String>();
        StmtIterator iter = model.listStatements(departmentsSelector);
        while (iter.hasNext())
        {
            Statement department = iter.nextStatement();
            departments.put(department.getString(),department.getSubject().toString());
        }

        return departments;
    }
```

Μετά την δημιουργία και επιλογή nodes του δεδομένου department κάνει iterate στο κάθε node, παρακάτω βλέπουμε τους καθηγητές:

```
ArrayList<Professor> professors = new ArrayList<Professor>();
        while (iter.hasNext())
        {
            Statement stmt = iter.nextStatement();

            // Assigning and adding values into professor object and adding professor to the
list
            String name = stmt.getSubject().getProperty(hasName).getString();
            String phone = stmt.getSubject().getProperty(hasPhone).getString();
            String age = stmt.getSubject().getProperty(hasAge).getString();

            professors.add(new Professor(name,age,phone));
        }

        return professors;
```

εκτελεί το SPARQL query και τυπώνει τα αποτελέσματα:

```
private static void executeSPARQL(String queryString)
    {
        Query query = QueryFactory.create(queryString);

        QueryExecution qe = QueryExecutionFactory.create(query, model);
        ResultSet results = qe.execSelect();

        ResultSetFormatter.out(System.out, results, query);
        qe.close();
    }
```

Για την προσθήκη οντολογιών στο rdf file ακολουθεί παράδειγμα του professor:

```
static void addProfesor(String name, String phone, String age, String department, String
lesson)
    {
        String newNodeString = "\t<!-- Added with rdf tool (Professor)-->\n" +
                "\t<rdf:Description rdf:about=\"http://www.krweb.org/" +
name.toLowerCase().replace(" ", "") + "\">\n" +
                "\t\t<rdf:type rdf:resource=\"http://www.krweb.org/#Professor\"/>\n" +
                "\t\t<krweb:has_name>" + name + "</krweb:has_name>\n" +
                "\t\t<krweb:has_phone>" + phone + "</krweb:has_phone>\n" +
                "\t\t<krweb:has_age>" + age + "</krweb:has_age>\n" +
                "\t\t<krweb:member_of
rdf:resource=\"http://www.krweb.org/"+department.toLowerCase().replace(" ","")+ "\"/>\n" +
                "\t\t<krweb:teaches rdf:resource=\"http://www.krweb.org/" + lesson + "\"
/>\n" +
                "\t</rdf:Description>";

        addNodeToFile(newNodeString);
    }
```

Προσθήκη ενός node στο root element:

```
private static void addNodeToFile(String node)
    {
        File inputFile = new File(file);
        DocumentBuilderFactory factory=DocumentBuilderFactory.newInstance();
        Document doc = null;
        DocumentBuilder builder;
        try
        {
            builder = factory.newDocumentBuilder();
            // creating input stream
            doc = builder.parse(inputFile );
            Node newNode = doc.createTextNode(node);
            doc.getDocumentElement().appendChild(newNode);
            TransformerFactory transformerFactory = TransformerFactory.newInstance();
            Transformer transformer = transformerFactory.newTransformer();
            transformer.setOutputProperty(OutputKeys.INDENT, "yes");
            transformer.setOutputProperty(OutputKeys.METHOD, "xml");
            DOMSource source = new DOMSource(doc);
            StreamResult streamResult =  new StreamResult(new File(file));
            transformer.transform(source, streamResult);
            fixXmlFile(file);
        }
        catch (Exception e)
        {
            e.printStackTrace();
        }
    }
```

Επιδιορθώνει escape χαρακτήρες στο δεδομένο xml αρχείο:

```
private static void fixXmlFile(String file) throws IOException
    {

        String xmlFile = readFile(file);
        xmlFile = xmlFile.replace("&lt;","<").replace("&gt;",">").replace("</rdf:RDF>",
"\r</rdf:RDF>");

        PrintWriter writer = new PrintWriter(file, "UTF-8");
        writer.print(xmlFile);
        writer.close();
    }
```

Διαβάζει το αρχείο:

```
private static String readFile(String fileName) throws IOException
    {
        BufferedReader br = new BufferedReader(new FileReader(fileName));
        try {
            StringBuilder sb = new StringBuilder();
            String line = br.readLine();

            while (line != null) {
                sb.append(line);
                sb.append("\n");
                line = br.readLine();
            }
            return sb.toString();
        } finally {
            br.close();
        }
    }
```

Και επιστρέφει τα statements του δεδομένου URI:

```
 static String selectDataByUri(String uri)
    {
        String uriInfo = "";
        // Selects all the resources with a dep_name property
        Model m = ModelFactory.createDefaultModel();
        //Property depName = m.createProperty(krweb,"dep_name" );
        Resource resource = m.createResource(uri);

        SimpleSelector selector = new SimpleSelector(resource, null,(RDFNode) null);

        StmtIterator iter = model.listStatements(selector);

        // Iterating through each node
        while (iter.hasNext())
        {
            Statement stmt = iter.nextStatement();
            uriInfo += stmt.toString()
                    .replace("[","")
                    .replace("]","")
                    .replace("\"","")
                    .replace(",","")
                    + "\n";
        }

        return uriInfo;
    }
}
```