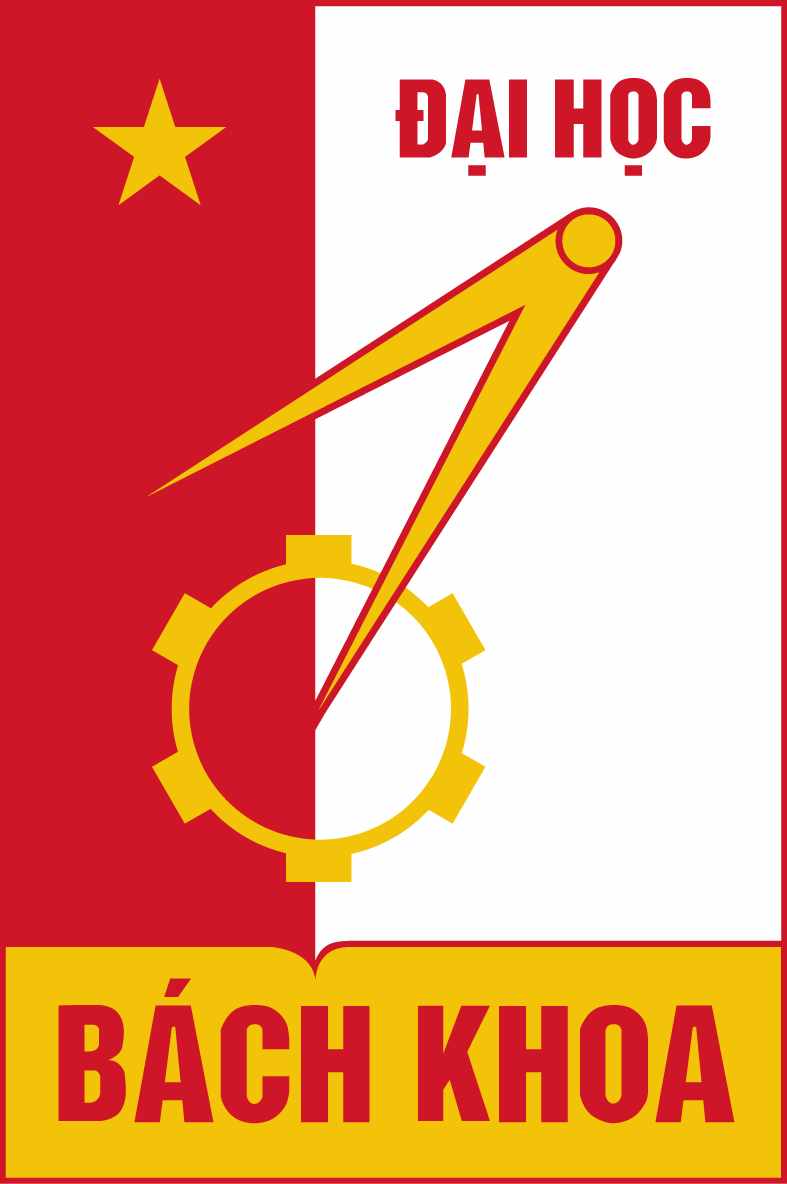
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

****

**Đề tài:**

**Tìm hiểu về công nghệ lưu trữ GraphDB**

Giảng viên hướng dẫn: TS. Trịnh Tuấn Đạt

Bộ môn công nghệ phần mềm

Lớp 104404 - nhóm 5:

Trần Trung Huỳnh 20161917 %

Dương Thị Thoa 20163902 %

Đinh Mạnh Hùng 20161925 %

Lâm Phúc Anh Tú 20167974 %

Lê Xuân Long 20162521 %

MỤC LỤC

[I. Nhiệm vụ của từng thành viên 3](#_Toc533285466)

[II. Tìm hiểu về công nghệ 4](#_Toc533285467)

[1. Giới thiệu về công nghệ 4](#_Toc533285468)

[2. Các thuật ngữ 4](#_Toc533285469)

[3. Hướng dẫn cài đặt 6](#_Toc533285470)

[4. Các kỹ thuật sử dụng 7](#_Toc533285471)

[5. Cách truy vấn 9](#_Toc533285472)

[6. Mô hình dữ liệu 11](#_Toc533285473)

[7. Cách giao tiếp bằng API 12](#_Toc533285474)

[(Tập chung vào các kiến thức nhóm đã dùng) 12](#_Toc533285475)

[III. Mô hình dữ liệu 13](#_Toc533285476)

[1. Mô hình dữ liệu 13](#_Toc533285477)

[2. Cấu trúc dữ liệu mô phỏng 13](#_Toc533285478)

[3. Giải thuật sinh dữ liệu mô phỏng 13](#_Toc533285479)

[IV. Chương trình 14](#_Toc533285480)

[1. Các package 14](#_Toc533285481)

[V. Kiểm thử hiệu năng 15](#_Toc533285482)

# I. Nhiệm vụ của từng thành viên

|  |  |  |
| --- | --- | --- |
| Tên | Phần trăm | Nhiệm vụ |
| Trần Trung Huỳnh |  |  |
| Dương Thị Thoa |  |  |
| Đinh Mạnh Hùng |  |  |
| Tú |  |  |
| Long |  |  |

# II. Tìm hiểu về công nghệ

## Giới thiệu về công nghệ

GraphDB là một cơ sở dữ liệu đồ thị mạnh mẽ và hiệu quả cao với sự hỗ trợ RDF và SPARQL. Nó tối ưu hoá việc tải và sử dụng các bộ dữ liệu điện toán đám mây được liên kết với nhau cũng như những tài nguyên của riêng bạn.GraphDB sử dụng RDF4J như là một thư viện, nhờ đó tận dụng được các API của RDF4J cho việc lưu trữ và truy vấn, cũng như hỗ trợ nhiều loại ngôn ngữ truy vấn như SPARQL, SeRQL, … và các cú pháp của RDF như RDF/XML, N3, Turtle, …

## Các thuật ngữ

**2.1. Resource Description Framework – RDF (Bộ khung mô tả tài nguyên)**

RDF là một tiêu chuẩn trao đổi dữ liệu trên Web. RDF là một tập hợp các nguyên tắc dành cho ngôn ngữ đánh dấu, cung cấp mô hình dữ liệu và cú pháp đơn giản sao cho các hệ thống độc lập có thể trao đổi và sử dụng. Đồng thời, nó được thiết kế sao cho hệ thống máy tính có thể hiểu được và có thể đọc được thông tin, chứ không chỉ đơn giản là để trình bày dữ liệu cho người dùng. Cú pháp của RDF được xây dựng dựa trên mô hình dữ liệu và mô hình này ảnh hưởng đến cách thức mà những thuộc tính được mô tả và nó cũng làm cho cấu trúc của những mô tả đó trở nên rõ ràng.

**2.2. Bộ ba Triples: Subject-Predicate-Object**

RDF sử dụng một mô hình trừu tượng để phân rã thông tin/kiến thức thành những mảnh con, với 1 số luật cơ bản về ngữ nghĩa cho các mảnh này. Mục tiêu là cung cấp 1 phương thức chung mà đủ đơn giản và linh hoạt để diễn giải bất kỳ sự thật (fact) nào, nhưng có cấu trúc để các ứng dụng máy tính có thể hiểu và diễn giải cấu trúc đó.

Mô hình trừu tượng gồm các thành phần chính sau:

* Statement (phát biểu hay mệnh đề).
* Predicate.
* Các nguồn tài nguyên subject và object.

Với các thành phần trên, chúng ta có 1 cấu trúc đồ thị của 1 phát biểu RDF như sau:



Mỗi mảnh kiến thức được phân ra từ mô hình RDF gọi là một statement, một statement RDF (phát biểu RDF) phải có định dạng sau: “subject-predicate-object “, thứ tự này không bao giờ được thay đổi (cố định). với subject và object là tên của 2 thứ bất kỳ và predicate là tên của 1 quan hệ kết nối giữa 2 thứ đó. Theo sơ đồ trên, chúng ta có subject chứa trong hình oval bên trái, subject chứa trong hình oval bên phải, và predicate là tên nhãn của mũi tên chỉ từ subject sang object.

Các statement luôn có 3 thành phần subject-predicate-object, vì vậy người ta thường gọi statement là 1 bộ ba (triple). Mỗi triple hay statement thể hiện 1 sự thật đơn (single fact), một tập các statement hay triple thể hiện 1 vài mảnh thông tin/kiến thức và hình thành nên mô hình RDF.

**2.3. Uniform Resource Identifiers – URIs (Mã định danh tài nguyên)**

"URI" là Uniform Resource Identifier. Nó là nhãn được sử dụng để xác định duy nhất các loại thực thể khác nhau trong biểu đồ RDF. Một URI điển hình trông rất giống địa chỉ web: http://example.org/project/ class#number. Mặc dù giống nhau, URI có thể là địa chỉ web hoặc không tùy người dùng định nghĩa. Nó chỉ đơn giản là một nhãn. URI được dùng với bộ mã ASCII còn IRI (được coi là bản nâng cấp của URI được dùng với mã UNICODE. IRI có 2 phần cơ bản, prefix và local. Một IRI có dạng như sau:

<http://abc.com/DEMO>

* Trong đó http://abc.com/ là prefix, đây là 1 đường dẫn bắt kì, nhưng nên bắt đầu bằng http://
* DEMO là phần local.

## 2.4. Literal

Literal cũng là 1 dạng RDF, nó thường được dùng để lưu trữ các dữ liệu có dạng String là chủ yếu, Literal chỉ có 1 thành phần, chính là thành phần mang ý nghĩa của literal.

## 2.5. Resource description

"Resource description" được định nghĩa là một tập hợp các triples mà tất cả đều có cùng một URI trong trường chủ đề. Nói cách khác, triples đều mô tả các thuộc tính của cùng một thứ.

## Hướng dẫn cài đặt

**3.1. Trên Windows**

* Bước 1: Tải tệp tin theo đường dẫn dưới đây:

<http://go.pardot.com/e/45622/b1b6b38-GraphDB-Free-8-8-0-exe/6l8mpj/1405360163?h=sBC4-S5zLBzNNozFnC77ZfCIolQig7oEDEAQb9nHjjU>

* Bước 2: Chạy tệp tin và làm theo chỉ dẫn trên màn hình
* Bước 3: Khởi chạy ứng dụng GraphDB trên màn hình Desktop. Cơ sở dữ liệu và công cụ quản trị được mở tại <http://localhost:7200/>

**3.2. Trên MacOS**

* Bước 1: Tải tệp tin .dmg

<http://go.pardot.com/e/45622/b1b6b38-GraphDB-Free-8-8-0-dmg/6l8mpl/1405360163?h=sBC4-S5zLBzNNozFnC77ZfCIolQig7oEDEAQb9nHjjU>

* Bước 2: Chạy tệp tin để mở ổ đĩa ảo và sao chép nội dung trong ổ ảo vào thư mục ứng dụng trong ổ đĩa cứng
* Bước 3: Khởi chạy ứng dụng GraphDB bằng cách nhấp chuột vào biểu tượng. Cử sở dữ liệu và công cụ quản trị được mở tại <http://localhost:7200/>

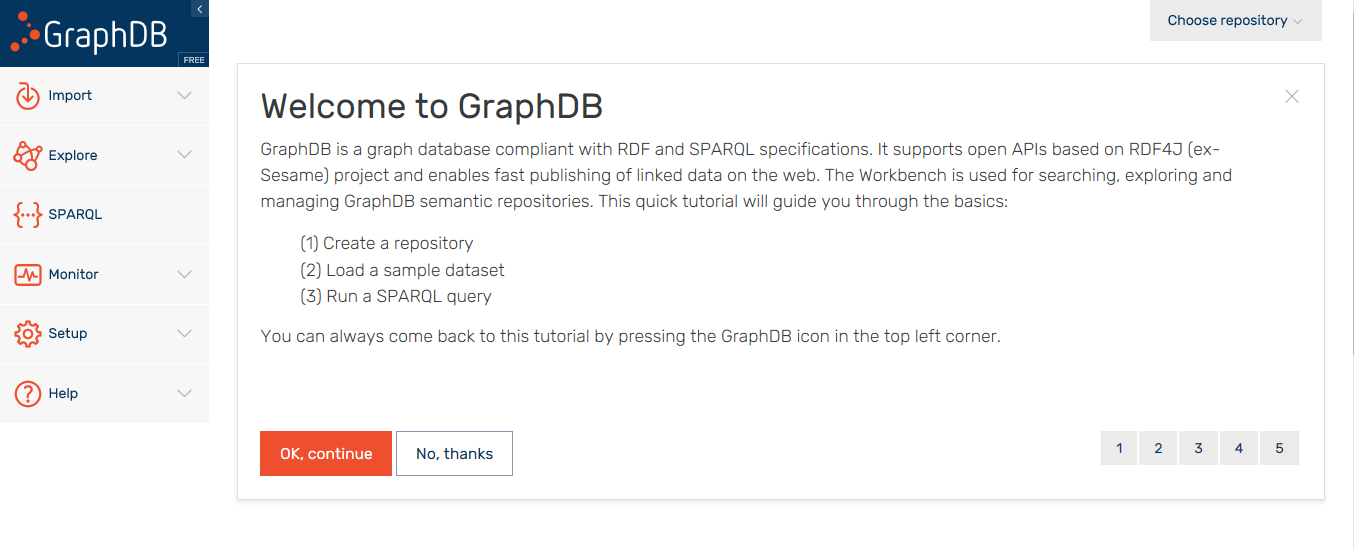
**3.3. Trên Linux**

* Bước 1: Tải tệp tin .rpm hoặc .deb

<http://go.pardot.com/e/45622/b1b6b38-graphdb-free-8-8-0-deb/6l8mpn/1405360163?h=sBC4-S5zLBzNNozFnC77ZfCIolQig7oEDEAQb9nHjjU>

<http://go.pardot.com/e/45622/raphdb-free-8-8-0-1-x86-64-rpm/6l8mpq/1405360163?h=sBC4-S5zLBzNNozFnC77ZfCIolQig7oEDEAQb9nHjjU>

* Bước 2: Cài đặt tệp tin với lệnh sudo rpm -i hoặc sudo dpkg -i cộng với tên của tệp đã tải
* Bước 3: Khởi chạy ứng dụng GraphDB bằng cách nhấp chuột vào biểu tượng. Cơ sở dữ liệu và công cụ quản trị được mở tại <http://localhost:7200/>

****

## 3.4. Chạy như một server riêng lẻ.

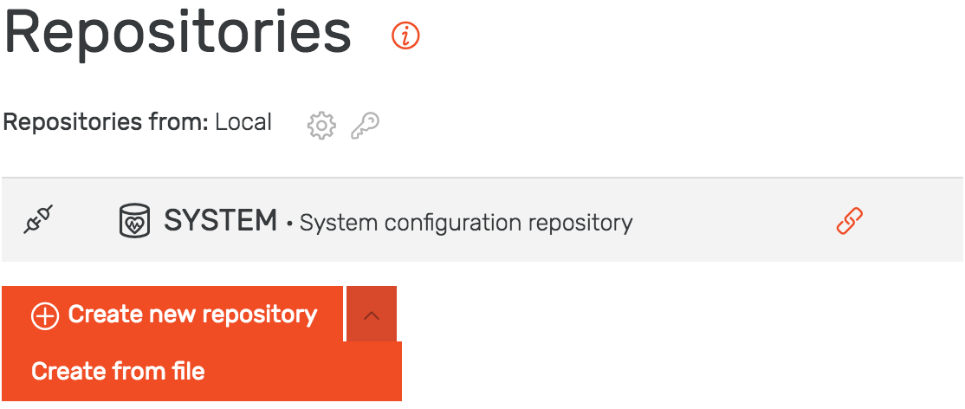
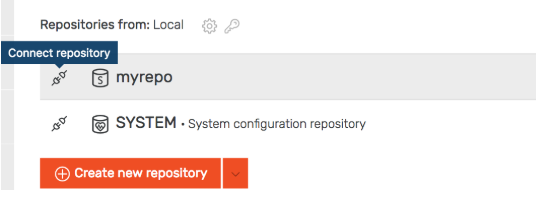
- Bước 1: Tải tài liệu theo link sau:

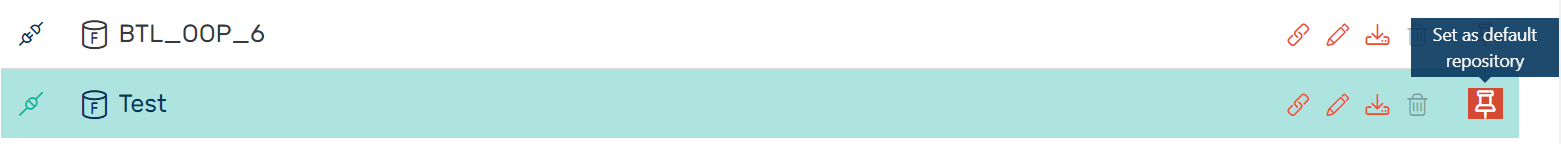
<http://go.pardot.com/e/45622/38-graphdb-free-8-8-0-dist-zip/6l8mps/1405360163?h=sBC4-S5zLBzNNozFnC77ZfCIolQig7oEDEAQb9nHjjU>

* Bước 2: Giải nén được thư mục graphdb-free-8.7.1 và vào trong graphdb-free-8.7.1\bin khởi chạy graphdb.cmd
* Bước 3: server đã bắt đầu chạy ở <http://localhost:7200/>

## Các kỹ thuật sử dụng

**4.1. Tạo một repository mới**

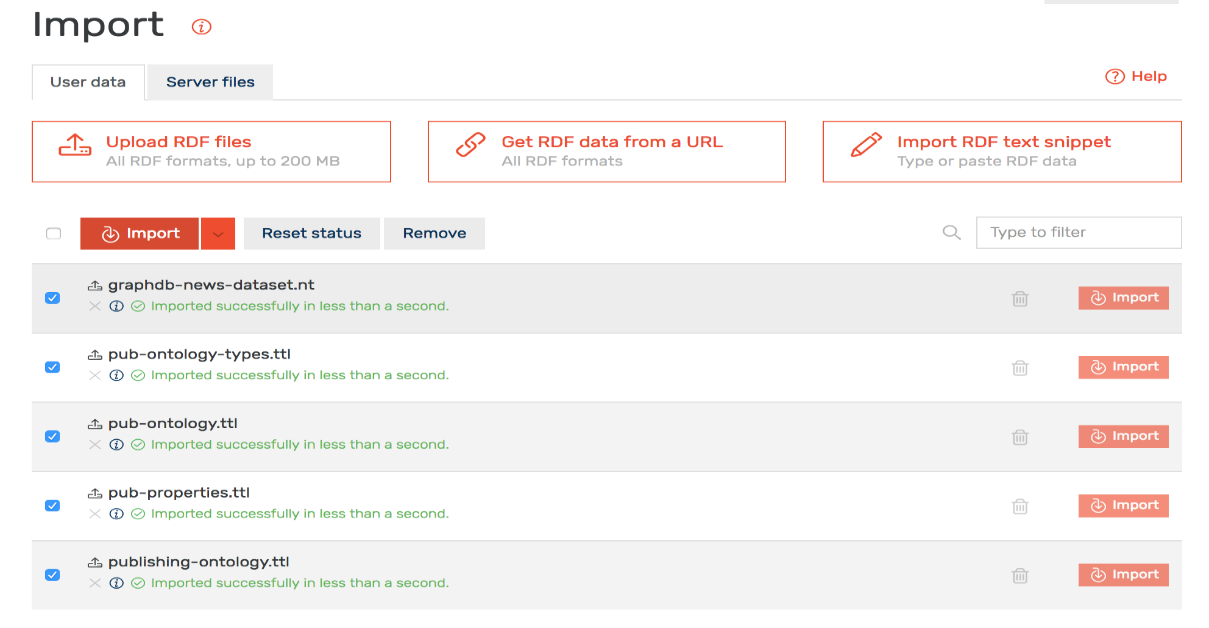
* Bước 1: Chọn *Setup -> Repositories*
* Bước 2: Chọn *Creat new repository*
* Bước 3: Điền *Repository ID* (ví dụ: myrepo), cách tuỳ chọn khác để giá trị mặc định
* Bước 4: Chọn nút *Connect* để đặt repository muốn sử dụng
* Bước 5: Chọn nút *Set as default repository* để đặt repository mặc định



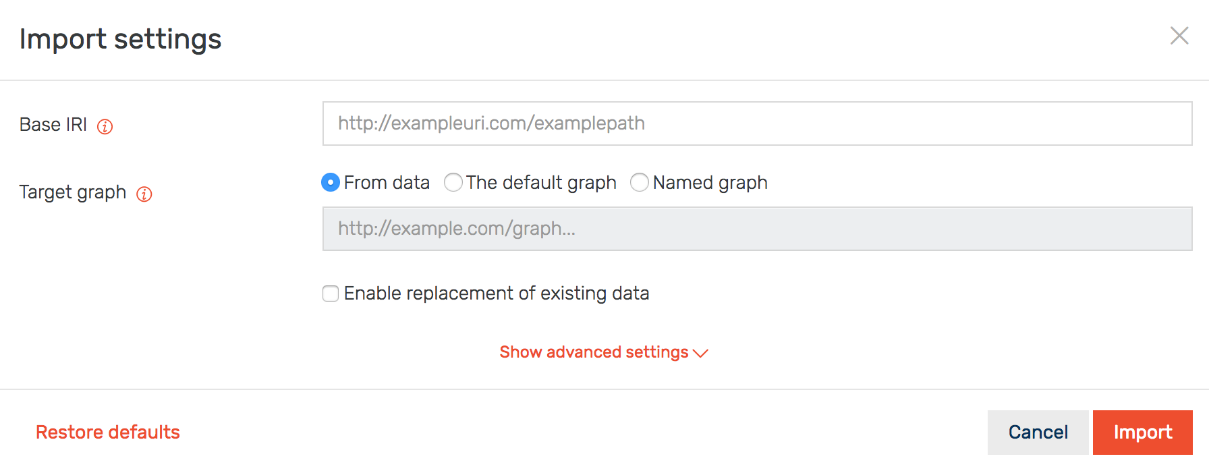
**4.2. Thêm dữ liệu vào repository và chuẩn bị API**

GraphDB hỗ trợ tải dữ liệu được lưu trữ sẵn trong máy thông qua công cụ quản trị GraphDB Workbench hoặc dùng API của RDF4J để thêm dữ liệu.

**4.2.1. Thông qua công cụ quản trị GraphDB Workbench**

* Bước 1: Chọn *Import -> RDF*

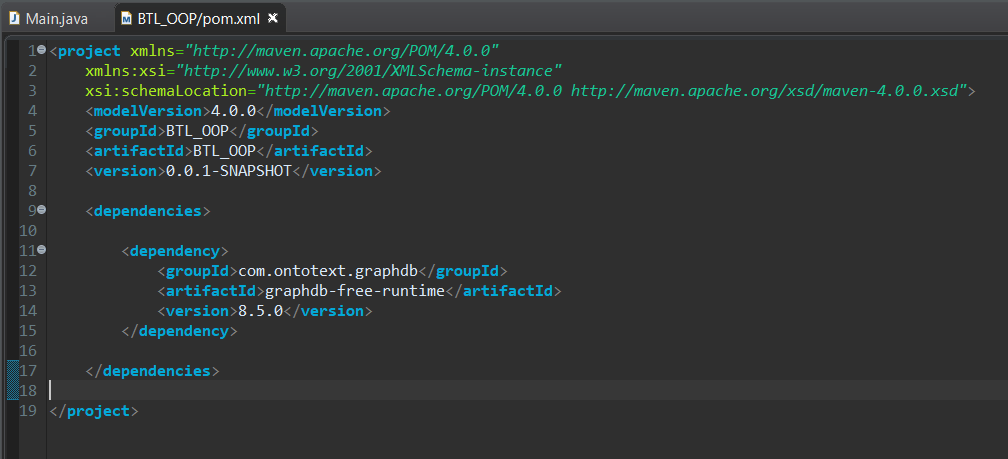
- Bước 2: Chọn tab *User Data* và chọn *Upload RDF files* để tải file lên repository

**-** Bước 3: Chọn nút *Import*, xuất hiện của sổ *Import settings*

**-** Bước 4: Nhập các thông số và ấn nút *Import* để tiến hành tải lên

**4.2.2. Thông qua API của RDF4J**

* Do GraphDB xây dựng dựa trên RDF4J nên ta có thể sử dụng các API của RDF4J để thêm dữ liệu từ máy khách bằng cách kết nối đến repository thông qua giao thức http.
* Chuẩn bị API:
* Đầu tiên cấu hình file pom.xml trong project maven như sau:



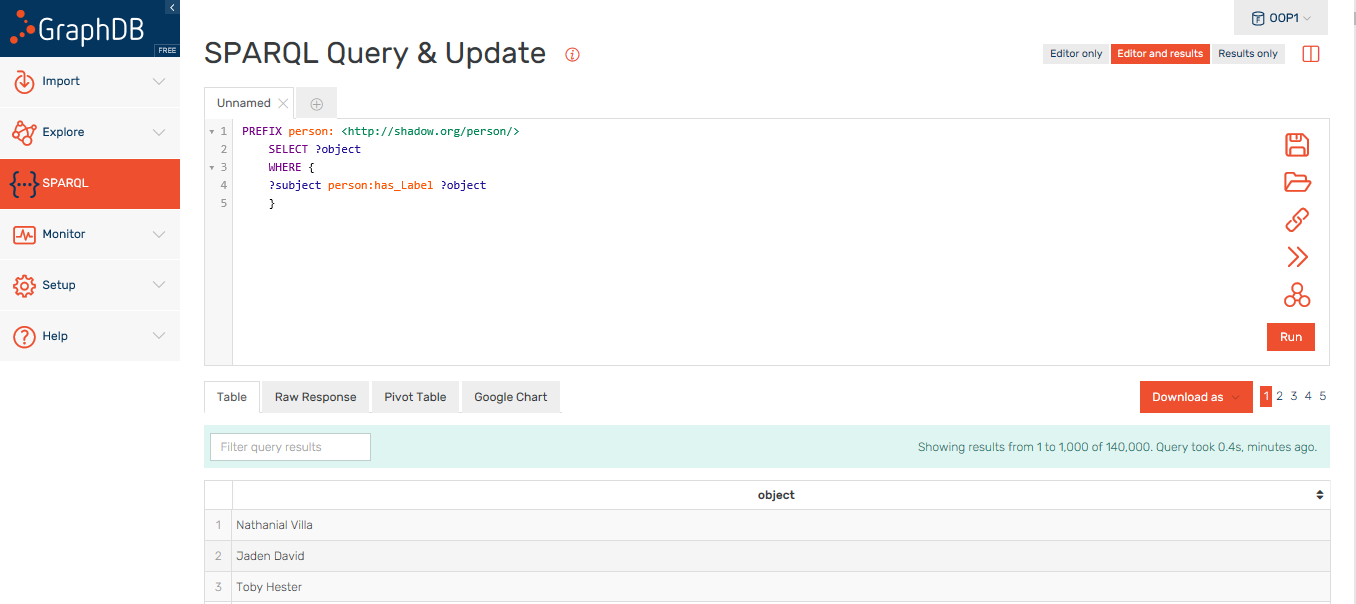
* Thứ hai tải file như hình dưới về máy tạo một thư mục trong project để nó trong đó.
* Thứ ba, kích chuột phải vào file.jar rồi chọn *Build Path* rồi chọn *Buil.*
* Thứ tư, kích chuột phải vào project chọn *Maven -> Update project.* Vậy là đã xog phần chuẩn bị API.

## Cách truy vấn

## Cách truy vấn

**5.1. Truy vấn GraphDB Wordbench**

Mở GraphDB Wordbench trong trình duyệt, trong menu chọn SPARQL và tiến hành gõ câu lệnh truy vấn vào khung text trong mục SPARQL, Run để xem kết quả hiện thị bằng bảng.



**5.2. Truy vấn bằng API của RDF4J**

**-** Do GraphDB được xây dựng với RDF4J là thư viện nên ta có thể sử dụng các API của nó để truy vấn dữ liệu bằng cách kết nối đến repository và nhập câu lệnh truy vấn trong lớp Query, khởi chạy chương trình và kết quả được ghi ở file ResultQueryBasis và ResultQuery.

- Bước 1: Kết nối đến repository

private static RepositoryConnection getRepositoryConnection() {

Repository repository = new HTTPRepository(GRAPHDB\_SERVER, REPOSITORY\_ID);

try{

repository.initialize();

RepositoryConnection repositoryConnection =

repository.getConnection();

return repositoryConnection;

}

catch (RepositoryException Ex)

{

Ex.printStackTrace();

}

Return null;

}

- Bước 2: Truy vấn

public void strQuery;

{

strQuery = “PREFIX person: <http://shadow.org/person/>\n” +

“SELECT ?object\n” +

“WHERE {\n” +

“?subject person:has\_Label ?object.\n” +

“}”;

TupleQuery tupleQuery = conn.prepareTupleQuery();

TupleQueryResult result = null;

try

{

result = tupleQuery.evaluate();

while (result.hasNext())

{

BindingSet bindingSet = result.next();

if(bindingSet.getValue("subject") != null)

{

System.***out***.print(bindingSet.getValue("subject") + "\t");

}

if(bindingSet.getValue("predicate") != null)

{

System.***out***.print(bindingSet.getValue("predicate") + "\t");

}

if(bindingSet.getValue("object") != null)

{

System.***out***.println(handingQuery(bindingSet.getValue("object").toString()));

}

}

}

catch (QueryEvaluationException qee)

{

qee.getStackTrace();

} finally {

result.close();

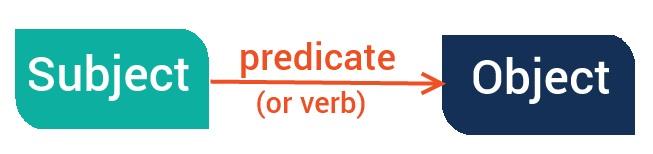
}

}

## Mô hình dữ liệu

- Các tài nguyên có thể được mô tả thông qua các phát biểu cho phép xác định thuộc tính và giá trị. RDF sử dụng bộ ba Triples để nói về những thành phần khác nhau của phát biểu. Một bộ ba được mô tả dưới dạng nút-cung-nút, cung có hướng bắt đầu từ subject đến object

* Subject (chủ ngữ): Là thành phần xác định đối tượng mà phát biểu đề cập
* Predicate (vị ngữ): Là thành phần xác định thuộc tính của chủ ngữ trong phát biểu
* Object (tân ngữ): Là thành phần xác định giá trị của thuộc tính, có thể là một tài nguyên (URI), hoặc một giá trị (Literal)



Tập hợp các bộ ba hợp lại tạo thành đồ thị RDF, các nút trong đồ thị có thể là subject hoặc object, các cung trong đồ thị là predicate

## Cách giao tiếp bằng API

(Tập chung vào các kiến thức nhóm đã dùng)

# III. Mô hình dữ liệu

## Mô hình dữ liệu

## Cấu trúc dữ liệu mô phỏng

## Giải thuật sinh dữ liệu mô phỏng

# IV. Chương trình

## Biểu đồ thể hiện sự phụ giữa các package

* Chương trình có các package là: graphdb, create.entity, main, model, writefile.io.
* Tác dụng của các lớp:

## Biểu đồ quan hệ giữa các lớp

# V. Kiểm thử hiệu năng

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (N, M) | TVCB1 | TVCB2 | TVCB3 | TVCB4 | TVCB5 | TVCB6 | TVCB7 | TVCB8 | TVCB9 | TVCB10 |
| 10000, 20000 |  |  |  |  |  |  |  |  |  |  |
| 40000, 56000 |  |  |  |  |  |  |  |  |  |  |
| 300000, 400000 |  |  |  |  |  |  |  |  |  |  |
| 1000000, 2000000 |  |  |  |  |  |  |  |  |  |  |
| 4000000, 8000000 |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (N, M) | TVNC1 | TVNC2 | TVNC3 | TVNC4 | TVNC5 | TVNC6 | TVNC7 | TVNC8 | TVNC9 | TVNC10 |
| 10000, 20000 |  |  |  |  |  |  |  |  |  |  |
| 40000, 56000 |  |  |  |  |  |  |  |  |  |  |
| 300000, 400000 |  |  |  |  |  |  |  |  |  |  |
| 1000000, 2000000 |  |  |  |  |  |  |  |  |  |  |
| 4000000, 8000000 |  |  |  |  |  |  |  |  |  |  |