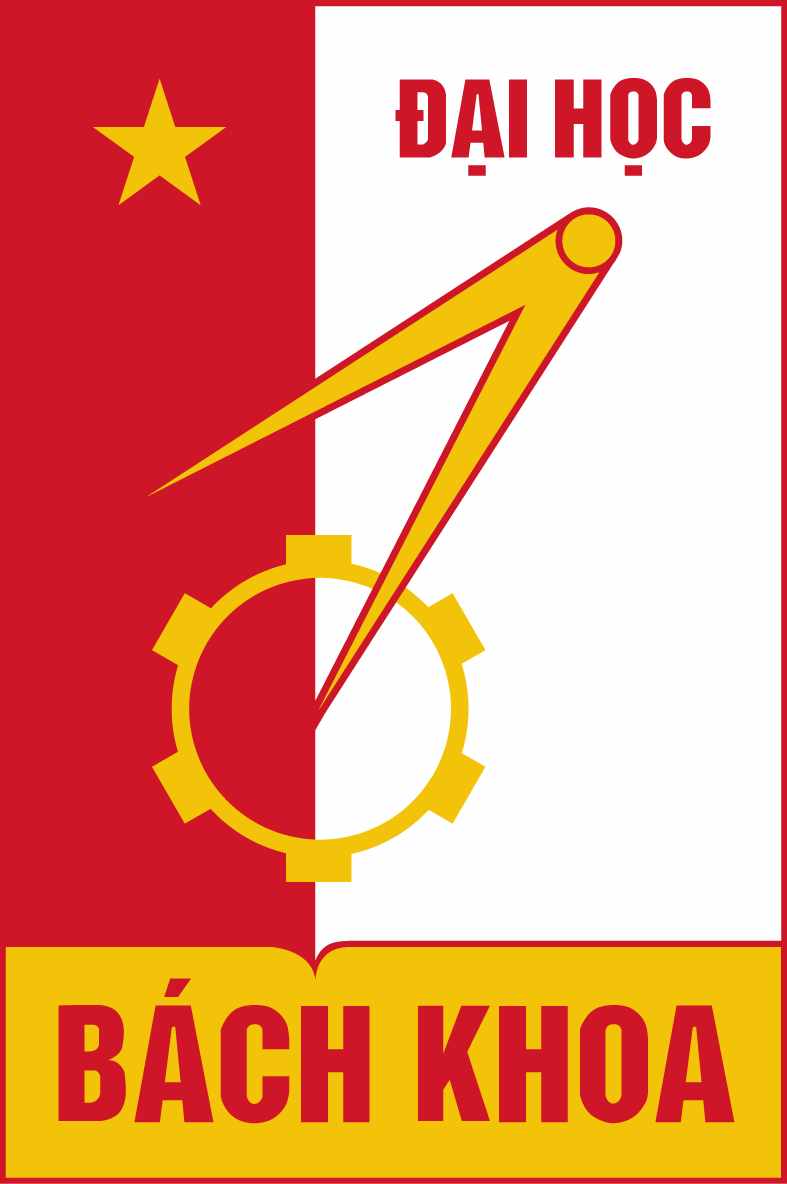
TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI

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

Nhóm 5:

Trần Trung Huỳnh 20161917

Dương Thị Thoa 2016

Đinh Mạnh Hùng 2016

Tú 2016

Long 2016

Nội dung

[I. Tìm hiểu về công nghệ 3](#_Toc533282163)

[1. Giới thiệu 3](#_Toc533282164)

[2. Các thuật ngữ 3](#_Toc533282165)

[3. Hướng dẫn chi tiết các bước cài đặt 5](#_Toc533282166)

[3.1. Trên Windows 5](#_Toc533282167)

[III. Mô hình dữ liệu 11](#_Toc533282168)

# Tìm hiểu về công nghệ

# 1. Giới thiệu

Onotext 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, …

# 2. Các thuật ngữ

**2.1. Semantic Web (Web ngữ nghĩa)**

Semantic Web là một mạng lưới thông tin được liên kết theo phương pháp sao cho thông tin có thể dễ dàng được xử lý bởi các máy tính trên toàn cầu. Nó được xem là cách mô tả thông tin rất hiệu quả trên World Wide Web, và nó cũng được xem là một cơ sở dữ liệu có khả năng liên kết toàn cầu.

Các công nghệ Semantic Web giúp con người có thể tạo nên các kho dữ liệu trên web, xây dựng bộ từ vựng và viết các luật để xử lý dữ liệu.

**2.2 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. 3. 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.

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

URI là một chuỗi ký tự được sử dụng để nhận diện các tài nguyên. URI xác định tài nguyên theo vị trí, theo tên, hoặc cả hai. URI có hai nhánh là URL và URN. Tài nguyên ở đây là tài nguyên mạng như các siêu văn bản, file hình ảnh, file âm thanh, file video…

Với một số URI, trình duyệt không thể yêu cầu nội dung trực tiếp, tuy nhiên nó vẫn là nhận diện một tài nguyên duy nhất và có thể có những ứng dụng máy chủ tạo ra dịch vụ để chuyển nó thành nội dung mà trình duyệt có thể yêu cầu được.

**2.5. SPARQL**

SPARQL là ngôn ngữ và giao thức truy vấn tiêu chuẩn cho Dữ liệu Mở Liên kết (Linked Open Data) trên web hoặc trong cơ sở dữ liệu đồ họa ngữ nghĩa (còn được gọi là RDF triplestore - bộ 3 RDF - Resource Description Framework).

SPARQL là viết tắt của “SPARQL Protocol and RDF Query Language”, cho phép những người sử dụng truy vấn thông tin từ các cơ sở dữ liệu hoặc bất kỳ nguồn dữ liệu nào có thể được ánh xạ tới RDF. SPARQL được chuẩn hóa bởi RDF Data Access Working Group (DAWG) của World Wide Web Consortium(W3C), và hiện được coi là một công nghệ Semantic Web quan trọng. Vào ngày 15 tháng 1 năm 2008, SPARQL đã trở thành Khuyến nghị chính thức của W3C.

**2.6. API – Application Programming Interface**

API là một giao diện mà một hệ thống máy tính hay ứng dụng cung cấp để cho phép các yêu cầu dịch vụ có thể được tạo ra từ các chương trình máy tính khác, cho phép dữ liệu có thể được trao đổi qua lại giữa chúng. Có nhiều loại hệ thống và ứng dụng hiện thực API, như các hệ thống đồ họa, cơ sở dữ liệu, mạng, dịch vụ web, và ngay cả một số trò chơi máy tính. Đây là phần mềm hệ thống cung cấp đầy đủ các chức năng và các tài nguyên mà các ta có thể rút ra từ đó để tạo nên các tính năng giao tiếp người- máy như: các trình đơn kéo xuống, tên lệnh, hộp hội thoại, lệnh bàn phím và các cửa sổ.

# 3. Hướng dẫn chi tiết các bước cài đặt

# 3.1. Trên Windows

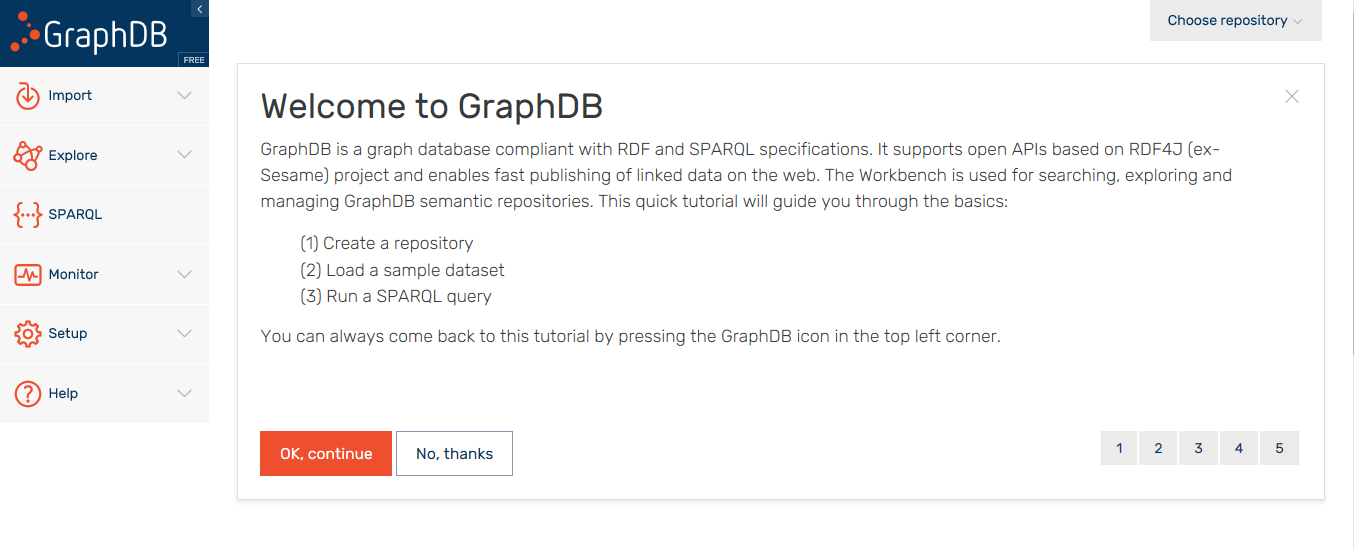
* Bước 1: Tải tệp tin .exe
* Bước 2: Chạy tệp tin và làm theo chỉ dẫn trên màn hình
* Bước 3: Khởi chạy ứng dụng GraphDB trên màn hình Desktop. Cơ sở dữ liệu và công cụ quản trị được mở tại <http://localhost:7200/>

**3.2. Trên MacOS**

* Bước 1: Tải tệp tin .dmg
* 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/>

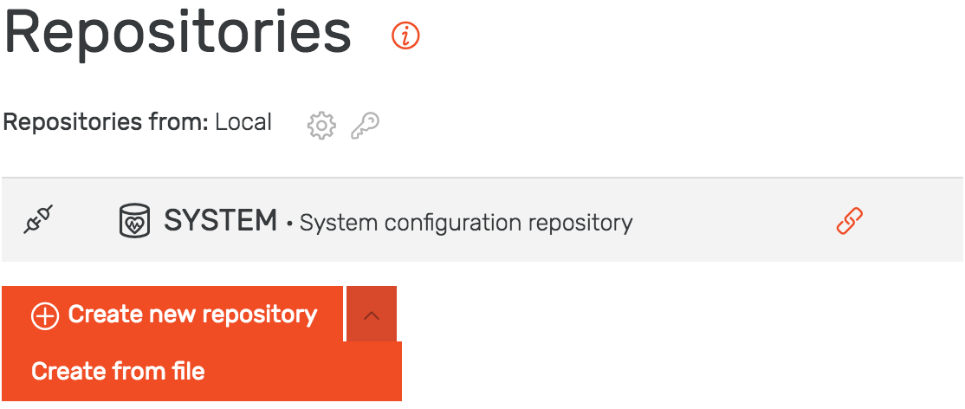
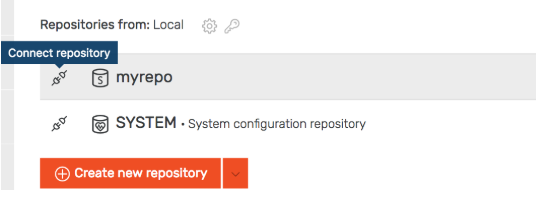
**3.3. Trên Linux**

* Bước 1: Tải tệp tin .rpm hoặc .deb
* 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/>

****

**4. Kĩ thuật sử dụng**

**4.1. Tạo một repository mới**

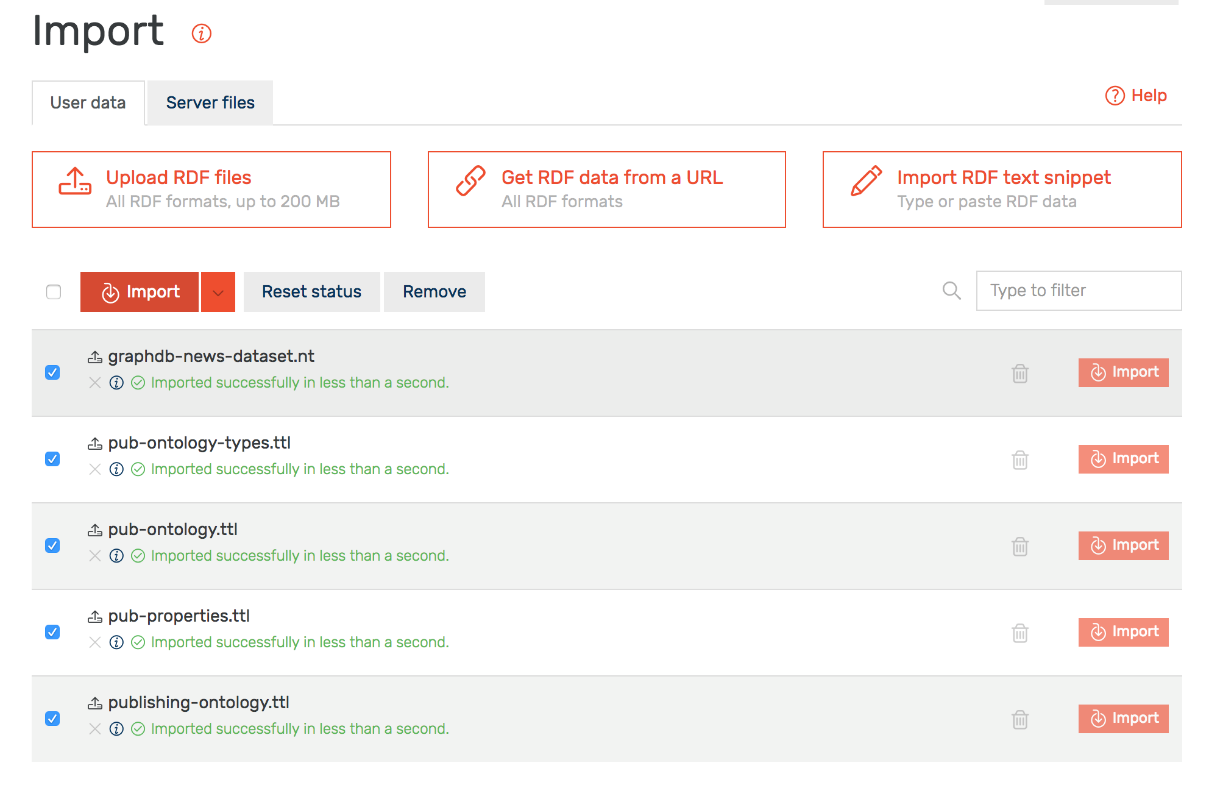
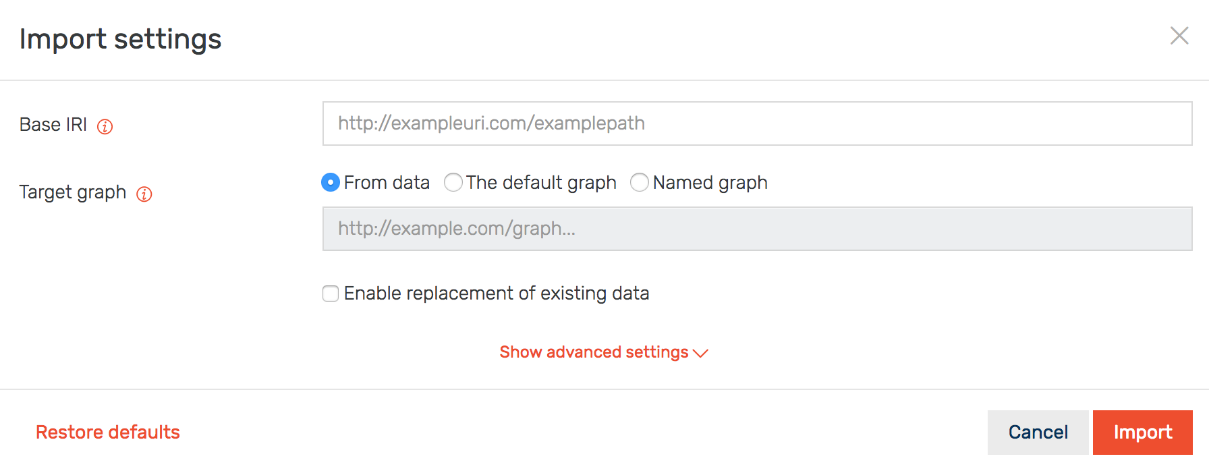
* Bước 1: Chọn *Setup -> Repositories*
* Bước 2: Chọn *Creat new repository*
* Bước 3: Điền *Repository ID* (ví dụ: myrepo), cách tuỳ chọn khác để giá trị mặc định
* Bước 4: Chọn nút *Connect* để đặt repository này thành repository cho máy tính này
* Bước 5: Chọn nút *pin* để đặt repository mặc định

**4.2. Thêm dữ liệu vào repository**

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

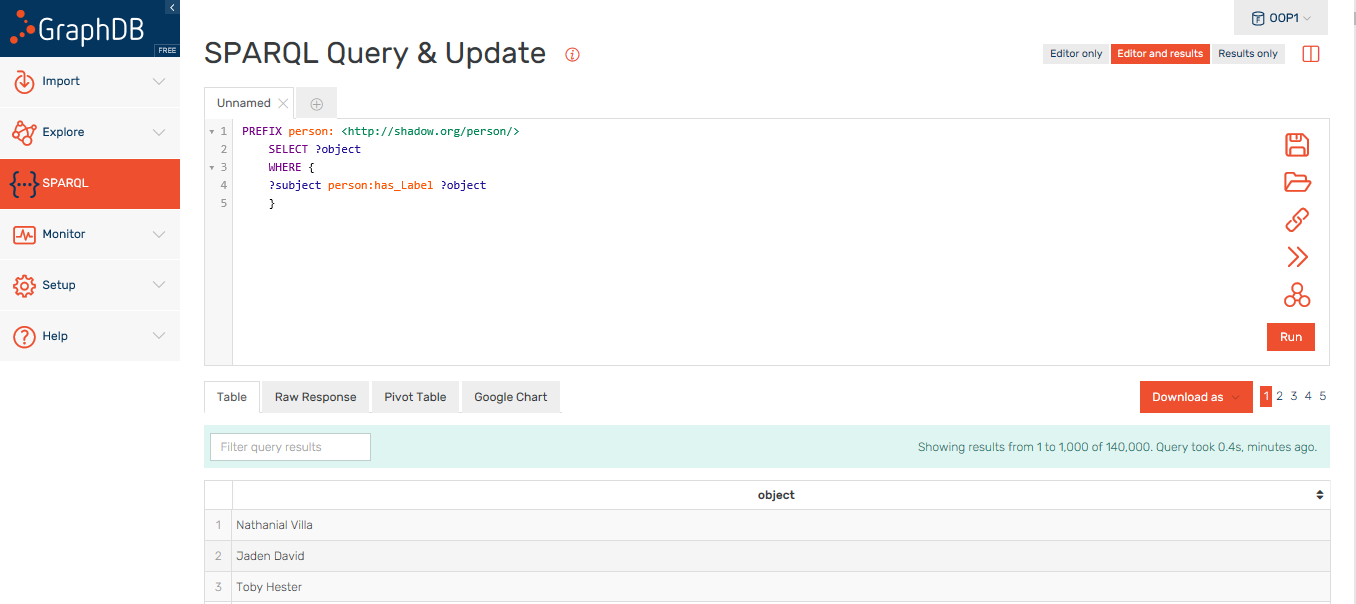
**4.2.2. Thông qua API của RDF4J**

* Do GraphDB xây dựng dựa trên RDF4J nên ta có thể sử dụng các API của RDF4J để thêm dữ liệu bằng cách kết nối đến repository thông qua giao thức http

**5. 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();

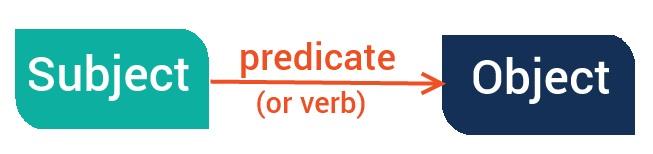
}

}

**6. 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  
**7. Cách giao tiếp bằng API**

# III. Mô hình dữ liệu

IV. Biểu đồ lớp của chương trìnhV. Xây dựng thông số kiểm thử hiệu năng của hệ thống.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (N, M) | TVCB1 | TVCB2 | TVCB3 | TVCB4 | TVCB5 | TVCB6 | TVCB7 | TVCB8 | TVCB9 | TVCB10 |
| 100, 200 |  |  |  |  |  |  |  |  |  |  |
| 5000, 7000 |  |  |  |  |  |  |  |  |  |  |
| 60000, 80000 |  |  |  |  |  |  |  |  |  |  |
| 500000, 1000000 |  |  |  |  |  |  |  |  |  |  |
| 1000000, 2000000 |  |  |  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| (N, M) | TVNC1 | TVNC2 | TVNC3 | TVNC4 | TVNC5 | TVNC6 | TVNC7 | TVNC8 | TVNC9 | TVNC10 |
| 100, 200 |  |  |  |  |  |  |  |  |  |  |
| 5000, 7000 |  |  |  |  |  |  |  |  |  |  |
| 60000, 80000 |  |  |  |  |  |  |  |  |  |  |
| 500000, 1000000 |  |  |  |  |  |  |  |  |  |  |
| 1000000, 2000000 |  |  |  |  |  |  |  |  |  |  |