

Requêter une base de données à un niveau conceptuel avec Ontop

Benjamin Cogrel

Université Libre de Bozen-Bolzano, Italie



Freie Universität Bozen

Libera Università di Bolzano

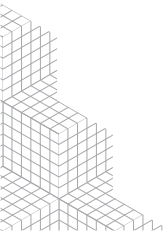
Free University of Bozen-Bolzano

Journées Ontologie en Sciences Humaines et Sociales
Tours, le 8 novembre 2015

Exemple issu des sciences du vivant

Optique

- **Source(s)**: bases de données décrivant des cancers
- **Ontologie**: Le vocabulaire du domaine (Patient, Cancer, LungCancer, etc)



- Java 7 ou 8
- Contenu en ligne :
`https://github.com/ontop/ontop-examples/tree/master/tours-tutoriel-2015`

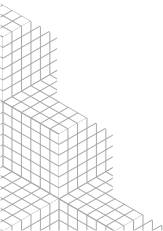


Table : tbl_patient

PatientId	Name	Type	Stage
1	Mary	false	4
2	John	true	7

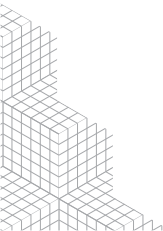


Table : tbl_patient

PatientId	Name	Type	Stage
1	Mary	false	4
2	John	true	7

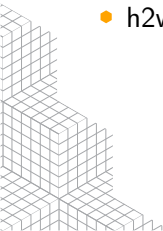
Type:

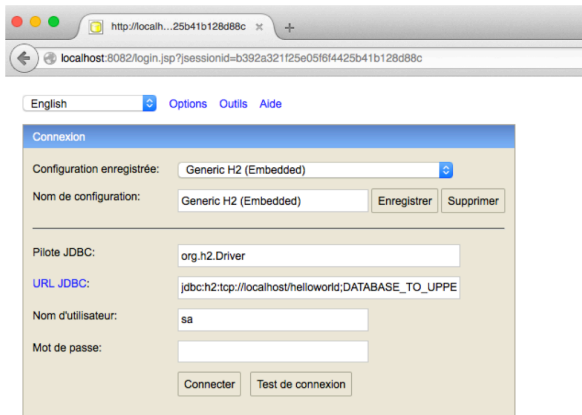
- faux : Non-Small Cell Lung Cancer (NSCLC)
- vrai : Small Cell Lung Cancer (SCLC)

Stage(*niveau d'avancement*):

- 1-6 : NSCLC: I,II,III,IIIa,IIIb,IV
- 7-8 : SCLC: Limited, Extensive

- H2 est une base de données relationnelle écrite en Java
- Décompresser le paquet téléchargé
- Lancer les scripts :
 - Ouvrir un terminal (Mac Terminal.app, Windows exécuter cmd.exe)
 - Aller dans le répertoire H2 (par exemple, `cd h2`)
- `sh h2.sh` (Mac/Linux - Si nécessaire, `"chmod u+x h2.sh "`)
- `h2w.bat` (Windows)





The screenshot shows a web browser window with the address bar displaying `http://localhost:8082/login.jsp?sessionId=b392a321f25e05f6f4425b41b128d88c`. The page title is "Connexion". Below the title, there is a language selector set to "English" and links for "Options", "Outils", and "Aide". The main form contains the following fields and buttons:

- Configuration enregistrée:** A dropdown menu showing "Generic H2 (Embedded)".
- Nom de configuration:** A text input field containing "Generic H2 (Embedded)", with "Enregistrer" and "Supprimer" buttons to its right.
- Pilote JDBC:** A text input field containing "org.h2.Driver".
- URL JDBC:** A text input field containing "jdbc:h2:tcp://localhost/helloworld;DATABASE_TO_UPPE".
- Nom d'utilisateur:** A text input field containing "sa".
- Mot de passe:** A text input field (currently empty).
- Buttons:** "Connecter" and "Test de connexion" buttons at the bottom.

- localhost : adresse de la base de données
- helloworld : nom de la base de données

Tables Info

Write you queries Here!

Commandes principales

- Affiche cette page d'aide
- Affiche l'historique des commandes
- Exécute la commande courante
- Déconnexion de la base de données

Exemple de script SQL

Effacer une table si elle existe	DROP TABLE IF EXISTS TEST;
Créer une nouvelle table avec les colonnes ID et NAME	CREATE TABLE TEST(ID INT PRIMARY KEY, NAME VARCHAR(255));
Ajouter un nouvel enregistrement	INSERT INTO TEST VALUES(1, 'Hello');
Ajouter un autre enregistrement	INSERT INTO TEST VALUES(2, 'World');
Requêter une table	SELECT * FROM TEST ORDER BY ID;
Modifier un enregistrement	UPDATE TEST SET NAME='Hi' WHERE ID=1;
Effacer un enregistrement	DELETE FROM TEST WHERE ID=2;
Aide	HELP ...

Example Queries

Utiliser les scripts create.sql et insert.sql

```
CREATE TABLE "tbl_patient" (  
  patientid INT NOT NULL PRIMARY KEY,  
  name VARCHAR(40),  
  type BOOLEAN,  
  stage TINYINT  
)
```

Ajout de données :

```
INSERT INTO "tbl_patient"  
(patientid,name,type,stage)  
VALUES  
(1,'Mary',false,4),  
(2,'John',true,7);
```

Patients ayant un cancer du poumon non à petites cellules au niveau IIIa ou au-delà (select.sql)

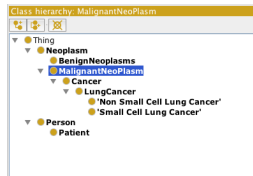
```
SELECT patientid  
FROM "tbl_patient"  
WHERE  
TYPE = false AND stage >= 4
```



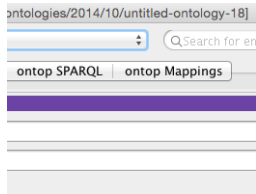
Definition

Un artéfact contenant (i) un vocabulaire et des relations entre les termes de ce vocabulaire et (ii) qui est exprimé dans un langage dont la syntaxe et la sémantique sont partagés et communément acceptés.

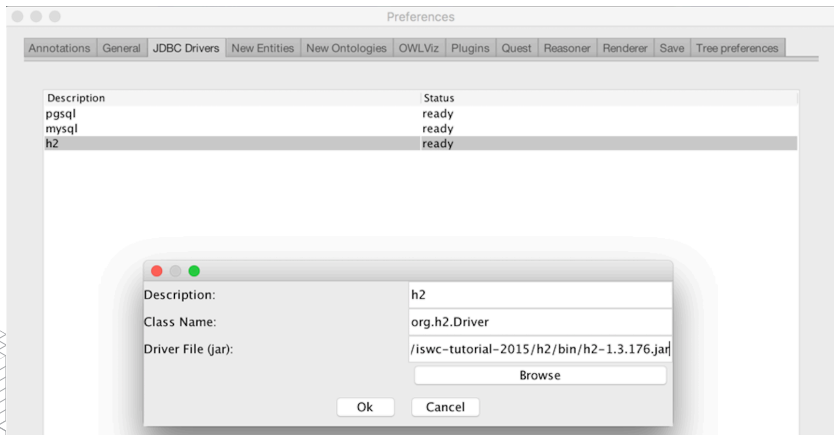
Mike **type** Patient
NSCLC **subClassOf** LungCancer
LungCancer **subClassOf** Cancer



- Aller dans le répertoire protégé-ontop. Il y a un paquet Protégé 5 contenant le plugin Ontop.
- Lancer Protégé depuis la console à partir des commandes `run.bat` ou `run.sh` scripts. Autrement dit, exécuter :
- `cd Protege_5/; run.sh`



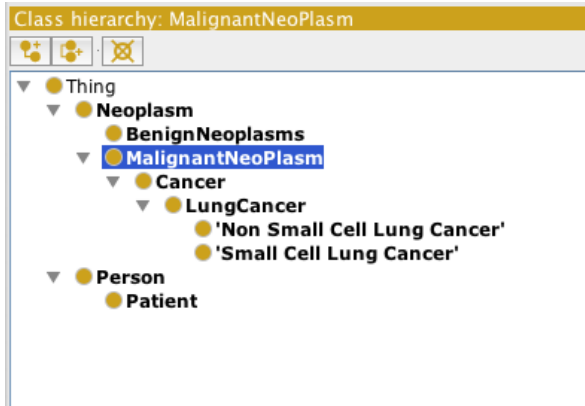
- Ouvrir “Preferences”, “JDBC Drivers” et ajouter la configuration pour H2
 - Description: h2
 - Class Name: org.h2.Driver
 - Driver File (jar): /path/to/h2/bin/h2-1.3.176.jar



L'ontologie : création de classes et de propriétés

Optique

Ajouter la classe Patient :

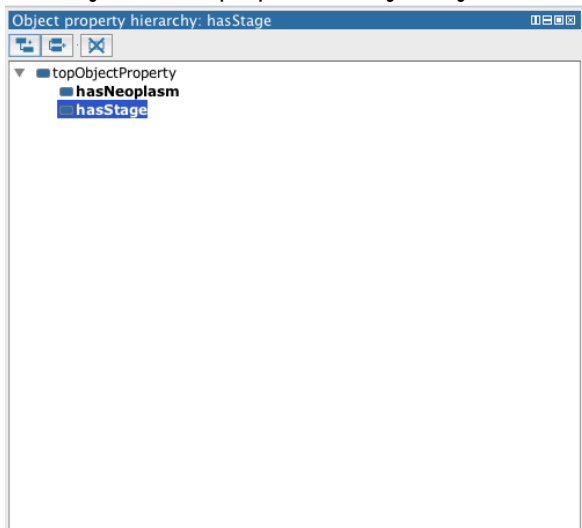


(Voir PatientOnto.owl)

L'ontologie : création de classes et de propriétés

Optique

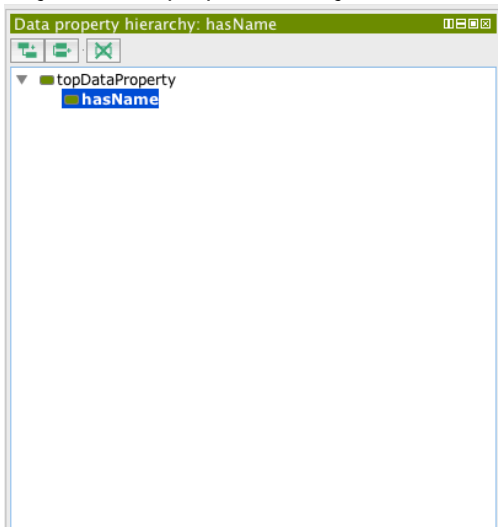
Ajouter ces propriétés “objet-objet” :



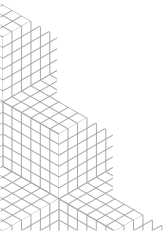
L'ontologie : création de classes et de propriétés

Optique

Ajouter ces propriétés “objet-littéral” :



- On a désormais le vocabulaire et la base de données, on va maintenant les lier.
- Les mappings définissent des triplets (sujet, propriété, objet) à partir de requêtes SQL.

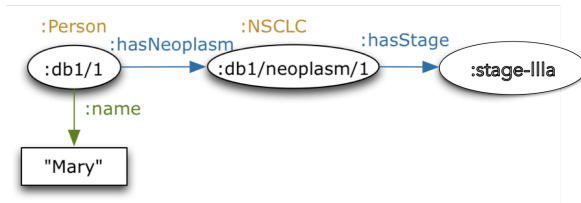


p.Id	Name	Type	Stage
1	Mary	false	2

- $(:db1/\{p.id\}, type, :Patient) \leftarrow \text{Select } p.id \text{ From } tbl_patient$
- $(:db1/\{p.id\}, :hasName, \{name\}) \leftarrow \text{Select } p.id, name \text{ From } tbl_patient$
- $(:db1/\{p.id\}, :hasNeoplasm, :db1/neoplasm/\{p.id\}) \leftarrow$
 $\text{Select } p.id \text{ From } tbl_patient$
- $(:db1/neoplasm/\{p.id\}, :hasStage, :stage-IIIa) \leftarrow$
 $\text{Select } p.id \text{ From } tbl_patient \text{ where } stage=4$

p.Id	Name	Type	Stage
1	Mary	false	2

- $(:db1/\{p.id\}, type, :Patient) \leftarrow \text{Select } p.id \text{ From } tbl_patient$
- $(:db1/\{p.id\}, :hasName, \{name\}) \leftarrow \text{Select } p.id, name \text{ From } tbl_patient$
- $(:db1/\{p.id\}, :hasNeoplasm, :db1/neoplasm/\{p.id\}) \leftarrow \text{Select } p.id \text{ From } tbl_patient$
- $(:db1/neoplasm/\{p.id\}, :hasStage, :stage-IIIa) \leftarrow \text{Select } p.id \text{ From } tbl_patient \text{ where stage}=4$



En utilisant l'onglet Ontop Mapping, définir les paramètres de connexion à notre base de données

Étapes :

1. Basculer vers l'onglet Ontop Mapping
2. Ajouter une nouvelle source de données (la nommer PatientDB)
3. Définir les paramètres de connexion suivants :
 - Connection URL: jdbc:h2:tcp://localhost/helloworld
 - Username: sa
 - Password: (leave empty)
 - Driver class: org.h2.Driver (à choisir dans le menu déroulant)
4. Tester la connexion en cliquant sur le bouton "Test Connection"

Datasource Manager Mapping Manager Mapping Assistant - BETA

Datasource editor:

OBDA Model information

Number of sources: 1

Connection parameters

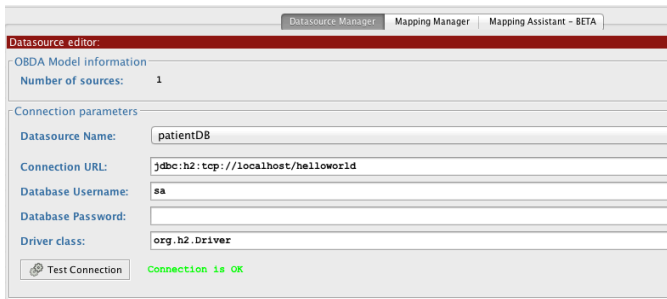
Datasource Name: patientDB

Connection URL: jdbc:h2:tcp://localhost/helloworld

Database Username: sa

Database Password:

Driver class: org.h2.Driver

 Test Connection Connection is OK

- Basculer sur l'onglet "Mapping Manager" dans l'onglet ontop mappings.
- Sélectionner la source de donnée
- Cliquer sur Create:

target: :db1/{patientid} a :Patient .

source: SELECT patientid FROM "tbl_patient"

target: :db1/{patientid} :hasName {name} .

source: Select patientid,name FROM "tbl_patient"

target: :db1/{patientid} :hasNeoplasm :db1/neoplasm/{patientid}.

source: SELECT patientid FROM "tbl_patient"

target: :db1/neoplasm/{patientid} :hasStage :stage-IIIa .

source: SELECT patientid FROM "tbl_patient" where stage=4

- Nous allons classifier la tumeur à partir de notre connaissance de la base de données.
- Nous savons que “false” dans la table patient indique un “Non Small Cell Lung Cancer”, donc nous classifions les tumeurs comme étant des :NSCLC.

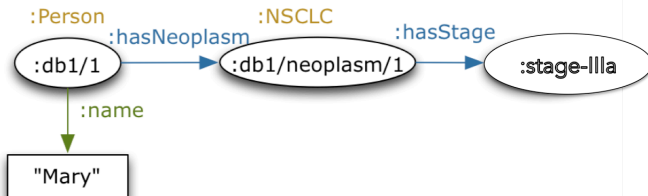
nsclc

```
:db1/neoplasm/{PATIENTID}/ a :NSCLC .  
select * from TBL_PATIENT WHERE TYPE = false
```

sclc

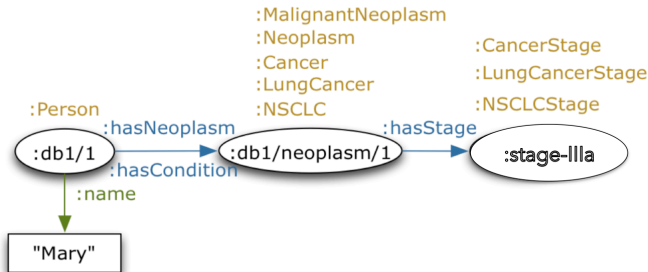
```
:db1/neoplasm/{PATIENTID}/ a :SCLC .  
select * from TBL_PATIENT WHERE TYPE = true
```

Données :

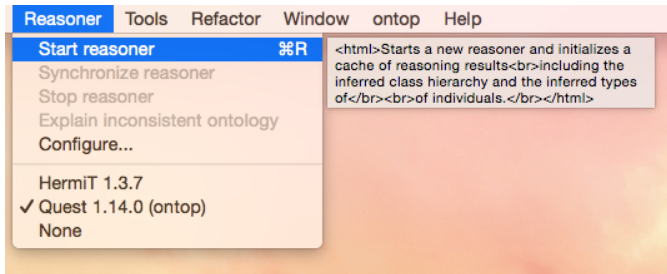


- Le vocabulaire est **centré sur le domaine** et **indépendant de la base de données**.
- Plus besoin de se soucier des codes des types et des états d'avancement.
- Ce vocabulaire rendra **l'intégration de nouvelles sources** plus facile.
- Nos sources de données sont désormais **documentées** !

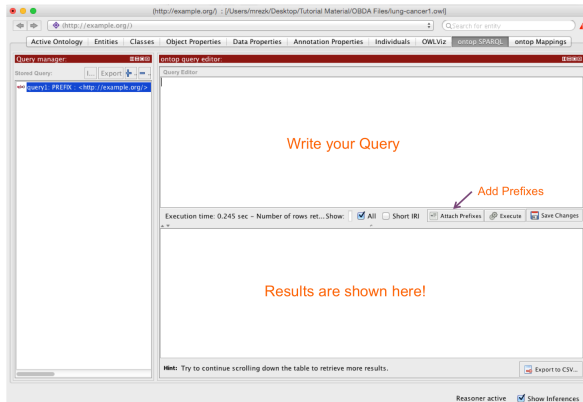
Données et inférence :



- Activer Ontop dans le menu "Reasoner"



- Aller dans l'onglet SPARQL
- Ajouter tous les préfixes

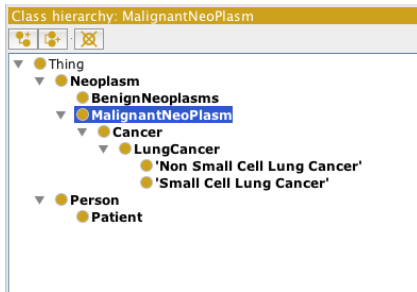


- Besoin d'information : L'identifiant et le nom des patients ayant une tumeur au niveau IIIa.
- Écrire la requête SPARQL suivante :

```
SELECT ?p ?name WHERE
{?p rdf:type :Patient .
?p :hasName ?name .
?p :hasNeoplasm ?tumor .
?tumor :hasStage :stage-IIIa .}
```
- Cliquer sur execute

- Requête : toutes les instances de la classe Neoplasm:

```
SELECT ?x WHERE { ?x a :Neoplasm). }
```



- Requête : toutes les instances de la classe Neoplasm:

```
SELECT ?x WHERE { ?x a :Neoplasm. }
```

- Traduction étape par étape (en théorie) :

```
SELECT ?x WHERE { { ?x a :Neoplasm. }
    UNION
    { ?x a :BenignNeoplasm. }
    UNION
    { ?x a :MalignantNeoplasm. }
    UNION
    :
    { ?x rdf:type :NSCLC). }
    UNION
    { ?x rdf:type :SCLC). } }
```

- Requête : toutes les instances de la classe Neoplasm:

```
SELECT ?x WHERE { ?x a :Neoplasm). }
```

- Traduction étape par étape (en théorie) :

```
SELECT ?x WHERE { {?x a :Neoplasm.}
UNION
{ ?x a :BenignNeoplasm. }
UNION
{ ?x a :MalignantNeoplasm. }
UNION
:
{ ?x rdf:type :NSCLC). }
UNION
{ ?x rdf:type :SCLC). } }
```

- Requête : toutes les instances de la classe Neoplasm:

```
SELECT ?x WHERE { ?x a :Neoplasm}. }
```

- Traduction étape par étape (en théorie) :

```
SELECT Concat(:db1/neoplasm/, TBL.PATIENT.id) AS ?x  
FROM TBL.PATIENT
```


Contraintes imposées par l'ontologie

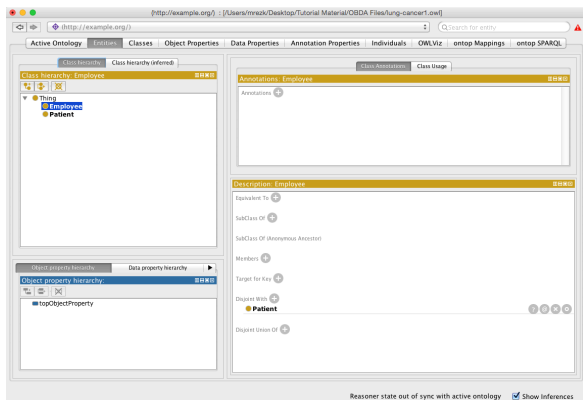
Sources d'incohérence (violation)

- Mappings
- Données de la base

Disjonction entre classes ou propriétés

- L'intersection entre deux classes/propriétés doit être vide.
- Ex : les classes Patient et Employee peuvent être disjointes (*choix de modélisation discutable*)

Cohérence : Définition d'une contrainte Optique

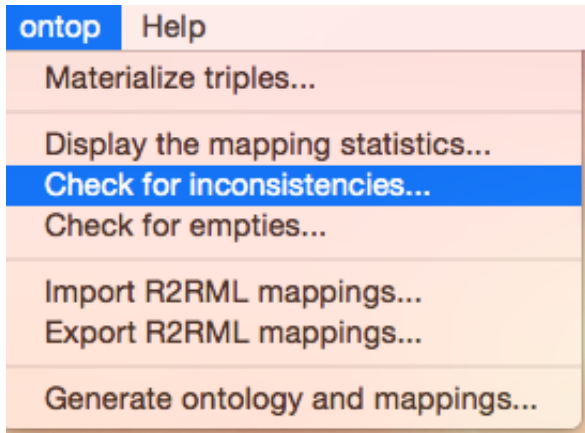


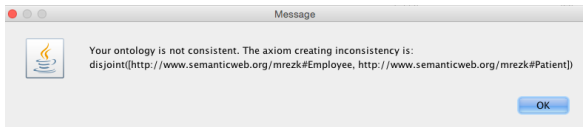
Incohérence : mauvais mapping

The screenshot shows a window titled "Edit Mapping" with three main sections:

- Mapping ID:** A text field containing "map-patient(0)".
- Target (Triples Template):** A text area containing the template `:db1/{PATIENTID} a :Employee .`, where `:Employee` is highlighted in orange.
- Source (SQL Query):** A text area containing the SQL query `select * from "tbl_patient"`.

At the bottom of the window, there is a "Test SQL Query" button with a gear icon, and two buttons: "Update" (with a green checkmark icon) and "Cancel" (with a red X icon).





T_Name

PId	Nombre
1	Anna
2	Mike

Information distribuée dans plusieurs tables. Pas de fédération SQL pour le moment.

T_NSCLC

Id	hosp	Stge
1	X	two
2	Y	one

Information codée différemment. Par exemple, le niveau d'avancement est sous forme textuelle

T_SCLC

key	hosp	St
1	XXX	
2	YYY	

Mapping ID:

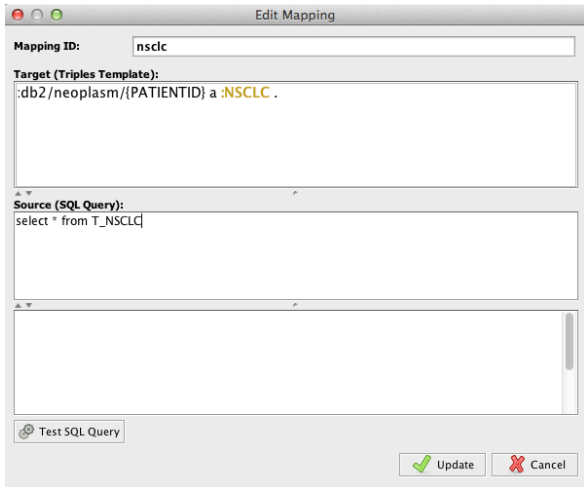
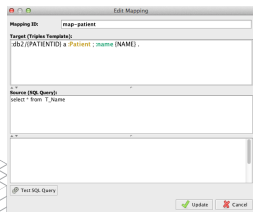
Target (Triples Template):
:db2/{PATIENTID} a :Patient ; :name {NAME} .

Source (SQL Query):
select * from T_Name

Test SQL Query

Update Cancel

Nouveaux mappings



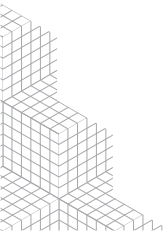
The screenshot shows the 'Edit Mapping' window for a mapping named 'map-patient'. It features three main sections: 'Mapping ID' with the value 'map-patient', 'Target (Triples Template)' containing the query ':db2/(PATIENTID) a :Patient ; :name (NAME) .', and 'Source (SQL Query)' containing 'select * from T_Name'. At the bottom, there is a 'Test SQL Query' button and 'Update' and 'Cancel' buttons.

The screenshot shows the 'Edit Mapping' window for a mapping named 'nscic'. It features three main sections: 'Mapping ID' with the value 'nscic', 'Target (Triples Template)' containing the query ':db2/neoplasm/(PATIENTID) a :NSCLC .', and 'Source (SQL Query)' containing 'select * from T_NSCLC'. At the bottom, there is a 'Test SQL Query' button and 'Update' and 'Cancel' buttons.

- Les patrons d'URI différente suivant la source (db2 ou db1)

<http://www.exareme.org>

- Développé à l'Université d'Athènes
- Module de traitement de requête SQL de la plateforme Optique
- Logiciel libre



Étapes :

- Installer Exareme
 - Pré-requis : Python 2.7 et APSW.
- Construire l'URL JDBC pour Ontop, qui est composée de :
 - 1 fragment pour le serveur où Exareme est installé
 - 1 fragment pour chaque base de données DB

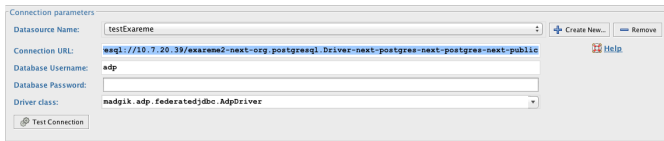
Étapes :

- Installer Exareme
 - Pré-requis : Python 2.7 et APSW.
- Construire l'URL JDBC pour Ontop, qui est composée de :
 - 1 fragment pour le serveur où Exareme est installé
 - 1 fragment pour chaque base de données DB

```
jdbc:fedadp:http://10.7.20.80:9090/tmp-  
fedDB-data1-next-jdbc:postgresql://10.7.20.80/exareme1-next-  
org.postgresql.Driver-next-postgres-next-postgres-next-public-  
fedDB-data2-next-jdbc:postgresql://10.7.20.39/exareme2-next-  
org.postgresql.Driver-next-postgres-next-postgres-next-public
```

Étapes :

- Installer Exareme
 - Pré-requis : Python 2.7 et APSW.
- Construire l'URL JDBC pour Ontop, qui est composée de :
 - 1 fragment pour le serveur où Exareme est installé
 - 1 fragment pour chaque base de données DB



Connection parameters

Datasource Name: testExareme + Create New... - Remove

Connection URL: jdbc://10.7.20.39/exareme2-next-org.postgresql.driver-next-postgres-next-postgres-next-public ? Help

Database Username: adp

Database Password:

Driver class: madgik.adp.federatedjdbc.AdpDriver

Test Connection

Vous pouvez ensuite écrire vos mappings et requêter comme d'ordinaire.

Quelques liens

<http://ontop.inf.unibz.it>

<http://www.optique-project.eu>

<https://twitter.com/ontop4obda>

