

The Semantic Web Technology Stack (not a piece of cake...)

Most apps use only a subset of the stack

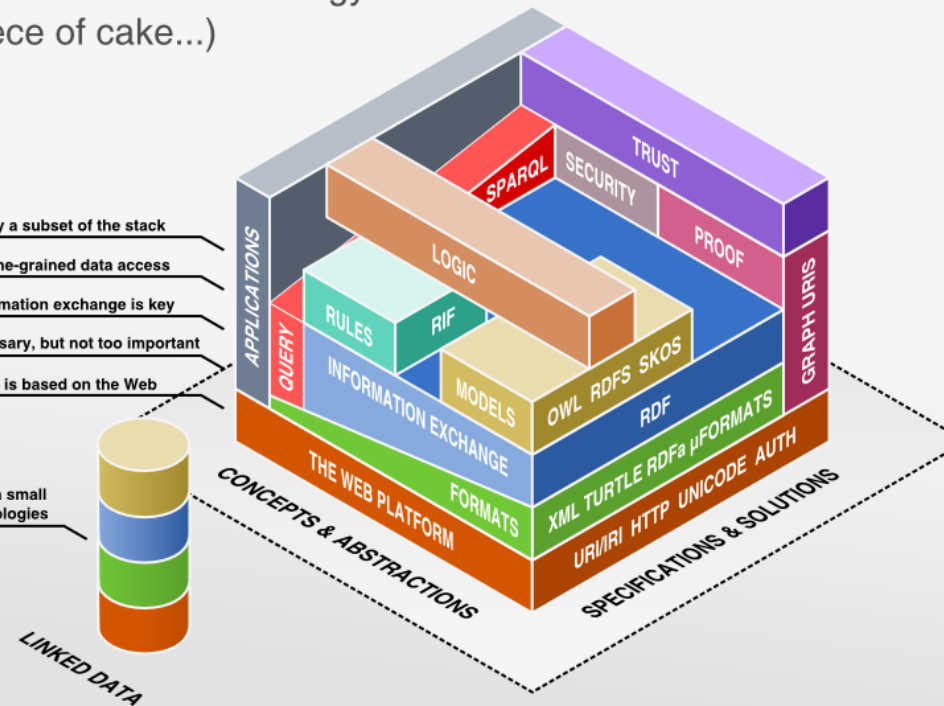
Querying allows fine-grained data access

Standardized information exchange is key

Formats are necessary, but not too important

The Semantic Web is based on the Web

Linked Data uses a small
selection of technologies



Partie 3 : Interroger le Web

Représenter pour interroger

Représentation des données

- **Modèle** pour représenter les données distribuées
- **Triplets** : modèle simple pour nommer des liens entre choses identifiées par URIs
- Sans moyen d'interrogation ou d'utilisation, décrire et stocker des données ne sert ... à rien.

Interrogation des données avec un standard : SPARQL

- **SPARQL Protocol And RDF Query Language**
- Le motif des requêtes suit la syntaxe Turtle (une des syntaxes concrètes possibles pour RDF)
- Une requête est un graphe avec des variables libres ; chaque réponse est un appariement de ce graphe dans les données

Versions et composants

SPARQL 1.0 (2008) et SPARQL 1.1 (2013)

- Dans ce cours, 1.0 en majorité, 1.1 pour les nouveautés
- Des liens avec SQL et XPATH vus en 3IF

Mais surtout

- Un langage de requêtes <http://www.w3.org/TR/rdf-sparql-query/>
- Un protocole <http://www.w3.org/TR/rdf-sparql-protocol/> pour soumettre une requête à un serveur distant et récupérer la réponse
- Un format de résultat (JSON, XML, CSV, etc.)

JSON <http://www.w3.org/TR/rdf-sparql-json-res/>

XML <http://www.w3.org/TR/rdf-sparql-XMLres/>

Plan du cours

Vers un accès à toutes les données du Web !

- 1/ Écrire des requêtes c'est appareiller des graphes
- 2/ Éléments de syntaxe de base pour les requêtes SELECT
- 3/ Pré- et post- traitements de requêtes
- 4/ Autres types de requêtes
- 5/ Points d'accès, protocoles et réponses
- 6/ Epilogue

1. Intuitions

Morphologie d'une requête

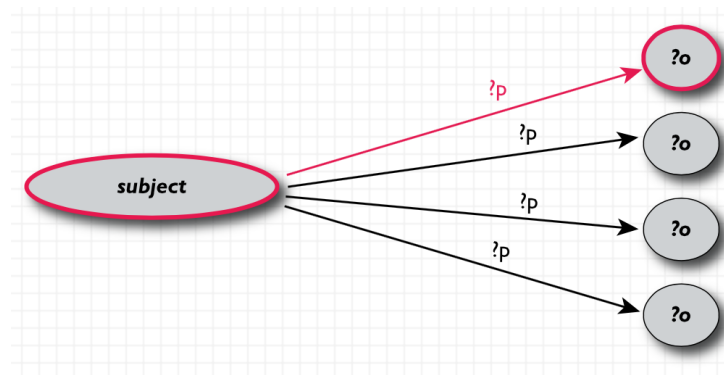
Les données sont triplets : on cherche des triplet

- ***Dis moi tout***

```
SELECT * WHERE {  
  ?s ?p ?o .  
}
```

- ***Dis moi tout au sujet de l'INSA***

```
SELECT * WHERE {  
  <http://dbpedia.org/resource/  
  Institut_national_des_sciences_Appliqu%C3%A9es_de_Lyon>  
  ?p ?o .  
}
```

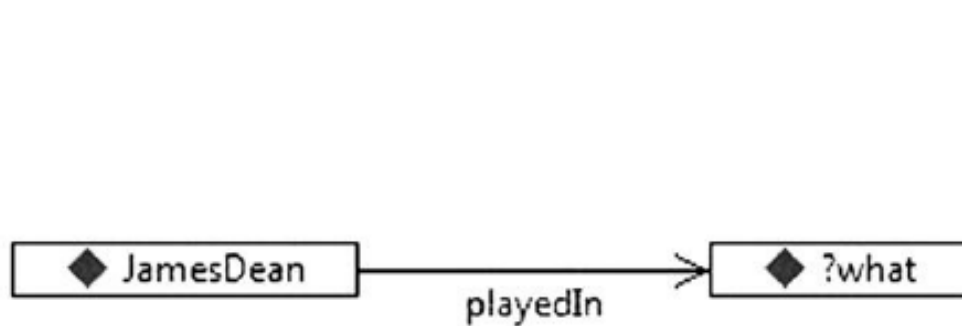


Des appariements de graphes

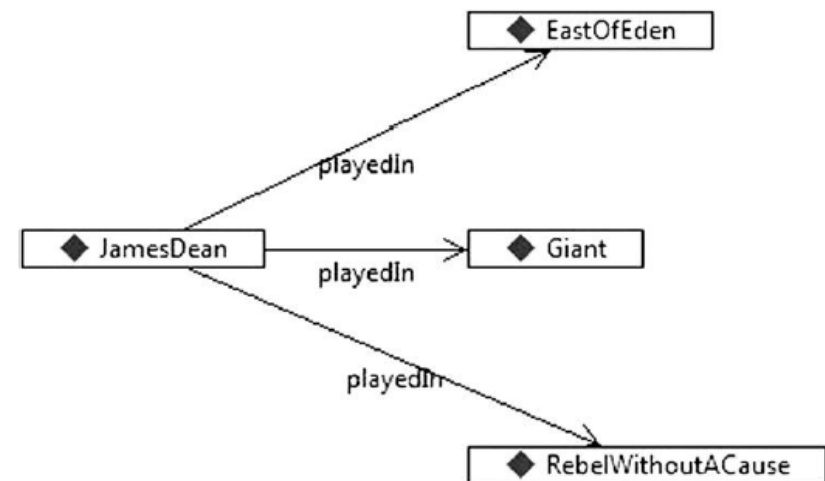
Ecrire une requête, c'est écrire un graphe requête

- on recherche ses occurrences dans un graphe cible (les données)
- appariement de graphes (graph pattern matching) : on cherche tous les sous-graphes qui correspondent au patron du graphe donné par la requête

```
Ask: SELECT ?what WHERE { :JamesDean :playedIn ?what }  
Answer: :Giant, :EastOfEden, :RebelWithoutaCause.
```



Graphe requête



Graphe données

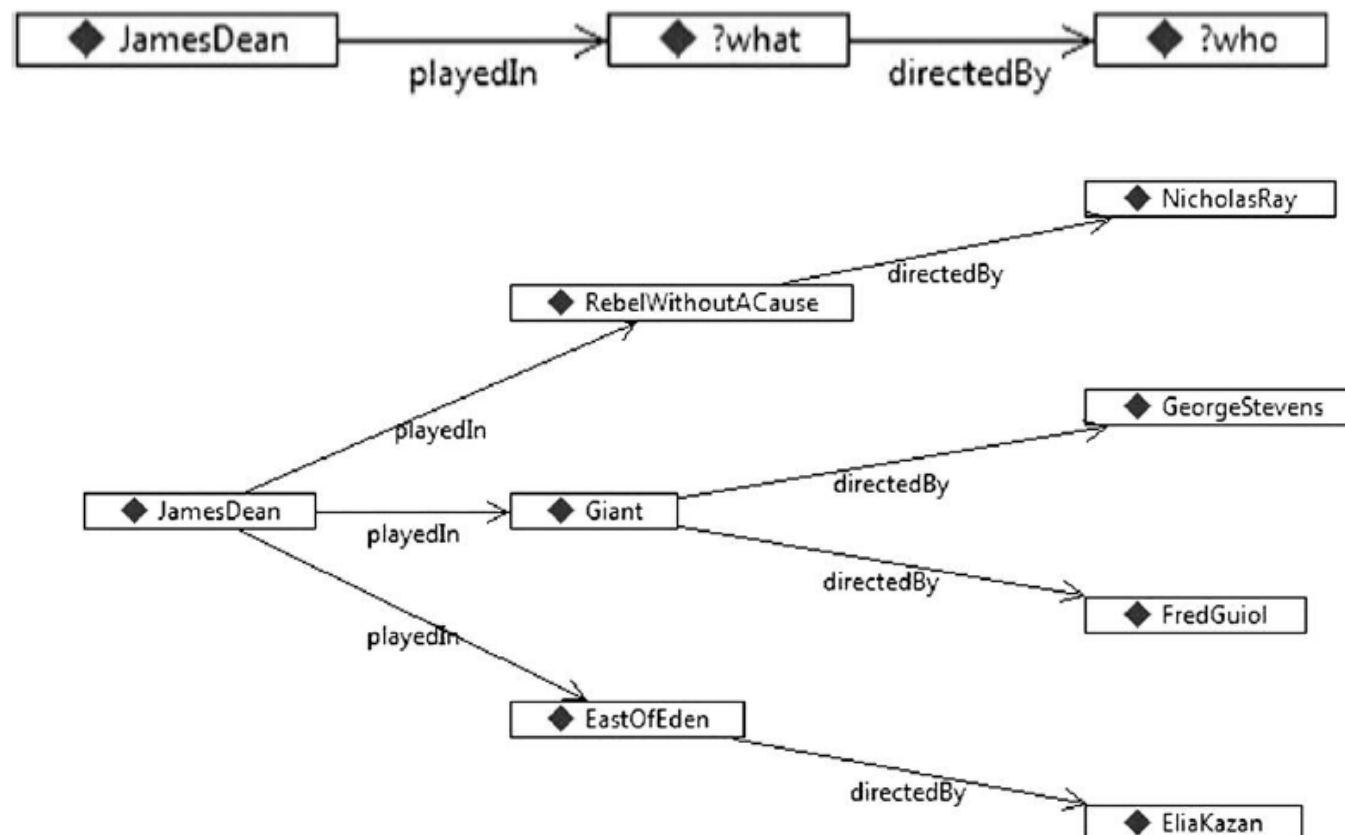
Requête « chemin »

Ask:

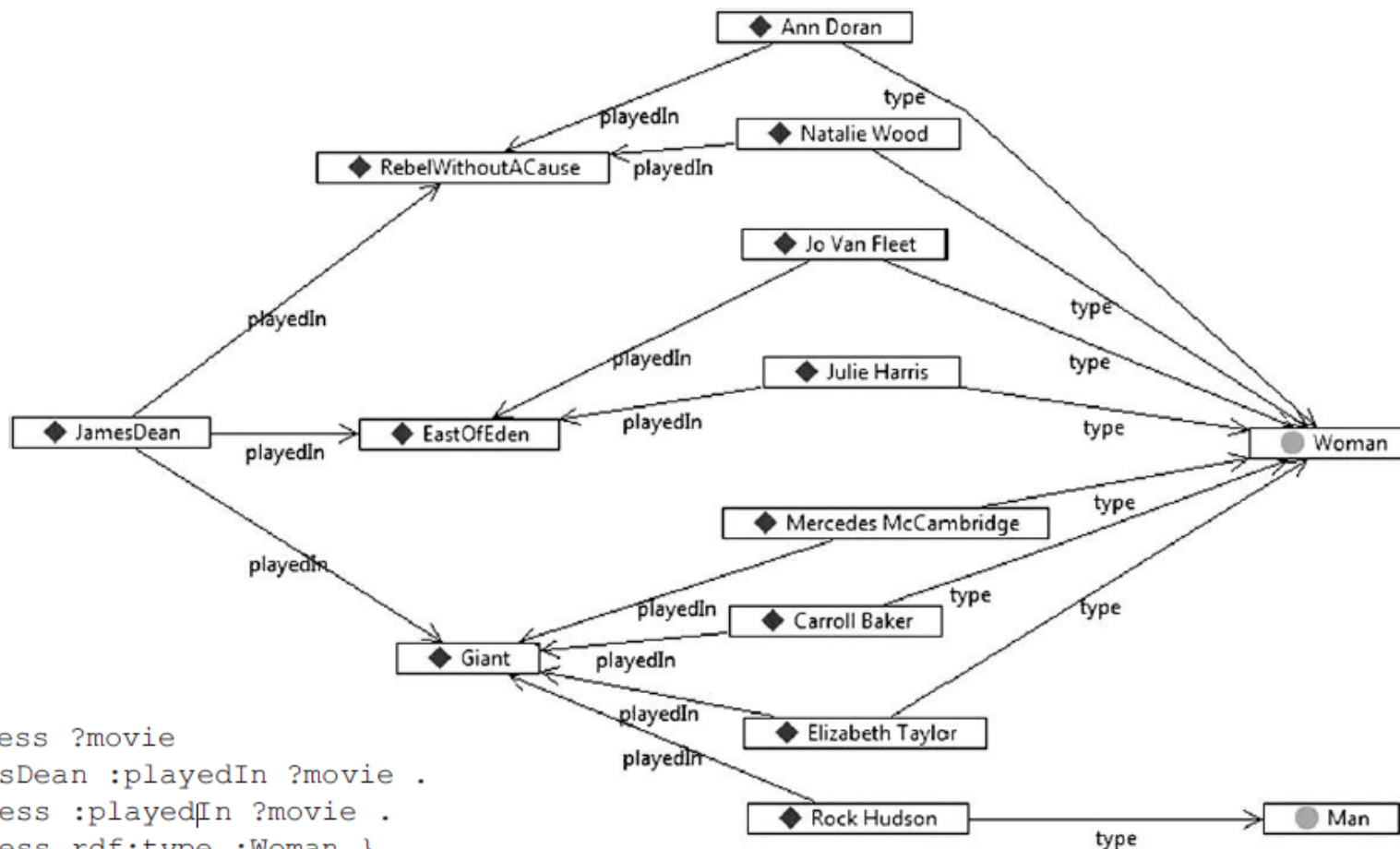
```
SELECT ?who
WHERE { :JamesDean :playedIn ?what .
        ?what :directedBy ?who . }
```

Answer:

:GeorgeStevens, :EliaKazan, :NicholasRay, :FredGuiol

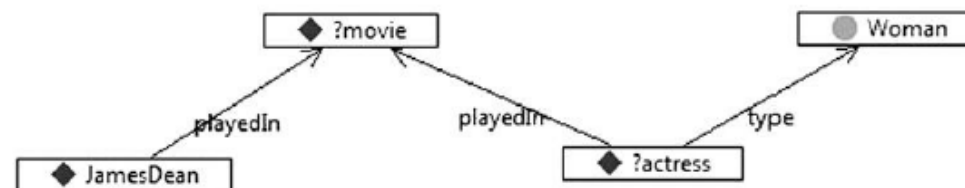


Vers la requête « graphe »



```

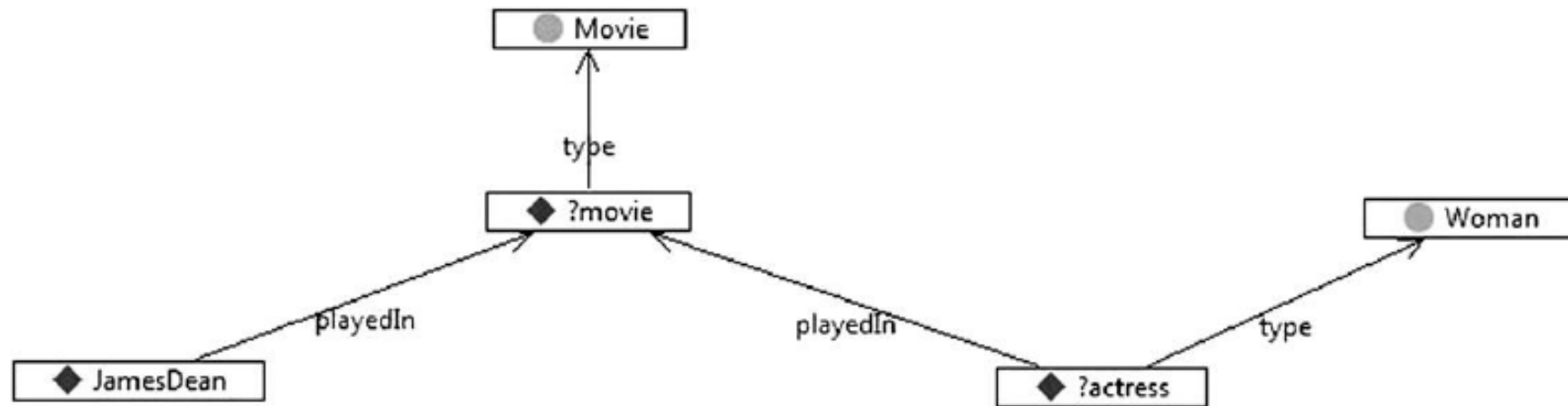
SELECT ?actress ?movie
WHERE {
  :JamesDean :playedIn ?movie .
  ?actress :playedIn ?movie .
  ?actress rdf:type :Woman }
  
```



Requête « graphe »

Le nom de variable ?movie est arbitraire. Il faut ?movie rdf:type :Movie .

```
SELECT ?director
WHERE { :JamesDean :playedIn ?movie .
       ?movie :director ?director . }
```



Des appariements de graphes

Autant de résultats que d'appariements possibles

```
SELECT ?isbn ?price ?currency
```

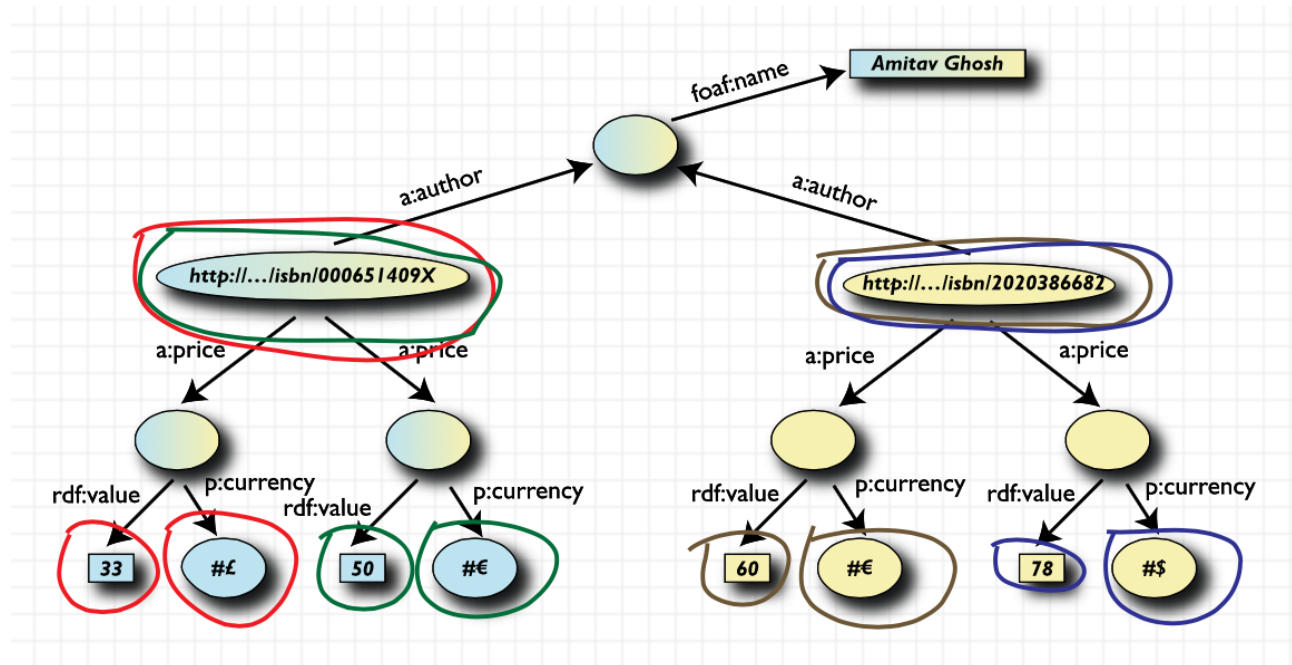
```
WHERE {
```

```
  ?isbn a:price ?x.
```

```
  ?x rdf:value ?price.
```

```
  ?x p:currency ?currency.
```

```
}
```



Résultat :

```
[[<..49X>,33,£], [<..49X>,50,€],  
 [<..6682>,60,€], [<..6682>,78,$]]
```

2. Éléments de base

Éléments de langage

- **Variables** ?x
- **Motif/patron de graphe** : une conjonction de triplets avec « . »

```
SELECT ?p WHERE {  
  ?p rdf:type foaf:Person .  
  ?p foaf:name ?name .  
}
```
- **a est équivalent à** `rdf:type`
- **Factorisation pour un même sujet** « ; »
- **Factorisation pour un même couple (propriété, valeurs)** avec « , »

```
SELECT * where {  
  ?p a foaf:Person ; foaf:name ?n .  
}
```

Éléments de langage

- **Utilisation de ressources anonymes**

```
SELECT * where { [] a foaf:Person ; foaf:name ?n . }
```

- **Déclarer et utiliser le préfixe d' un espace de nom**

```
PREFIX nsdbpedia: <http://dbpedia.org/resource/>
```

```
SELECT * where { ?s ?o nsdbpedia:Barack_Obama . }
```

- **Déclarer et utiliser un URI de base**

```
BASE <http://dbpedia.org/resource/>
```

```
SELECT * where{ ?s ?o <Barack_Obama> . }
```

- **Spécifier le type des littéraux**

```
SELECT ?s, ?p where
```

```
{ ?s [] "Frankreich"@de . ?s ?p "50"^^xsd:integer . }
```

Éléments de langage

- Un motif optionnel est une partie de graphe non obligatoire, p.e., on autorise le fait qu'une variable ne soit pas liée.

```
SELECT distinct ?p, ?homepage WHERE {  
  ?p ?s <http://dbpedia.org/resource/France> .  
  ?p a foaf:Person .  
  OPTIONAL { ?p foaf:homepage ?homepage .}  
}
```

Éléments de langage

Nous disposons du ET, mais qu'en est-il du OU ?

- **UNION** combine deux motifs graphes
- Les variables de chaque motif sont indépendantes, mais les résultats sont combinés
- **Exemple** : Les « personnes » nées à Brooklyn ou de prénom anglais John

```
SELECT distinct ?p where
```

```
{
```

```
  { ?p dbo:birthPlace dbr:Brooklyn . }
```

```
UNION
```

```
  { ?p foaf:givenName "John"@en . }
```

```
}
```


Éléments de langage

Soustraire un motif des résultats

- Supprimer des résultats correspondants à un pattern
- **Exemple** : *Les endroits peuplés de France sauf ceux du Bas-Rhin*

```
SELECT distinct ?p where
```

```
{
```

```
  ?p a dbo:PopulatedPlace .
```

```
  ?p dbo:country dbr:France .
```

```
  ?p dbo:department [] . # est-ce important ?
```

```
MINUS
```

```
{ ?p dbo:department dbr:Bas-Rhin . }
```

```
}
```

Éléments de langage

Requêtes transitives

- Répétition +?
 - Répétition bornée {n,m}
 - Optionnel ?
 - Séquence /

```
SELECT ?type WHERE {
```
 - Alternative |

```
{
```
 - inverse : ^

```
SELECT * WHERE
```
 - négation : !

```
{ ?x rdfs:subClassOf+ ?type .}
```
- ```
}
```
- ```
FILTER (?x = dbo:Hospital)
```
- ```
}
```

# Éléments de langage

**Négation : si quelque chose n'est pas dit, on ne peut dire qu'il est faux**

- Un mot-clé : UNSAID qui introduit un sous-graphe {...}

*Tous les acteurs vivants qui ont joué dans :Giant*

```
SELECT ?actor
WHERE {?actor :playedIn :Giant .
 UNSAID {?actor :diedOn ?deathdate .}
}
```

- *Les acteurs qui ne sont pas des producteurs cad les acteurs pour lesquels il n'y a pas de données dans le jeu considéré qui affirme qu'ils sont aussi producteurs*

```
SELECT ?actor
WHERE {?actor a :Actor .
 UNSAID {?actor a :Producer}
}
```

# Aspects déclaratifs et procéduraux

La clause **WHERE** fournit un nombre de contraintes sous la forme de triplets avec variables, URI et littéraux :

- Les contraintes sont listées dans un ordre et évaluées dans celui-ci (cela peut dépendre des implémentations)
- Peut importe l'ordre, les résultats sont les mêmes : une conjonction !

```
SELECT ?q3
WHERE { :JamesDean :playedIn ?q1 .
 ?q3 :playedIn ?q1 .
 ?q3 :playedIn ?q2 .
 ?q2 :directedBy :JohnFord . }
```

```
SELECT ?q3
WHERE { ?q3 :playedIn ?q1 .
 :JamesDean :playedIn ?q1 .
 ?q3 :playedIn ?q2 .
 ?q2 :directedBy :John Ford. }
```

# Aspects déclaratifs et procéduraux

```
SELECT ?q3
WHERE { :JamesDean :playedIn ?q1 .
 ?q3 :playedIn ?q1 .
 ?q3 :playedIn ?q2 .
 ?q2 :directedBy :JohnFord . }
```

```
SELECT ?q3
WHERE { ?q3 :playedIn ?q1 .
 :JamesDean :playedIn ?q1 .
 ?q3 :playedIn ?q2 .
 ?q2 :directedBy :John Ford. }
```

La plupart des moteurs ont une approche en « *top-down* » : une règle consiste à contraindre au plus tôt l'ensemble de solutions

- On veille à ordonner les triplets d'une requête de sorte à introduire le moins de variables possibles à chaque triplets
- Par exemple, dans ces deux cas, une seule variable est introduite

```
:JamesDean :playedIn ?q1 .
?q3 :playedIn ?q1 .
```

```
?q2 :directedBy :JohnFord.
?q3 :playedIn ?q2 .
```

## 3. Pré- et Post-traitements

# Filtres

## Fonctions de test XPATH 2.0 et d'autres

- **Post-traitement : contraintes en fin de la clause WHERE**

- **Exemple :** *Les présidents français nés après 1950*

```
SELECT * WHERE {
 ?p dbp:title dbr:President_of_France.
 ?p dbo:birthDate ?d .
 FILTER (?d >= "1950-01-01"^^xsd:date)
}
```

- **Comparateurs** < <= = > >= = !=

- **Expressions régulières :** `select * where {?p dbp:name ?n . FILTER ( regex(?n, ".*Jacques.*") )}`

- **Tests et casts** `isURI(?x) isLiteral(?y) isBlank(?z) bound(?t) lang(?x)='en' datatype(?y)=xsd:string xsd:integer(?x)`

23

# Filtres

## D'autres fonctions

- **Sur les dates** YEAR() NOW() ...
- **Sur les chaînes** CONCAT() CONTAINS() ...
- **Sur les nombres** ABS() FLOOR() ROUND() RAND() ...
- **Autres** MD5() URI() IRI() ...

## Combinaison de tests

- **Connecteurs** &&, ||, !, ()
- **Branchements conditionnels IF/THEN/ELSE**  
`FILTER(if(langMatches(lang(?name),"FR"),?age>=18, ?age>=21))`
- **Appartenance à une liste** `FILTER (?n IN ("N", "S", "E","O"))`  
`FILTER (?n NOT IN ("N", "S", "E","O"))`
- **Absence d'un motif** `FILTER NOT EXISTS { ?x foaf:age -1 }`



# Sélectionner le graphe source

## Un graphe source par défaut sur un access point

- On veut intégrer un (ou plus) graphe de données RDF publié par nos soins ou une tierce personne

- ```
SELECT *  
FROM <http://ns.inria.fr/fabien.gandon/foaf.rdf>  
WHERE { ?x ?y ?z . }
```
- ```
SELECT *
FROM NAMED <http://ns.inria.fr/fabien.gandon/foaf.rdf>
FROM NAMED <http://ns.inria.fr/zizou/foaf.rdf>
WHERE { GRAPH ?g { ?x ?y ?z . } }
```

# Trier, compter, limiter les résultats

## Très similaire à SQL

- `SELECT ?name`  
`WHERE { ?x foaf:name ?name . }`  
`ORDER BY ?name`  
`LIMIT 20`  
`OFFSET 20`
- `ORDER BY DESC`
- `ORDER BY ASC`
- `SELECT (COUNT (?movie) AS ?howmany)`  
`WHERE { :JamesDean ?playedIn ?movie . }`

# Groupes et agrégats (SPARQL 1.1)

| Company | Amount | Year |
|---------|--------|------|
| ACME    | \$1250 | 2010 |
| PRIME   | \$3000 | 2009 |
| ABC     | \$2500 | 2009 |
| ABC     | \$2800 | 2010 |
| PRIME   | \$1950 | 2010 |
| ACME    | \$2500 | 2009 |
| ACME    | \$3100 | 2010 |
| ABC     | \$1500 | 2009 |
| ACME    | \$1250 | 2009 |
| PRIME   | \$2350 | 2009 |
| PRIME   | \$1850 | 2010 |

```
:row1 a :Sale .
:row1 :company :ACME .
:row1 :amount 1250 .
:row1 :year 2010 .
```

```
SELECT (SUM (?val) AS ?total)
WHERE {?s a :Sale .
?s :amount ?val }
```

| ?total   |
|----------|
| 24050.00 |

```
SELECT ?year (SUM (?val) AS ?total)
WHERE {?s a :Sale .
?s :amount ?val .
?s :year ?year }
GROUP BY ?year
```

| ?year | ?total   |
|-------|----------|
| 2009  | 13100.00 |
| 2010  | 10950.00 |

# Groupes et agrégats (SPARQL 1.1)

```
SELECT ?year ?company (SUM (?val) AS ?total)
WHERE {
 ?s a :Sale .
 ?s :amount ?val .
 ?s :year ?year .
 ?s :company ?company .
}
GROUP BY ?year ?company
```

| ?year | ?company | ?total  |
|-------|----------|---------|
| 2009  | ACME     | 3750.00 |
| 2009  | ABC      | 4000.00 |
| 2009  | PRIME    | 5350.00 |
| 2010  | ACME     | 4350.00 |
| 2010  | PRIME    | 3800.00 |
| 2010  | ABC      | 2800.00 |

| Company | Amount | Year |
|---------|--------|------|
| ACME    | \$1250 | 2010 |
| PRIME   | \$3000 | 2009 |
| ABC     | \$2500 | 2009 |
| ABC     | \$2800 | 2010 |
| PRIME   | \$1950 | 2010 |
| ACME    | \$2500 | 2009 |
| ACME    | \$3100 | 2010 |
| ABC     | \$1500 | 2009 |
| ACME    | \$1250 | 2009 |
| PRIME   | \$2350 | 2009 |
| PRIME   | \$1850 | 2010 |

# Groupes : la clause HAVING

```
SELECT ?year ?company (SUM (?val) AS ?total)
WHERE {
 ?s a :Sale .
 ?s :amount ?val .
 ?s :year ?year .
 ?s :company ?company .
}
```

```
GROUP BY ?year ?company
HAVING (?total > 5000)
```

| ?year | ?company | ?total  |
|-------|----------|---------|
| 2009  | PRIME    | 5350.00 |

| Company | Amount | Year |
|---------|--------|------|
| ACME    | \$1250 | 2010 |
| PRIME   | \$3000 | 2009 |
| ABC     | \$2500 | 2009 |
| ABC     | \$2800 | 2010 |
| PRIME   | \$1950 | 2010 |
| ACME    | \$2500 | 2009 |
| ACME    | \$3100 | 2010 |
| ABC     | \$1500 | 2009 |
| ACME    | \$1250 | 2009 |
| PRIME   | \$2350 | 2009 |
| PRIME   | \$1850 | 2010 |

## HAVING n'est pas FILTER

**FILTER** concerne les variables liées à un graph-pattern en particulier et apparaît entre {...}

**HAVING** concerne les variables définies par des agrégations dans la clause SELECT et apparaît en dehors d'un graph-pattern

## 4. Divers types de requête

# Requête SPARQL classique

- **PREFIX** espaces de noms  
**SELECT** selection des variables,  
préfixées ou ?
- [BASE]
- [FROM]
- **WHERE** contraintes à respecter :  
le graphe recherché (motifs)  
les filtres
- **OFFSET, LIMIT, ORDER BY**  
traitements sur les résultats

```
PREFIX p1: <...>
PREFIX p2: <...>
...
SELECT ...
WHERE {
 ...
}
OFFSET ...
LIMIT ...
ORDER BY ...
```

# Vérifier l'existence d'une réponse

## Décrire une ressource

### Afin d'éviter d'énumérer tous les résultats

- `ASK WHERE { :ElizabethTaylor :diedOn ?any }`
- `ASK WHERE { UNSAID { :ElizabethTaylor :diedOn ?any } }`
- `ASK WHERE { ?any :playedIn :Giant.  
?any :bornOn ?birthday .  
FILTER ( ?birthday > "1950-01-01"^^xsd:date ) }`

### Pour en savoir plus au sujet d'une ressource dont on ne sait que peu

- On récupère des données sur les présidents  
`DESCRIBE ?a where { ?a a :President }`



# Construire un graphe comme résultat

## Le résultat d'une requête donné par un graphe

- Définir un « **template** » pour construire le graphe résultat
- **Construire** un graphe RDF en guise de réponse

```
CONSTRUCT { ?etudiant a :AvenirDeLaNation . }
WHERE { ?etudiant a insa:etudiant . }
```

## Quelques raisons ...

- Insérer les nouveaux triplets dans les données originales
- Créer un nouveau jeu de données à publier
- Sérialiser les résultats dans un fichier pour utilisation future
- Autres applications, autre base de données, etc.

# Modifier une base en SPARQL 1.1

## Langage de type CRUD (create, read, update, delete)

- clear
- create
- crop
- copy
- move
- insert
- ...

**PREFIX** foaf:<....>

**DELETE** { ?person foaf:givenName « Medhi » }

**INSERT** { ?person foaf:givenName « Mehdi » }

**WHERE**

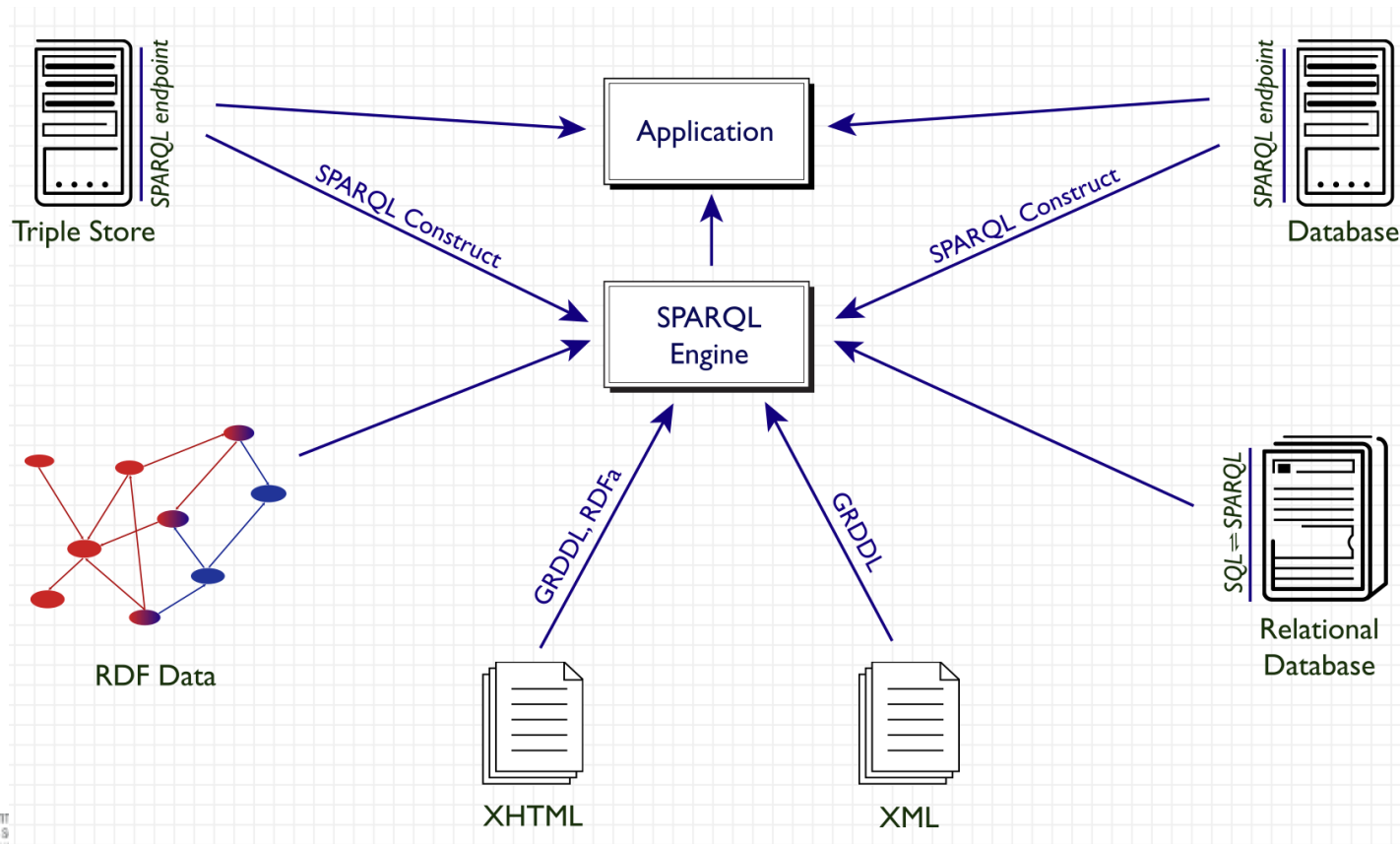
```
{
 ?person a foaf:Person .
 ?person foaf:givenName 'Medhi' .
 ?person foaf:name 'Kaytoue' .
}
```

## 5. Accès et protocoles

# Accès en pratique

## En local ou à distance

- Accéder, parser, interroger un fichier RDF/XML
- Contacter un point d'accès via HTTP/SOAP
- Le retour est un ensemble de *bindings* ou du RDF dans une syntaxe concrète négociée (en retour de CONSTRUCT par exemple)



# Communication HTTP

PREFIX dc: <http://purl.org/dc/elements/1.1/>

SELECT ?book ?who

WHERE { ?book dc:creator ?who }

GET /sparql/?query=*EncodedQuery* HTTP/1.1

Host: www.example

User-agent: my-sparql-client/0.1

HTTP/1.1 200 OK

Date: Fri, 06 May 2005 20:55:12 GMT

Server: Apache/1.3.29 (Unix) PHP/4.3.4 DAV/1.0.3

Connection: close

Content-Type: application/sparql-results+xml

<?xml version="1.0"?>

<sparql xmlns="http://www.w3.org/2005/sparql-results#">

<head>

<variable name="book"/>

<variable name="who"/>

</head>

<results distinct="false" ordered="false">

<result>

<binding name="book"><uri>http://www.example/book/book5</uri></binding>

<binding name="who"><bnode>r29392923r2922</bnode></binding>

</result> ...

# Apache JENA



## Un outil de choix (en java)

- Parser un graphe, ajouter des triplets et le sérialiser
- Comparer des graphes RDF
- Convertir entre différentes syntaxes
- Raisonner en RDFS
- Une implémentation SPARQL
- Une couche BD
- ...

<https://jena.apache.org/>

<http://www.w3.org/wiki/>

```
// create a model
Model model=new ModelMem();
Resource subject=model.createResource("URI_of_Subject")
// 'in' refers to the input file
model.read(new InputStreamReader(in));
StmtIterator iter=model.listStatements(subject,null,null);
while(iter.hasNext()) {
 st = iter.next();
 p = st.getProperty();
 o = st.getObject();
 do_something(p,o);
}
```

## 6. Épilogue

# Avant le TD...



DBpedia.org - Français

Accueil Tutoriels Documentation Agenda Données

## Exemples SPARQL

Pour étudier/modifier ces requêtes vous pouvez soit [accéder directement au serveur SPARQL](#), soit cliquer sur les liens qui afficheront directement le résultat de la requête.

Sommaire [\[masquer\]](#)

- 1 Par thèmes
  - 1.1 À propos de villes
  - 1.2 À propos de Musées
- 2 Par type de requête
  - 2.1 Pages de catégories

### Par thèmes

#### À propos de villes

- [Qu'est ce que l'on sait sur Paris ?](#)

```
select * where {<http://fr.dbpedia.org/resource/Paris> ?r ?p}
```

#### • [Quelles sont les communes d'Île de France ?](#)

```
prefix db-owl: <http://dbpedia.org/ontology/>

select * where {
 ?ville db-owl:region <http://fr.dbpedia.org/resource/Île-de-France> .
 ?ville rdf:type db-owl:Settlement
}
```

## Découvrir, naviguer à travers le nuage

- <http://fr.dbpedia.org/sparql>
- <http://xmlns.com/foaf/spec/>

### Virtuoso SPARQL Query Editor

Default Data Set Name (Graph IRI)

#### Query Text

```
select distinct * where {?s foaf:name ?l. ?s rdf:type <http://schema.org/Book>} LIMIT 100
```

(Security restrictions of this server do not allow you to retrieve remote RDF data, see [details](#).)

Results Format:

HTML

Execution timeout:

0

milliseconds (values less than 1000 are ignored)

Options:

☒ Strict checking of void variables

(The result can only be sent back to browser, not saved on the server, see [details](#))

Run Query

Reset

| s                                                                                                                                                     | l                                         |
|-------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------|
| <a href="http://fr.dbpedia.org/resource/Conquête_(roman)">http://fr.dbpedia.org/resource/Conquête_(roman)</a>                                         | "Conquête"@fr                             |
| <a href="http://fr.dbpedia.org/resource/Dune_(roman)">http://fr.dbpedia.org/resource/Dune_(roman)</a>                                                 | "Dune"@fr                                 |
| <a href="http://fr.dbpedia.org/resource/Au_cœur_des_ténèbres">http://fr.dbpedia.org/resource/Au_cœur_des_ténèbres</a>                                 | "Au cœur des ténèbres"@fr                 |
| <a href="http://fr.dbpedia.org/resource/L'Ultime_Commandement">http://fr.dbpedia.org/resource/L'Ultime_Commandement</a>                               | "L'Ultime Commandement"@fr                |
| <a href="http://fr.dbpedia.org/resource/La_Guerre_du_Bacta">http://fr.dbpedia.org/resource/La_Guerre_du_Bacta</a>                                     | "La Guerre du Bacta"@fr                   |
| <a href="http://fr.dbpedia.org/resource/La_Bataille_des_Jedi">http://fr.dbpedia.org/resource/La_Bataille_des_Jedi</a>                                 | "La Bataille des Jedi"@fr                 |
| <a href="http://fr.dbpedia.org/resource/L'Héritier_de_l'Empire">http://fr.dbpedia.org/resource/L'Héritier_de_l'Empire</a>                             | "L'Héritier de l'Empire"@fr               |
| <a href="http://fr.dbpedia.org/resource/Les_Thanatonautes">http://fr.dbpedia.org/resource/Les_Thanatonautes</a>                                       | "Les Thanatonautes"@fr                    |
| <a href="http://fr.dbpedia.org/resource/Les_Rêveries_du_promeneur_solitaire">http://fr.dbpedia.org/resource/Les_Rêveries_du_promeneur_solitaire</a>   | "Les Rêveries du promeneur solitaire"@fr  |
| <a href="http://fr.dbpedia.org/resource/La_Société_du_spectacle_(livre)">http://fr.dbpedia.org/resource/La_Société_du_spectacle_(livre)</a>           | "La Société du spectacle"@fr              |
| <a href="http://fr.dbpedia.org/resource/Shining_l'enfant_lumière">http://fr.dbpedia.org/resource/Shining_l'enfant_lumière</a>                         | "Shining, l'enfant lumière"@fr            |
| <a href="http://fr.dbpedia.org/resource/Un_piège_nommé_Krytos">http://fr.dbpedia.org/resource/Un_piège_nommé_Krytos</a>                               | "Un piège nommé Krytos"@fr                |
| <a href="http://fr.dbpedia.org/resource/Vent_de_trahison">http://fr.dbpedia.org/resource/Vent_de_trahison</a>                                         | "Vent de trahison"@fr                     |
| <a href="http://fr.dbpedia.org/resource/Le_Capitaine_Fracasse">http://fr.dbpedia.org/resource/Le_Capitaine_Fracasse</a>                               | "Le Capitaine Fracasse"@fr                |
| <a href="http://fr.dbpedia.org/resource/Yvain_ou_le_Chevalier_au_lion">http://fr.dbpedia.org/resource/Yvain_ou_le_Chevalier_au_lion</a>               | "Yvain ou le Chevalier au lion"@fr        |
| <a href="http://fr.dbpedia.org/resource/Des_souris_et_des_hommes">http://fr.dbpedia.org/resource/Des_souris_et_des_hommes</a>                         | "Des souris et des hommes"@fr             |
| <a href="http://fr.dbpedia.org/resource/Iliade">http://fr.dbpedia.org/resource/Iliade</a>                                                             | "Iliade"@fr                               |
| <a href="http://fr.dbpedia.org/resource/Odyssée">http://fr.dbpedia.org/resource/Odyssée</a>                                                           | "Odyssée"@fr                              |
| <a href="http://fr.dbpedia.org/resource/L'Ultime_Secret">http://fr.dbpedia.org/resource/L'Ultime_Secret</a>                                           | "L'Ultime Secret"@fr                      |
| <a href="http://fr.dbpedia.org/resource/Eugène_Onéguine">http://fr.dbpedia.org/resource/Eugène_Onéguine</a>                                           | "Eugène Onéguine"@fr                      |
| <a href="http://fr.dbpedia.org/resource/Frankenstein_ou_le_Prométhée_moderne">http://fr.dbpedia.org/resource/Frankenstein_ou_le_Prométhée_moderne</a> | "Frankenstein ou le Prométhée moderne"@fr |
| <a href="http://fr.dbpedia.org/resource/Voyage_au_centre_de_la_Terre">http://fr.dbpedia.org/resource/Voyage_au_centre_de_la_Terre</a>                 | "Voyage au centre de la Terre"@fr         |
| <a href="http://fr.dbpedia.org/resource/Le_Péril_bleu">http://fr.dbpedia.org/resource/Le_Péril_bleu</a>                                               | "Le Péril bleu"@fr                        |

Copyright © 2014 [OpenLink Software](#)

Virtuoso version 06.01.3127 on Linux (x86\_64-unknown-linux-gnu), Single Server Edition



# Des requêtes pour comprendre les schémas

## Web sémantique : données distribuées, données en masse

- Slogan AAA
- Il est impossible de tout savoir, et encore moins de savoir comment les choses sont décrites ('par quoi')
- « Que savez-vous de James Dean ? »
- Pratique pour explorer les jeux avec lesquels nous ne sommes pas familier

Ask:

```
SELECT ?property ?value
WHERE { :JamesDean ?property ?value }
```

Answer:

| ?property  | ?value             |
|------------|--------------------|
| bornOn     | 1931-02-08         |
| diedOn     | 1955-09-30         |
| playedIn   | RebelWithoutaCause |
| playedIn   | EastOfEden         |
| playedIn   | Giant              |
| rdf:type   | Man                |
| rdfs:label | James Dean         |

# Requêtes pour comprendre le schémas

## Autres exemples

Ask:

```
SELECT ?property
WHERE { :JamesDean ?property ?value }
```

Answer:

| ?property  |
|------------|
| bornOn     |
| diedOn     |
| playedIn   |
| playedIn   |
| playedIn   |
| rdf:type   |
| rdfs:label |

Ask:

```
SELECT DISTINCT ?property
WHERE { :JamesDean ?property ?value }
```

Answer:

| ?property  |
|------------|
| bomOn      |
| diedOn     |
| playedIn   |
| rdf:type   |
| rdfs:label |

# Requêtes pour comprendre le schémas

Ask:

```
SELECT DISTINCT ?property
WHERE {?q0 a :Actor .
 ?q0 ?property ?object .}
```

Answer:

| ?property  |
|------------|
| bomOn      |
| diedOn     |
| playedIn   |
| rdf:type   |
| rdfs:label |
| produced   |
| sang       |
| wrote      |

Ask:

```
SELECT DISTINCT ?class
WHERE {?class rdfs:subClassOf :Person}
```

Answer:

| ?class      |
|-------------|
| :Actor      |
| :Actress    |
| :Man        |
| :Woman      |
| :Politician |
| :Producer   |

```
SELECT DISTINCT ?class
WHERE {?q0 a ?class}
```

```
SELECT DISTINCT ?property
WHERE {?q0 ?property ?q1}
```

Au final : **reverse-engineering** du schéma (partiel)  
à partir de requêtes sur les données !

# Biblio-webographie

- Harald Sack. **Semantic Web technologies**. Open course Hasso Plattner Institut (2013) : <https://open.hpi.de/course/semanticweb>
- Gandon F., Faron-Zucker C., Corby O. **Le Web sémantique. Comment lier les données et les schémas sur le Web ?** Dunod eds. 2012.
- Allemang D., Hendler J. **Semantic Web for the working ontologist : Effective modelling in RDFS and OWL**. Morgan Kaufmann eds. 1st edition (2008)
- <http://www.w3.org/People/Ivan/CorePresentations/RDFTutorial/>
- Actes annuels des conférences **WWW, ISWC, ESWC** (voir DBLP)
- [http://www.viseo.net/sites/default/files/viseo-group/rdc\\_cours\\_5.pdf](http://www.viseo.net/sites/default/files/viseo-group/rdc_cours_5.pdf)
- MOOC INRIA Web sémantique par Gandon et al.



***Les recommandations et RFC sont toujours d'excellentes sources !***