# BDBIO - RDF et SPARQL

Fabien Duchateau

*fabien.duchateau [at] univ-lyon1.fr*

Université Claude Bernard Lyon 1

2024 - 2025

https://perso.liris.cnrs.fr/fabien.duchateau/BDBIO/

# Rappels

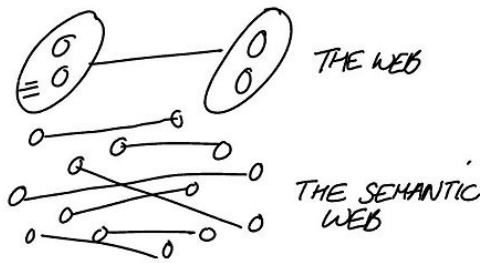
Les données d'application sont principalement gérées par des systèmes de gestion de bases de données (SGBD), qui suivent un modèle de données

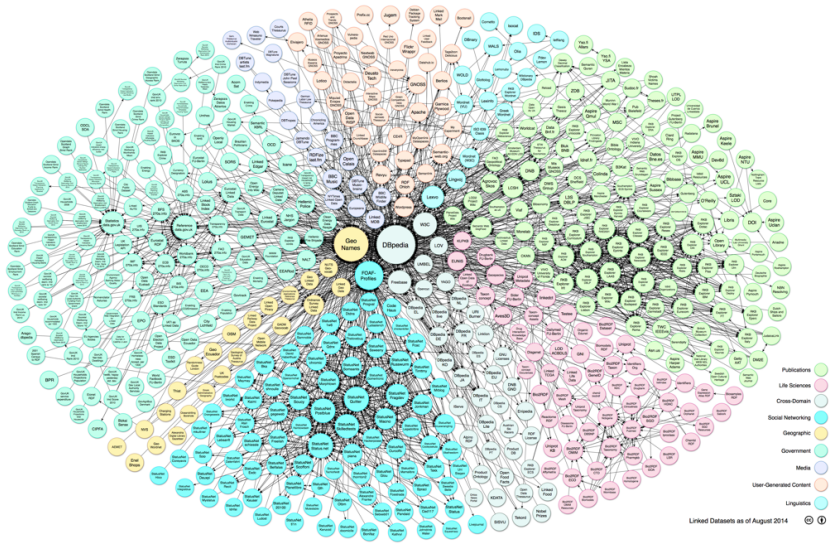
| **Modèle** | **Sérialisation** | **LDD** | **LMD** | **Exemples SGBD** |
|---|---|---|---|---|
| Relationnel | n-uplets | SQL | SQL | SQLite, MariaDB, PostgreSQL |
| Arbre | XML | DTD, XML Schema | XPath, XQuery | XBase, existDB |
| Document | JSON | ? | ? | MongoDB, CouchDB |

Comment apporter du sens aux données du web, notamment pour un traitement automatique par des machines ?

# Le web sémantique

- Des liens sémantiques (relations) entre des ressources (intégration de données)
- Compréhensible par l'humain, exploitable par les machines
- Des millions de sites web contiennent des balises propres au web sémantique (e.g., schema.org)
- Précurseur des graphes de connaissances




---

https://en.wikipedia.org/wiki/Semantic_Web et https://lod-cloud.net/

Hogan *et al.*, Knowledge graphs, ACM Computing Surveys (2021)

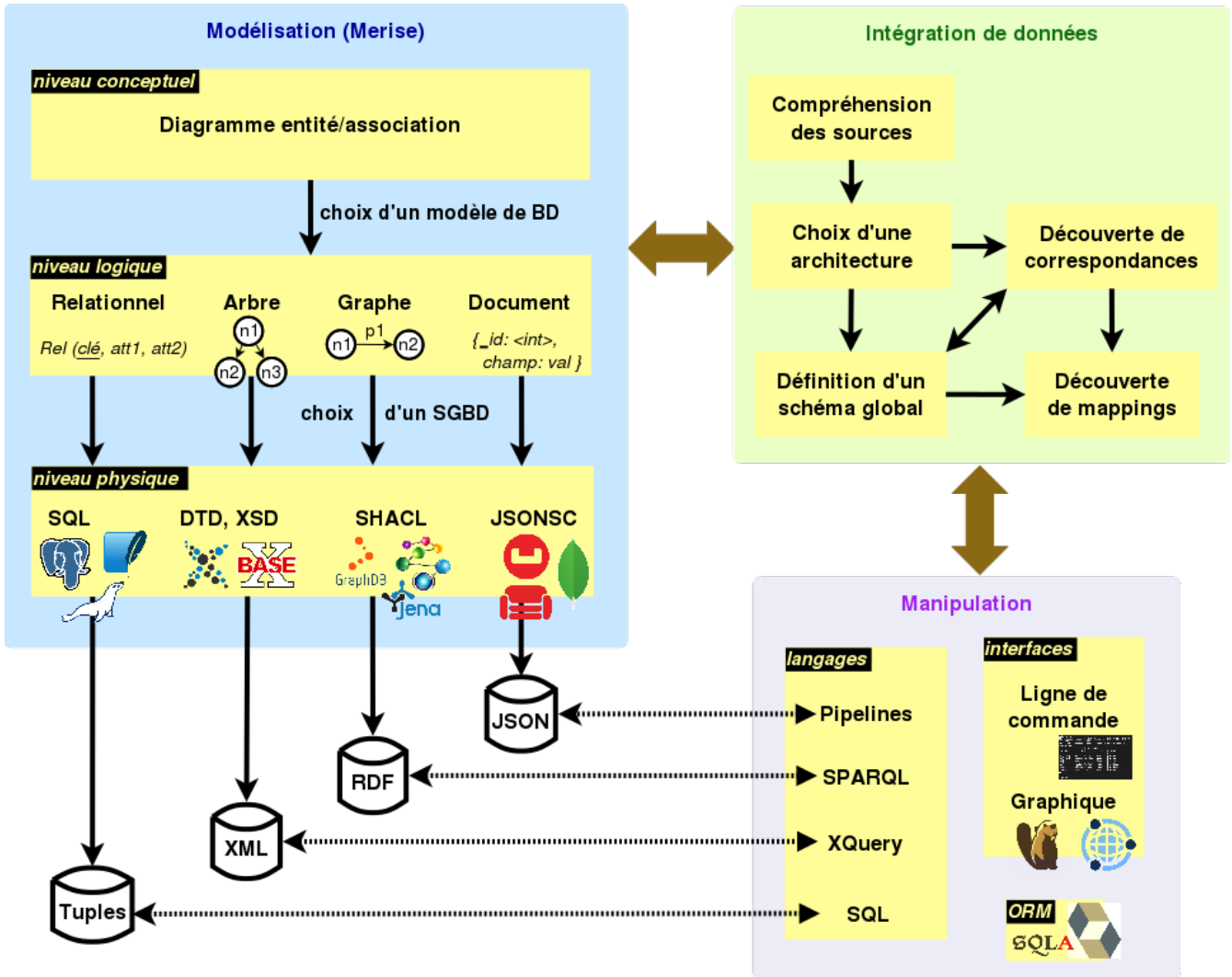
# Quelques concepts du web sémantique

- **RDF**, un modèle de graphe pour décrire et relier des ressources
- **RDFs**, un schéma pour RDF
- **Ontologies** : représentation des connaissances d'un domaine
  - en RDFs, OWL, etc.
- **RDF-store** ou **triplestore**, une famille de SGBD gérant nativement du RDF
- **SPARQL**, un langage d'interrogation pour RDF

---

https://en.wikipedia.org/wiki/Resource_Description_Framework
https://en.wikipedia.org/wiki/RDF_Schema
https://en.wikipedia.org/wiki/Web_Ontology_Language

# Aperçu des notions de BDBIO




Ces diapositives utilisent le genre féminin (e.g., chercheuse, développeuses) plutôt que l'écriture inclusive (moins accessible, moins concise, et pas totalement inclusive)

# Plan

Concepts de RDF

Modélisation en RDF

Manipulation avec SPARQL

# Généralités

RDF, un modèle de données (pour le web) :

- Créé en 1997, standard du W3C en 1999
- Spécifications 1.1 en 2014
- Basé sur la notion de fait (affirmation), sous forme de triplet *sujet - prédicat - objet*
  - Le sujet est une ressource
  - Le prédicat définit une caractéristique du sujet, une relation du sujet avec l'objet
  - L'objet est une ressource qui qualifie la relation

(bd:serie/GenieDesAlpages , bd:inclut, bd:tome/CheptelMaudit)

---

https://www.w3.org/TR/rdf11-concepts/

# Triplets

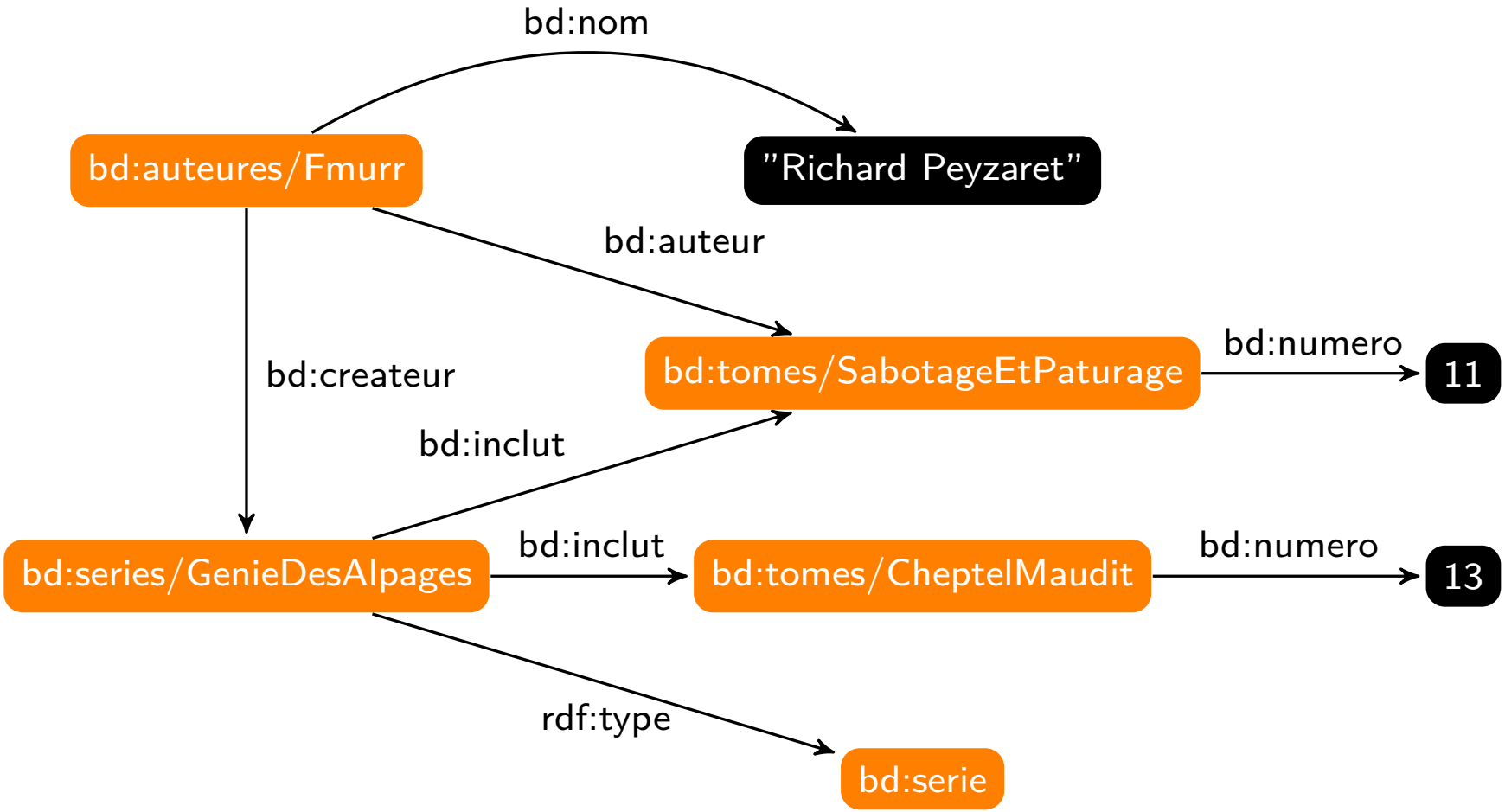
Les données forment des graphes RDF :

- Orientés
- Étiquetés (arêtes et sommets) :
  - soit par une IRI (Resource Identifier), qui peut être symbolique
  - soit par un littéral (valeur), uniquement pour les sommets du graphe
- Une étiquette ne peut être donnée qu'à un seul sommet (i.e., noeud du graphe)
- La même étiquette peut être utilisée sur plusieurs arêtes
- Des sommets anonymes, préfixés par un tiret bas (e.g., _ :anon)

---

https://en.wikipedia.org/wiki/Internationalized_Resource_Identifier

# Exemple de graphe RDF




*Le préfixe `bd` correspond (par exemple) à `http://www.bd.db/`*

# Sérialisations

Différentes sérialisations des données RDF :

- **RDF/XML** (syntaxe originale, mais verbeuse)
- **Notation3** ou **N3** (support de règles)
- **Turtle** (version simplifiée de N3)
- **N-triples** (version basique de Turtle pour les machines)
- **N-quads** (N-triples avec contexte)
- **JSON-LD**, basé sur JSON
- ...

Choix de la sérialisation selon le contexte (e.g., Turtle pour utilisation humaine, RDFa pour produire du HTML, N-triples pour des échanges/dumps)

# Exemple de sérialisation RDF/XML

```
<?xml version="1.0" encoding="UTF-8"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#" xmlns:bd="http://www.bd.db/">
<rdf:Description rdf:about="http://www.bd.db/auteures/Fmurr">
    <bd:nom>Richard Peyzaret</bd:nom>
    <bd:createur rdf:resource="http://www.bd.db/series/GenieDesAlpages"/>
    <bd:auteur rdf:resource="http://www.bd.db/tomes/SabotageEtPaturage"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.bd.db/series/GenieDesAlpages">
    <bd:inclut rdf:resource="http://www.bd.db/tomes/SabotageEtPaturage"/>
    <bd:inclut rdf:resource="http://www.bd.db/tomes/CheptelMaudit"/>
    <rdf:type rdf:resource="http://www.bd.db/serie"/>
</rdf:Description>
<rdf:Description rdf:about="http://www.bd.db/tomes/SabotageEtPaturage">
    <bd:numero rdf:datatype="http://www.w3.org/2001/XMLSchema#int">11</bd:numero>
</rdf:Description>
<rdf:Description rdf:about="http://www.bd.db/tomes/CheptelMaudit">
    <bd:numero rdf:datatype="http://www.w3.org/2001/XMLSchema#int">13</bd:numero>
</rdf:Description>
</rdf:RDF>
```

*Graphe RDF sérialisé (formaté) avec XML*

# Exemple de sérialisation Notation3 (N3)

**Triplet en N3 de la forme :**
sujet prédicat objet .
sujet prédicat objet ; prédicat objet .
sujet prédicat objet , objet ; prédicat objet , objet .

```
@prefix bd: <http://www.bd.db/> .
<http://www.bd.db/auteures/Fmurr>
    bd:nom "Richard Peyzaret" ;
    bd:createur <http://www.bd.db/series/GenieDesAlpages> ;
    bd:auteur <http://www.bd.db/tomes/SabotageEtPaturage> .
<http://www.bd.db/series/GenieDesAlpages>
    bd:inclut <http://www.bd.db/tomes/SabotageEtPaturage> ;
    bd:inclut <http://www.bd.db/tomes/CheptelMaudit> ;
    rdf:type <http://www.bd.db/serie> .
<http://www.bd.db/tomes/SabotageEtPaturage> bd:numero 11 .
<http://www.bd.db/tomes/CheptelMaudit> bd:numero 13 .
```

*Même sérialisation avec Turtle (Terse RDF Triple Language), mais pas avec N-triples qui nécessite un triplet complet sur chaque ligne*

# Plan

Concepts de RDF

Modélisation en RDF

Manipulation avec SPARQL

# Modélisation

Comment représenter le modèle logique pour un graphe ?

- Des méthodologies pour transformer un diagramme entité/association en RDF
- Des dizaines d'outil de conversion d'un schéma relationnel ou arbre XML vers RDF, dont les spécifications R2RML du W3C

**Représentation libre pour le niveau logique**

---

Bagui et al. Mapping rdf and rdf-schema to the entity relationship model, Journal of Emerging Trends in CIS (2014)

Franck et al. A survey of RDB to RDF translation approaches and tools, rapport de recherche (2014)

https://www.w3.org/2001/sw/rdb2rdf/

# Schémas de description

Plusieurs schémas de description de données RDF :

- RDF (vocabulaire de base), pour décrire les ressources
- RDFs (RDF Schema), pour décrire les classes et les prédicats
- ...

Un schéma définit un vocabulaire (classes et propriétés) qui peut être utilisé pour décrire des données ou une ontologie

Ce vocabulaire permet de décrire, mais pas de contraindre !

---

https://www.w3.org/TR/rdf-schema/
https://en.wikipedia.org/wiki/RDF_Schema

# Quelques classes de RDF / RDFS

Rappel : un triplet est de la forme *sujet* - *prédicat* - *objet*

- **rdfs:Resource**, une classe pour déclarer une ressource (tout)
- **rdfs:Class** déclare une ressource de type classe, qui peut être utilisée ensuite par d'autres ressources
- **rdfs:Literal**, une classe pour les littéraux (e.g., entiers, chaines)
- **rdf:Property**, une classe pour les propriétés
- ...

---

https://www.w3.org/TR/rdf-schema/#ch_classes

# Quelques propriétés de RDF / RDFS

- **rdf:type** indique qu'une ressource sujet est une instance d'une classe donnée
- **rdfs:domain** indique la classe du sujet du prédicat pour une propriété donnée
- **rdfs:range** indique la classe de l'objet du prédicat pour une propriété donnée
- **rdfs:subClassOf** pour hiérarchiser les classes
- ...

```
bd:serie rdf:type rdfs:Class .
bd:inclut rdf:type rdf:Property .
bd:inclut rdfs:domain bd:serie .
bd:inclut rdfs:range bd:tome .
```

https://www.w3.org/TR/rdf-schema/#ch_properties

# Propriétés d'utilité de RDFS

- **rdfs:seeAlso** fournit une ressource contenant d'autres informations sur la ressource sujet
- **rdfs:isDefinedBy** indique une ressource qui décrit la ressource sujet (e.g., un vocabulaire)

```
<http://www.bd.db/auteures/Fmurr> rdfs:seeAlso
↪   <http://fr.wikipedia.org/wiki/F'murr> .
<http://www.bd.db/series/GenieDesAlpages> rdfs:isDefinedBy
↪   <http://www.bd.db/> .
```

https://www.w3.org/TR/rdf-schema/#ch_utilvocab

# Exemples de schémas/ontologies utilisant RDFS

- Simple Knowledge Organization System (SKOS), un standard W3C pour décrire un thésaurus
- Dublin Core (DC), qui décrit des ressources du web (e.g., images, vidéos)
- Schema.org, un ensemble de schémas pour structurer les données du web et ses domaines populaires (e.g., personnes, restaurants, e-commerce, événements)
- Friend of a friend (FOAF), une ontologie pour décrire des personnes, des activités et des relations

---

https://en.wikipedia.org/wiki/Simple_Knowledge_Organization_System
https://dublincore.org/
https://schema.org/
https://www.foaf-project.org/

# Exemple de modélisation - version incorrecte

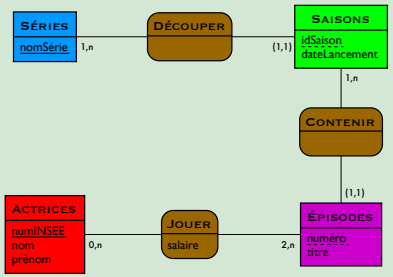
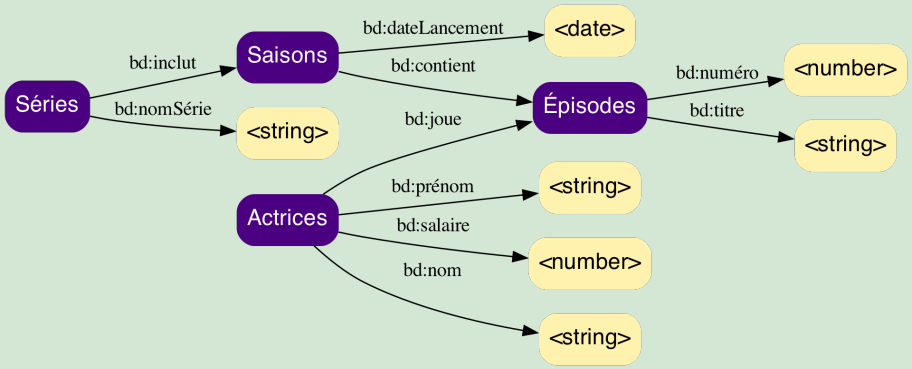



*Diagramme E/A (niveau conceptuel)*




*Représentation libre (niveau logique)*

**Pourquoi cette modélisation logique est incorrecte ?**

# Exemple de modélisation - version incorrecte

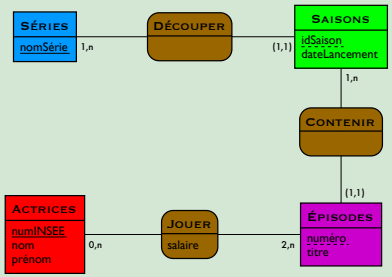
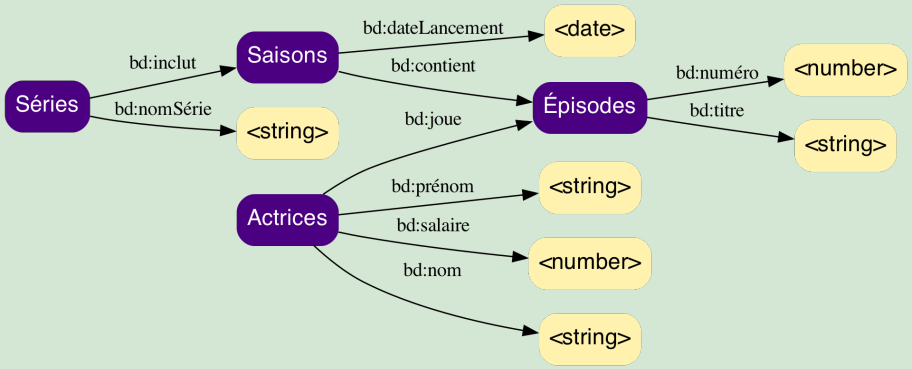


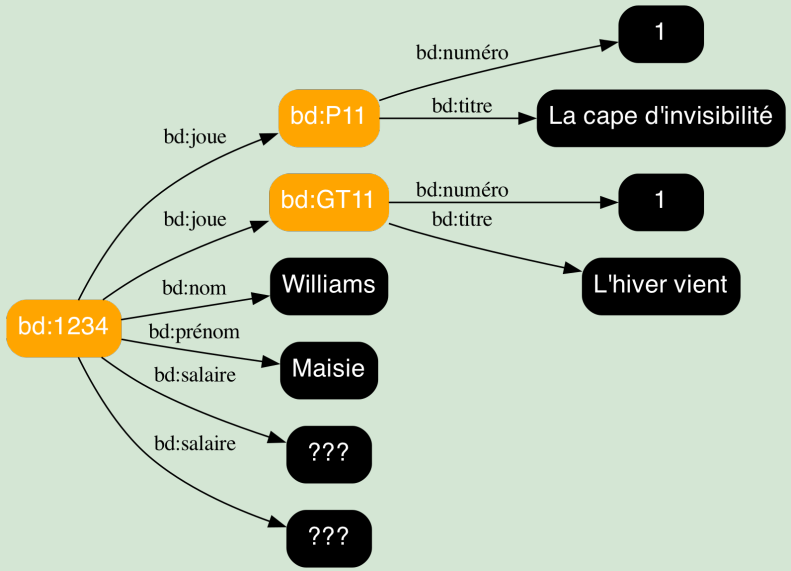

*Diagramme E/A (niveau conceptuel)*




*Représentation libre (niveau logique)*

**Pourquoi cette modélisation logique est incorrecte ?**

*Ci-contre des instances (2 épisodes et 1 actrice)*



# Exemple de modélisation - version correcte

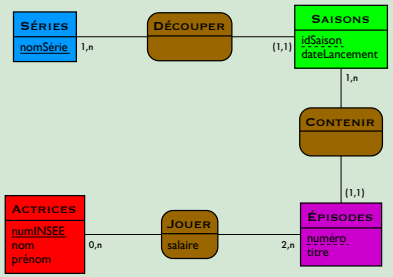
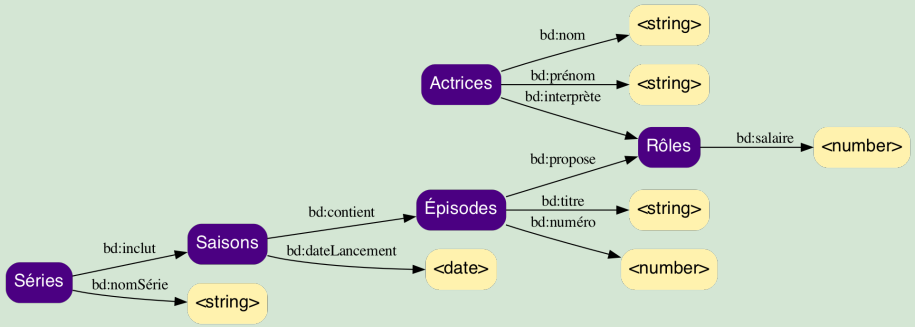



*Diagramme E/A (niveau conceptuel)*




*Représentation libre (niveau logique)*

```
bd:Séries rdf:type rdfs:Class .
bd:Saisons rdf:type rdfs:Class .
bd:Épisodes rdf:type rdfs:Class .
bd:Actrices rdf:type rdfs:Class .
bd:Rôles rdf:type rdfs:Class .

bd:inclut rdf:type rdf:Property .
bd:inclut rdfs:domain bd:Séries .
bd:inclut rdfs:range bd:Saisons .

bd:nomSérie rdf:type rdf:Property .
bd:nomSérie rdfs:domain bd:Séries .
bd:nomSérie rdfs:range xsd:string .
```

```
bd:prénom rdf:type rdf:Property .
bd:prénom rdfs:domain bd:Actrices .
bd:prénom rdfs:range xsd:string .

bd:interprète rdf:type rdf:Property .
bd:interprète rdfs:domain bd:Actrices .
bd:interprète rdfs:range bd:Rôles .

bd:salaire rdf:type rdf:Property .
bd:salaire rdfs:domain bd:Rôles .
bd:salaire rdfs:range xsd:float .
```

...

*Triplets de description (niveau physique)*

# En résumé

RDFs permet de décrire des classes (e.g., type des ressources, relations entre les types des éléments d'un triplet), mais :

- Pas de validation (i.e., deux triplets peuvent être incohérents)
- Perçu comme un vocabulaire plus qu'un schéma, du fait de son expressivité limitée
- Nouveaux langages de description de contraintes pour des graphes RDF avec Shape Expressions (ShEX) et Shapes Constraint Language (SHACL)

---

https://www.w3.org/community/shex/
https://www.w3.org/TR/shacl/

# Plan

Concepts de RDF

Modélisation en RDF

Manipulation avec SPARQL

# Généralités

Plusieurs langages d'interrogation des données RDF :

- **SPARQL**, SquishQL, RDQL, et TriQL (requêtage sans schéma / ontologie)
- **RQL**, SeRQL, et eRQL (requêtage du schéma ou des données, plus expressifs)
- Des langages basés sur XQuery (après sérialisation des données RDF en RDF/XML)
- ...

---

https://en.wikipedia.org/wiki/RDF_query_language

# Le langage SPARQL

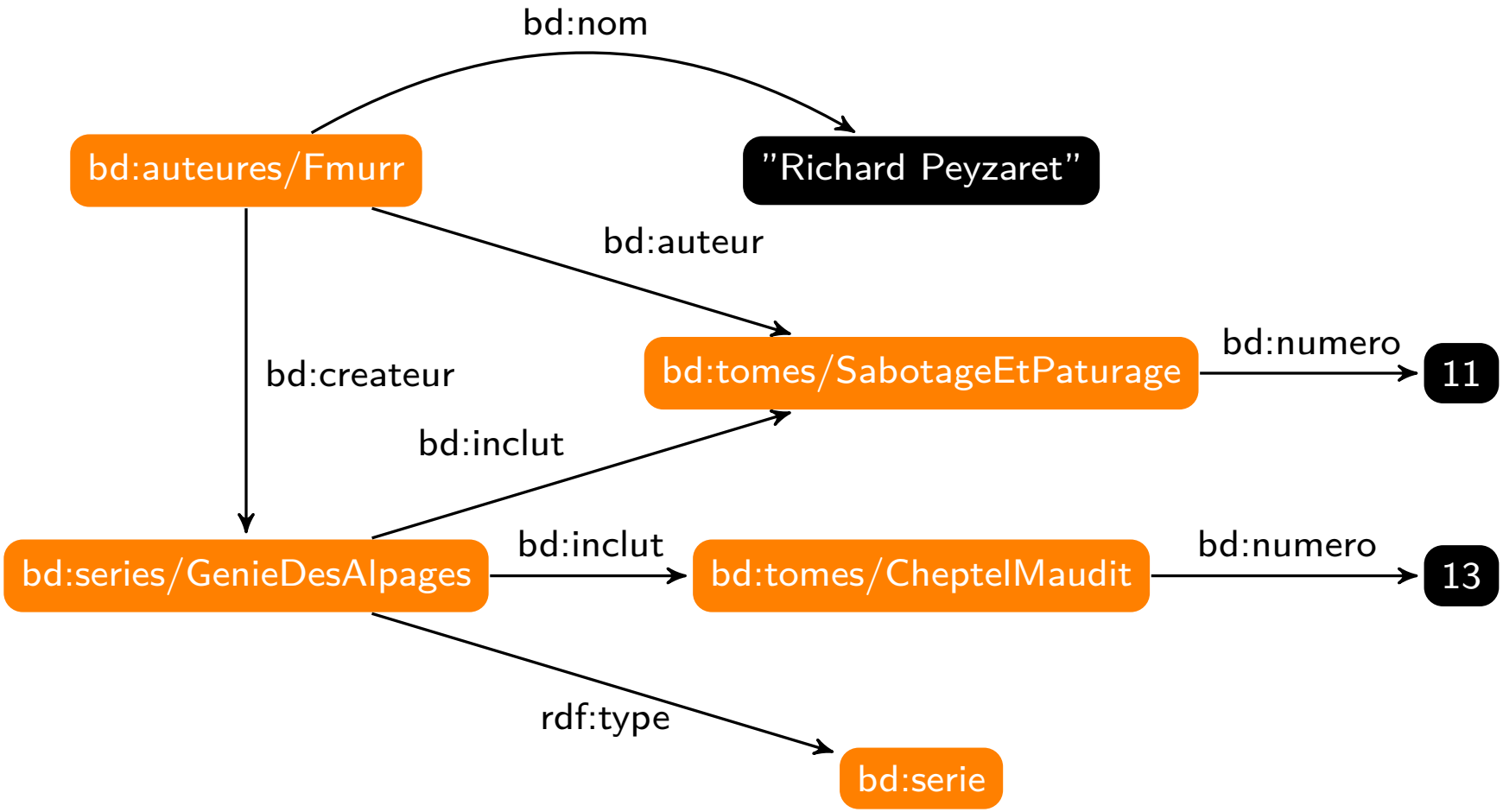
SPARQL Protocol and RDF Query Language (SPARQL) :

- Standard et recommandation W3C
- Syntaxe basée sur Turtle
- Proche des langages d'interrogation relationnel (opérateurs)
- Quatre variantes de requête, dont les clauses :
  - SELECT retourne un résultat tabulaire
  - CONSTRUCT retourne un résultat sous forme de graphe RDF
  - ASK retourne un résultat booléen
- Requêtage d'un graphe
  - "pattern matching" (appariement de motif)

---

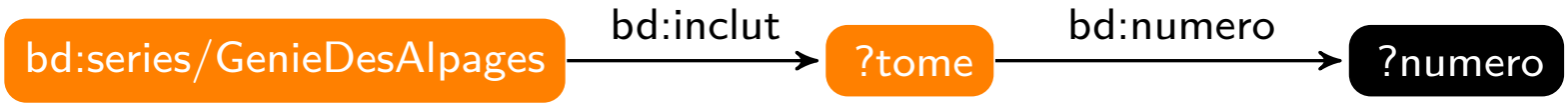
https://www.w3.org/TR/sparql11-query/

# Exemple de "pattern matching"



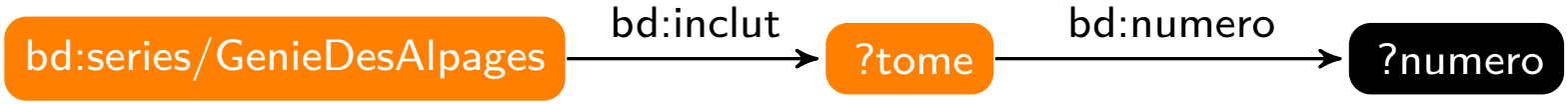
# Exemple de "pattern matching"

**Requête :** le numéro des tomes de la série *Génie des alpages*



# Exemple de "pattern matching"

**Requête :** le numéro des tomes de la série *Génie des alpages*




**Résultat de la requête :**

# Syntaxe d'une requête SPARQL

```
PREFIX prefix1 : <uri1>
SELECT ?var_1 ... ?var_n
WHERE { triplet_1 .
        ...
        triplet_k .
}
```

- Possibilité d'avoir zéro ou plusieurs préfixes
- Clause SELECT remplaçable par CONSTRUCT ou ASK
- Une variable s'écrit **?nom_variable**
  - tout IRI, littéral et prédicat peut être remplacé par une variable

# Exemples de requête basique

```
PREFIX bd: <http://www.bd.db/>
SELECT ?t
WHERE {
  <http://www.bd.db/auteures/Fmurr> bd:auteur ?t .
}
```

*Tomes dont l'auteur est Fmurr*

```
PREFIX bd: <http://www.bd.db/>
PREFIX bds: <http://www.bd.db/series/>
SELECT ?t ?num
WHERE {
    bds:GenieDesAlpages bd:inclut ?t .
    ?t bd:numero ?num .
}
LIMIT 10
```

*Numéros des tomes de la série Génie des Alpages (avec plusieurs préfixes, un opérateur ET implicite et une clause LIMIT)*

# Exemples de requête basique

```
PREFIX bd: <http://www.bd.db/>
SELECT ?t
WHERE {
  <http://www.bd.db/auteures/Fmurr> bd:auteur ?t .
}
```

*Tomes dont l'auteur est Fmurr*

```
PREFIX bd: <http://www.bd.db/>
PREFIX bds: <http://www.bd.db/series/>
SELECT ?t ?num
WHERE {
    bds:GenieDesAlpages bd:inclut ?t .
    ?t bd:numero ?num .
}
LIMIT 10
```

| t | num |
|---|---|
| http://www.bedetheque.com/tomes/SabotageEtPaturage | 11 |
| http://www.bedetheque.com/tomes/CheptelMaudit | 13 |

*Numéros des tomes de la série Génie des Alpages (avec plusieurs préfixes, un opérateur ET implicite et une clause LIMIT)*

# Noeuds anonymes et filtres

- Noeuds anonymes : des noeuds dont l'étiquette n'a pas d'importance pour la requête
  - syntaxe Turtle : **[ ]** ou **_ :label** (réutilisable)
- Opérateur OU :

{ triplet$_1$ } UNION { triplet$_2$ }

- Filtres (complémentaires au WHERE), mais pas du matching

FILTER( condition )

  - une condition est une combinaison (&&, ||, !) d'expressions (=, !=, <, >, etc.)
  - attention, les littéraux doivent être typés

# Exemples de requête avec filtre, union et noeuds anonymes

```
PREFIX bd: <http://www.bd.db/>
SELECT ?a
WHERE {
    ?a bd:createur _:s .
    _:s rdf:type [] .
}
```

*Créateurs dont l'oeuvre est typée (avec deux noeuds anonymes, dont _ :s qui est réutilisable)*

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX bd: <http://www.bd.db/>
SELECT ?t
WHERE {
    {[] bd:auteur ?t . }
    UNION
    {?t bd:numero ?num .
    FILTER(?num > "12"^^xsd:integer)}
}
```

*Tomes ayant un auteur ou un numéro supérieur à 12*

# Exemples de requête avec filtre, union et noeuds anonymes

```
PREFIX bd: <http://www.bd.db/>
SELECT ?a
WHERE {
    ?a bd:createur _:s .
    _:s rdf:type [] .
}
```

*Créateurs dont l'oeuvre est typée (avec deux noeuds anonymes, dont _ :s qui est réutilisable)*

```
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX bd: <http://www.bd.db/>
SELECT ?t
WHERE {
    {[] bd:auteur ?t . }
    UNION
    {?t bd:numero ?num .
    FILTER(?num > "12"^^xsd:integer)}
}
```

| t |
| --- |
| http://www.bedetheque.com/tomes/SabotageEtPaturage |
| http://www.bedetheque.com/tomes/CheptelMaudit |

*Tomes ayant un auteur ou un numéro supérieur à 12*

# Projection et regroupements

Clause SELECT (projection) :

- Renommage des variables : AS
- Suppression des doublons : DISTINCT
- Fonctions (natives ou XPath)

Clause GROUP BY et HAVING :

- Une ou plusieurs variables de regroupement
- Condition sur les groupes
- Fonctions d'agrégation (COUNT, MAX, AVG, etc.)

---

https://www.w3.org/TR/sparql11-query/#expressions
https://www.w3.org/TR/sparql11-query/#aggregates

# Exemple de requête avec regroupement

```
PREFIX bd: <http://www.bd.db/>
SELECT ?s (COUNT( ?t ) AS ?nbTomes )
WHERE { ?s bd:inclut ?t .     }
GROUP BY ?s
```

*Le nombre de tomes par série*

```
PREFIX bd: <http://www.bd.db/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT (CONCAT( ?s , " : ", ?a ) AS ?infos )
WHERE {
    ?s bd:inclut ?t .
    ?a bd:createur ?s .
    ?t bd:numero ?num .
}
GROUP BY ?s ?a
HAVING(MAX( ?num ) > 10)
```

*Pour les séries dont le tome maximal est supérieur à 10, le nom de la série concaténé au nom de son créateur*

# Exemple de requête avec regroupement

```
PREFIX bd: <http://www.bd.db/>
SELECT ?s (COUNT( ?t ) AS ?nbTomes )
WHERE { ?s bd:inclut ?t .     }
GROUP BY ?s
```

| s | nbTomes |
|---|---|
| http://www.bedetheque.com/series/GenieDesAlpages | 2 ^^xsd:integer |

*Le nombre de tomes par série*

```
PREFIX bd: <http://www.bd.db/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
SELECT (CONCAT( ?s , " : ", ?a ) AS ?infos )
WHERE {
    ?s bd:inclut ?t .
    ?a bd:createur ?s .
    ?t bd:numero ?num .
}
GROUP BY ?s ?a
HAVING(MAX( ?num ) > 10)
```

*Pour les séries dont le tome maximal est supérieur à 10, le nom de la série concaténé au nom de son créateur*

# Outils utilisant SPARQL

- **Endpoints** SPARQL (jeux de données fournis) : éditeurs DBpedia et Wikidata, YASQUI, SPARQLer, Twinkle, etc.
- **Librairies** ou **frameworks** : RDFlib (Python), Apache Jena, Apache Marmotta, etc.
- **SGBD** : GraphDB, BlazeGraph, AllegroGraph, BrightstarDB, Dydra, Stardog, etc.

---

https://dbpedia.org/sparql
https://query.wikidata.org/
https://legacy.yasgui.org/
https://rdflib.readthedocs.io/en/latest/
https://jena.apache.org/
https://marmotta.apache.org/
https://en.wikipedia.org/wiki/List_of_SPARQL_implementations

# Pour aller plus loin

- Interrogation de plusieurs graphes (clause FROM)
- Rendre certains patterns (triplets) optionnels (OPTIONAL)
- Pas de correspondance pour un pattern (NOT EXISTS)
- Sous-requêtes imbriquées
- Requêtes de création, suppression, mise à jour (CREATE GRAPH, INSERT DATA, etc.)

---

https://www.w3.org/TR/sparql11-query/
https://www.w3.org/TR/sparql11-update/

# Bilan

- Des données sous forme de triplets qui forment des graphes orientés et étiquetés
- Plusieurs sérialisations de RDF
- Un schéma RDFs pour classifier les ressources et décrire les données
- Langage de requêtes SPARQL (syntaxe similaire à SQL)

Notions approfondies dans l'UE MIF24 - BD NoSQL