MLBDA

Modèles et Langages pour les Bases de Données Avancées

UE 4I801

Master d'informatique

spécialité DAC niveau M1

Anne Doucet

anne.doucet@lip6.fr

Septembre-décembre 17

http://www-bd.lip6.fr/

Objectifs

Présenter des modèles et langages pour le développement de nouveaux types d'applications (Web 2.0, réseaux sociaux, réseaux de capteurs, open data, recherche d'information, …).

Apprendre les technologies de gestion de données du Web (XML, RDF, SPARQL…)

Plan

- Modélisation des données
- Les modèles objet et relationnel-objet
- Langage de requêtes SQL3
- Modèle semi-structuré et XML
- XSchema
- XPath
- XQuery
- Modèle sémantique et RDF
- SPARQL
- NoSQL

Bibliographie

- G. Gardarin : Bases de Données – objet et relationnel, Eyrolles, 2003.
- H. Garcia-Molina, J.D.Ullman, J. Widom : Database System Implementation, Prentice Hall, 2000.
- R. Ramakrishnan, Gehrke J. : Database Management Systems, mc-Graw Hill, 3ème édition.
- G. Gardarin : XML : des bases de données aux services Web, Dunod, 2002.
- Documentation XML : www.w3c.org/TR/REC-xml
- http://www.w3.org/TR
- F. Gandon, C. Faron-Zucker, O. Corby : Le Web sémantique, Dunod, 2012
- Documentation RDF : http://www.w3.org/RDF/
- S. Abiteboul, I. Manolescu, P. Rigaux, MC.Rousset, P. Senellart : Web Data Management, 2011, Cambridge University Press

Module MLBDA
Master Informatique
Spécialité DAC

Cours 1- Modélisation des données

PLAN

- Objectifs et fonctions des SGBD
- Forces et limites du modèle relationnel
- Evolution des modélisations et des applications
  - Données structurées complexes
  - Données semi-structurées
  - Données du Web sémantique
  - NoSQL
- Conclusion

Objectifs des SGBD

- **Contrôle intégré des données**
  - cohérence et intégrité
  - partage
  - performances d'accès
  - sécurité

- **Indépendance des données**
  - logique : cache les détails de l'organisation conceptuelle des données
  - physique : cache les détails du stockage physique des données

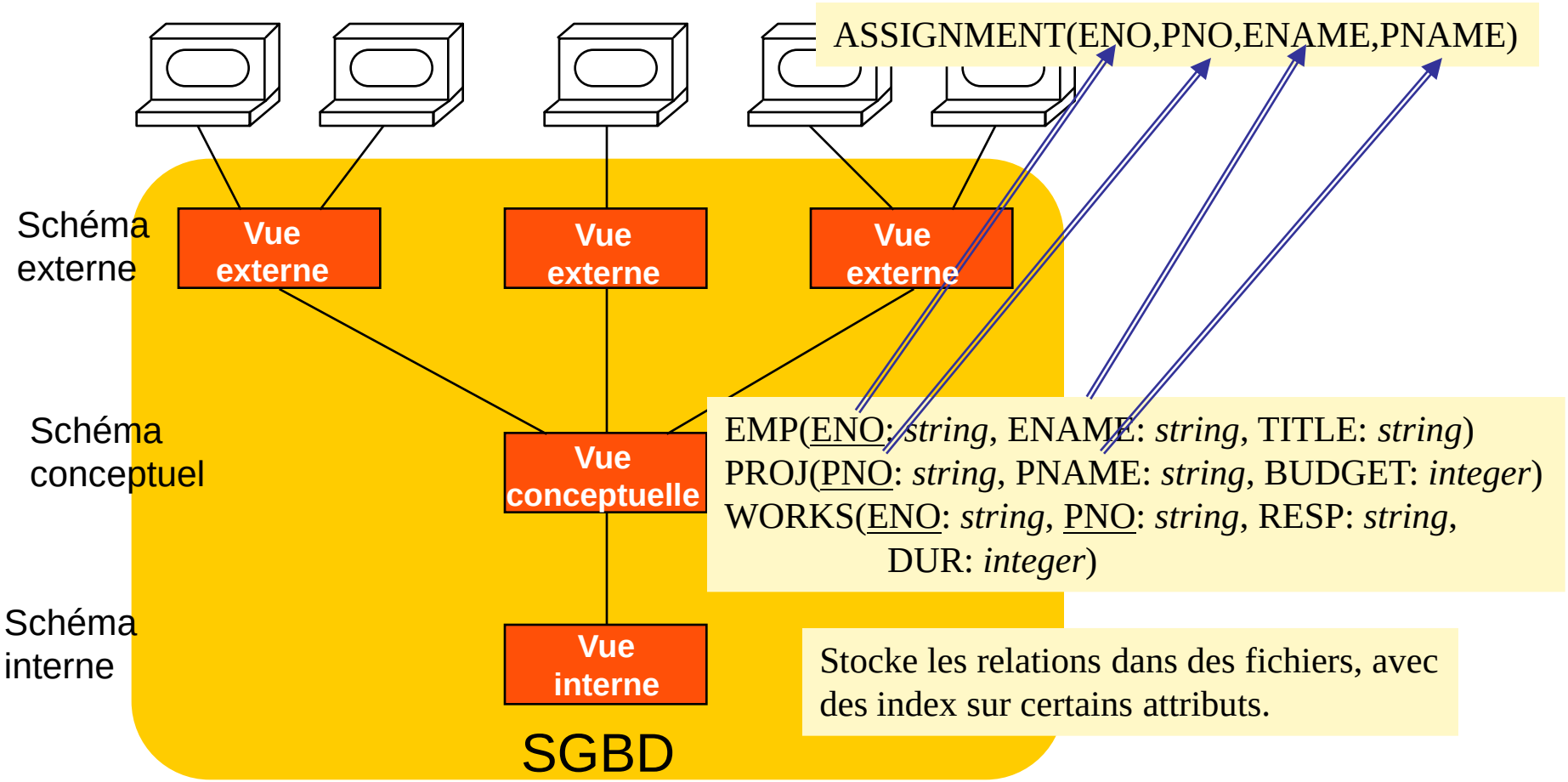
Fonctions

- **Schéma intégré**
  - vue uniforme des données, par ex. sous formes de relations (ou tables)
- **Intégrité déclarative et cohérence**
  - $24000 \leq \text{Salaire} \leq 250000$
  - l'utilisateur spécifie et le SGBD valide
- **Vues**
  - réorganisation de relations pour certaines classes d'utilisateurs
- **Accès déclaratif**
  - avec un langage de requête (SQL), l'utilisateur spécifie ce qu'il veut obtenir et non ce qu'il faut faire pour l'obtenir (le quoi et non le comment)

Fonctions

- **Traitement et optimisation de requêtes**
  - performances obtenues automatiquement
- **Transactions**
  - exécution des requêtes par des unités atomiques
  - indépendance à la concurrence multi-utilisateurs et aux pannes
- **Conception d'applications BD**
  - conception visuelle des schémas de BD
  - conception des traitements et des interfaces graphiques
- **Administration système**
  - outils d'audit et de réglage (tuning)
  - visualisation des plans d'accès

Architecture ANSI/SPARC



Modèle relationnel

EMP

ENO	ENAME	TITLE
01	Max	Ingénieur
02	Paul	Développeur
03	Marie	Développeur
04	Léa	Ingénieur
05	Luc	Développeur

PROJ

PNO	PNAME	BUDGET
P1	Paye	500M
P2	Stocks	800M
P3	Livraisons	200M

WORKS

ENO	PNO	RESP	DUR
01	P1	01	24
02	P2	04	20
03	P1	01	20
04	P2	04	18
05	P1	01	15

Select ENAME, TITLE from EMP;

Select ENAME from EMP where Title='Ingénieur';

```
Select ENAME
from EMP E, PROJ P, WORKS W
where E.ENO=W.ENO
and W.PNO = P. PNO
and P.PNAME = 'Paye';
```

Apports du modèle relationnel

- Simplicité des concepts et du schéma
- Bon support théorique
- Langage d'interrogation déclaratif
- Haut degré d'indépendance des données
- Optimisation des accès à la BD
  - bonnes performances
- Gestion de contraintes d'intégrité

Limites du modèle relationnel

- **Trop grande simplicité du modèle de données**
  - 1ère forme normale de Codd
    - attributs mono-valués : n-uplets plats
  - Pauvreté du système de typage
    - Types prédéfinis (entier, réel, chaîne, …) : pas de possibilité d'extension
  - Inadapté aux objets complexes (ex: documents structurés)
    - Un objet du monde réel est modélisé à l'aide de plusieurs relations : mauvaise lisibilité, perte d'information sémantique, nombreuses jointures
- **Très bon support pour les applications de gestion, mais mal adapté pour d'autres types d'applications**
  - CAO, CFAO
  - BD Géographiques
  - BD techniques, documentation
  - …

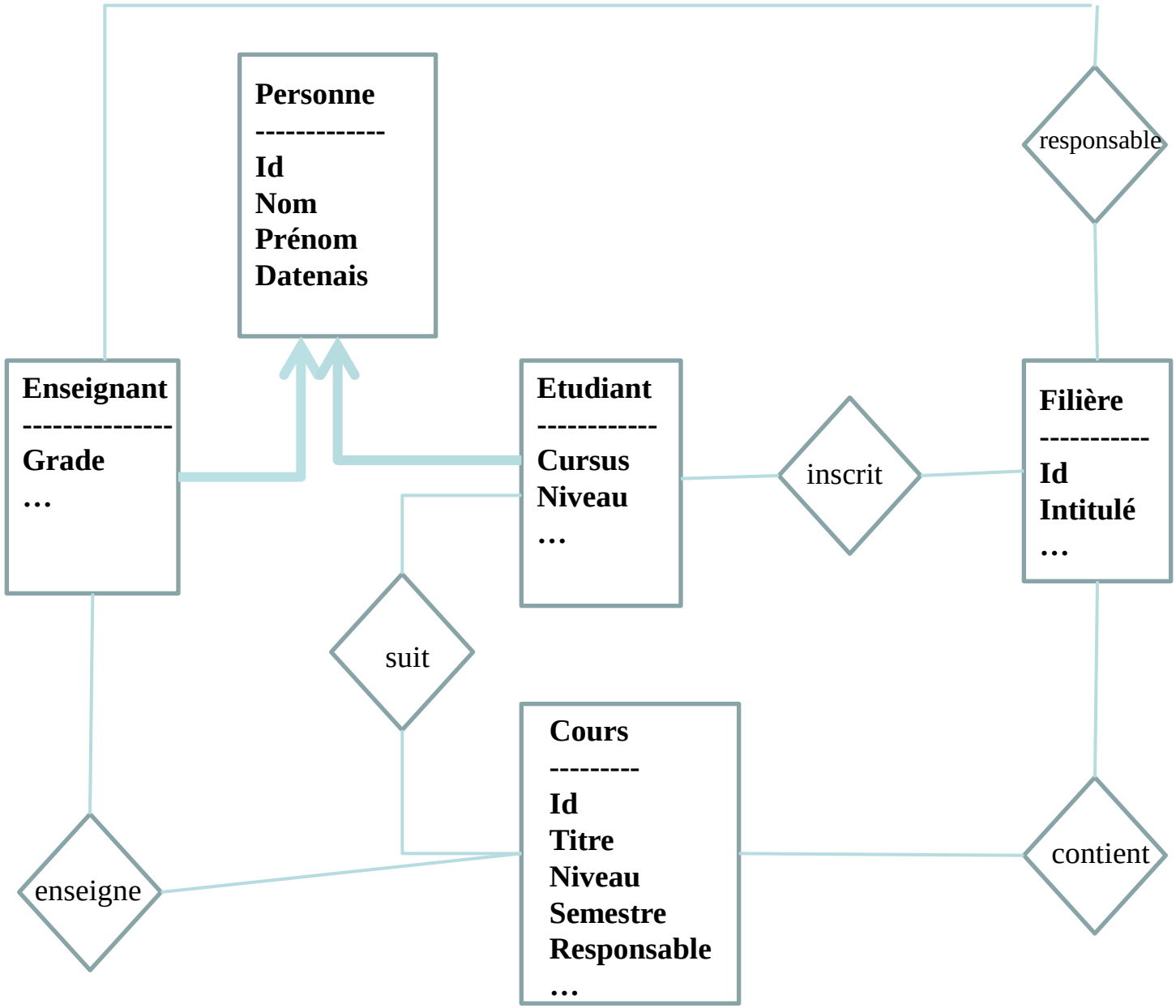
Limites du modèle relationnel

- **Langage d'interrogation et de manipulation non complet**
  - Pas de récursion
  - Pas de structures de contrôle : conditionnelles, boucles
  - L'utilisation de deux langages (SQL + un langage de programmation) provoque un dysfonctionnement du système
    - Sql déclaratif, LP procédural
    - Systèmes de typage différent
    - Espaces de noms différents
    - Utilisation de curseurs pour manipuler les ensembles
    - Mauvaises performances
- **Ensemble fermé d'opérateurs : algèbre relationnelle**
  - Critères d'optimisation liés à ces opérateurs
- **Index restreints aux types de base**
- **Pas de versions, pas de transactions longues**

SGBD relationnels : bilan

- Les SGBD relationnels occupent une place majeure dans l'industrie depuis plusieurs décennies:
  - Une technologie efficace, sûre, éprouvée
  - Très bonnes performances
  - Très nombreuses applications
- Les SGBDR sont bien adaptés aux données de gestion (modélisation simple), peu dispersées (peu de SGBD répartis).

Données complexes



Constat

- Eclatement des entités, informations dispersées dans plusieurs relations
- Multiplication des relations, nombreuses jointures
- Informations redondantes, non factorisées (étudiant, enseignant)
- Difficile de représenter simplement les objets complexes (constitués d'autres objets et/ou d'ensembles)

Besoins

- Modèle de données plus riche
- Conception plus proche du monde réel

Et aussi

- Gestion de gros objets (données multimedia) avec structures de stockage adaptées
- Nouveaux modèles de transactions (transactions longues, distribuées, imbriquées..)
- Prise en compte des versions
- Indépendance des objets et des traitements
- Extensibilité
- Meilleure intégration des langages d'interrogation et de manipulation

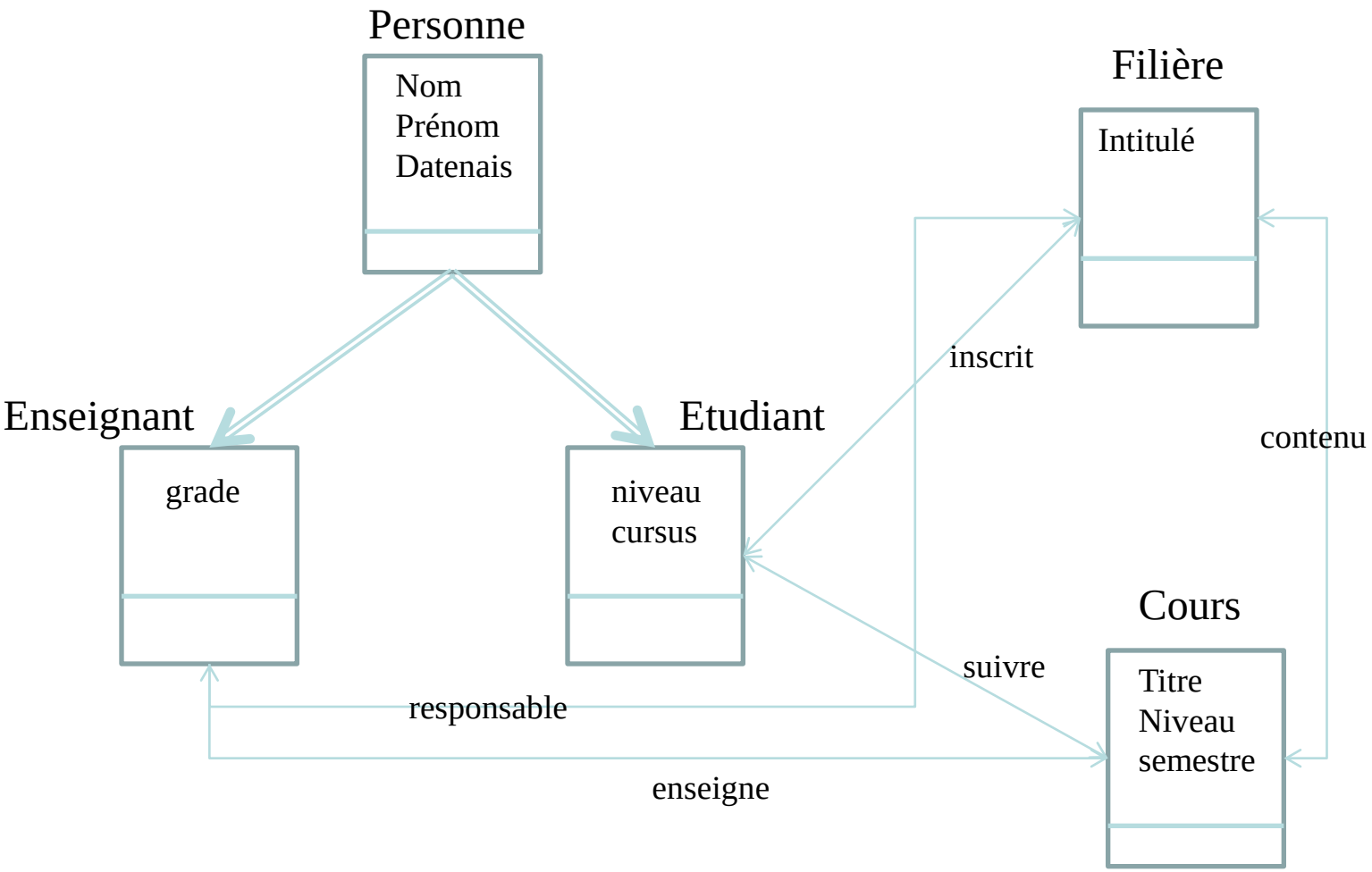
Approches

- Extensions du modèle relationnel
- Langages de programmation persistants
- Systèmes orienté-objet
  - Modèle objet : ODMG et OQL
  - Modèle relationnel-objet : SQL3

Modélisation en objets

- Les entités du monde réel sont modélisées par des objets.
- Un objet peut être atomique, ou composé d'autres objets.
- Chaque objet a un identificateur unique.
- Un objet a une interface et des opérations qui lui sont applicables.
- Les objets de même nature sont regroupés dans des classes, reliées par des liens de composition et/ou d'héritage.
- Un schéma de base de données objet est un graphe de classes.

Exemple


Personne
Nom
Prénom
Datenais
Filière
Intitulé
inscrit
Enseignant
grade
Etudiant
niveau
cursus
contenu
Cours
Titre
Niveau
semestre
suivre
responsable
enseigne

Données du Web

- Des données très hétérogènes
  - Documents HTML, SGML, Latex, PDF, txt, …
  - Multimédia : son, graphique, images, vidéos, photos, dessins, etc.
- Des applications très diverses
  - Commerce électronique
  - Portail d'information
  - Intranet
  - Publication en ligne
- Toutes les catégories d'utilisateurs

Exemples

- annexe_intro\a1_Abeille.pdf
- annexe_intro\a2_Moustique.pdf
- annexe_intro\f1.pdf
- annexe_intro\f2.pdf
- annexe_intro\v1.pdf
- annexe_intro\v2.pdf

Constat

- Des informations différentes pour décrire des données similaires
- Des présentations différentes, mais on retrouve une certaine régularité
- Des modifications fréquentes, une évolution rapide.

Enjeux

- **Représenter des données**
  - dont la structure est irrégulière, implicite, partielle, indicative
  - dont le schéma est incomplet, très évolutif, vaste, non défini à l'avance

- **Et les interroger**
  - efficacement
  - avec un langage déclaratif

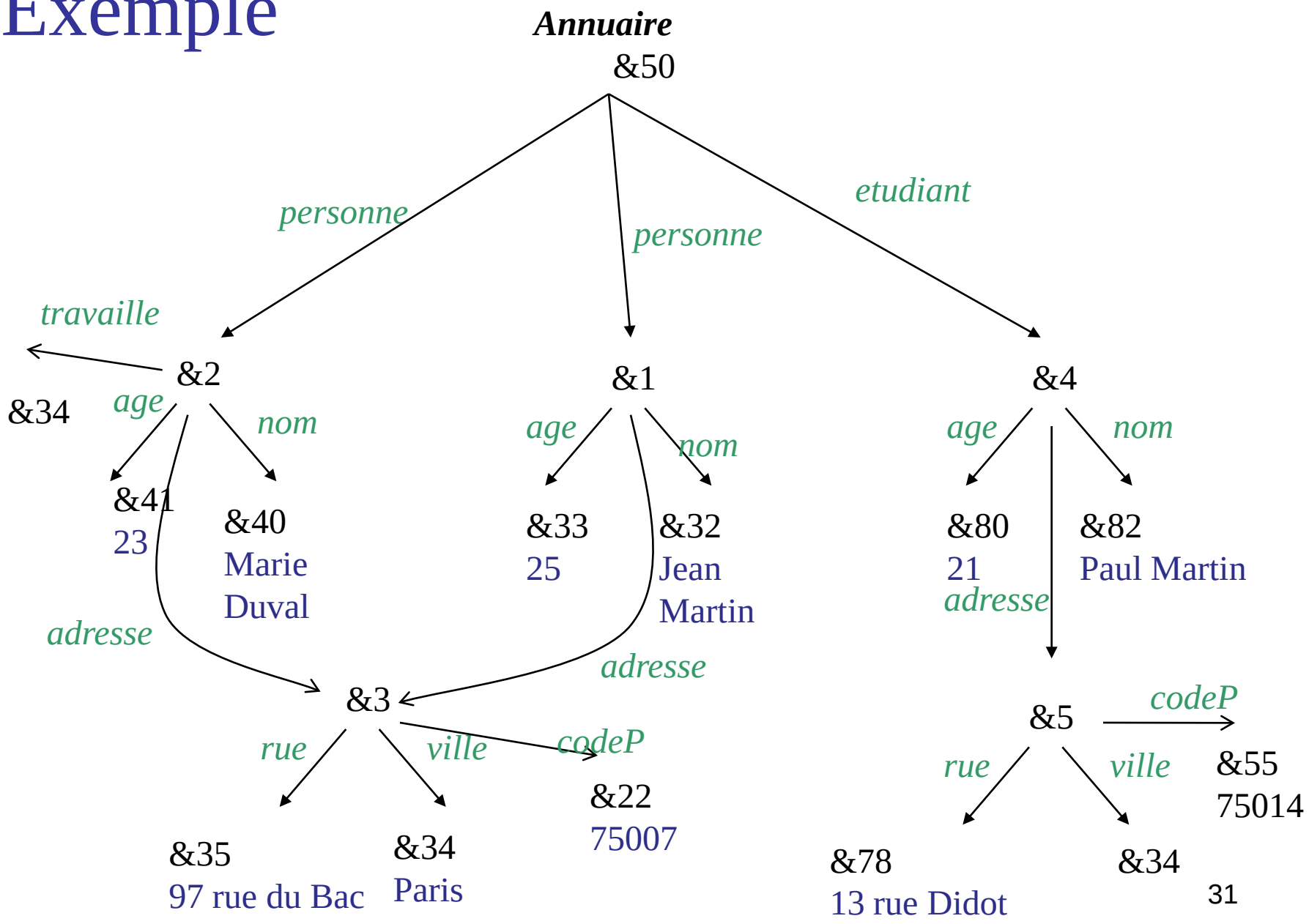
Données semi-structurées

- **Les données semi-structurées sont**
  - sans schéma
  - autodescriptives, càd.
    - pas de séparation entre données et types
    - une donnée contient son propre type
    - elles peuvent être interprétées indépendamment de toute autre information.

Modèles semi-structurés

- Graphes dont les nœuds sont des objets et dont les arcs sont étiquetés.
- Les données sont stockées dans les feuilles (objets atomiques).
- Les étiquettes donnent des informations sur le schéma.

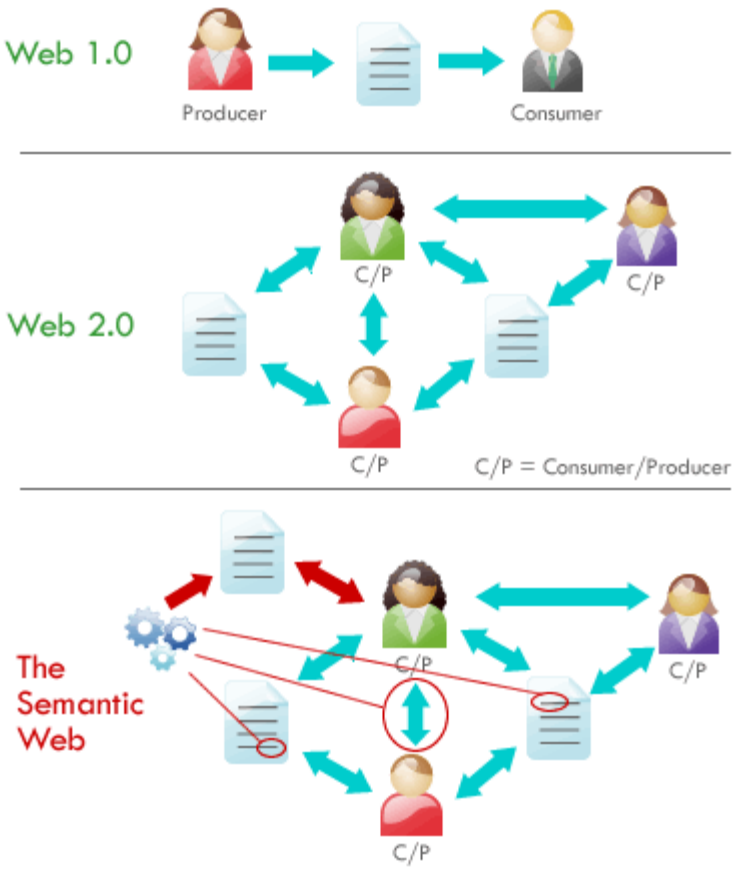
Exemple


Annuaire
&50
personne
personne
etudiant
travaille
&2
&1
&4
&34
age
nom
age
nom
age
nom
&41
23
&40
Marie
Duval
&33
25
&32
Jean
Martin
&80
21
&82
Paul Martin
adresse
adresse
adresse
&3
&5
codeP
rue
ville
codeP
rue
ville
&55
75014
&22
75007
&35
97 rue du Bac
&34
Paris
&78
13 rue Didot
&34

XML

- XML : représentation des données
- Xschema : description du schéma
- Xpath : navigation dans l’arbre XML
- Xquery : interrogation

Le Web sémantique



Rendre le contenu des ressources du Web plus accessibles et plus utilisables.

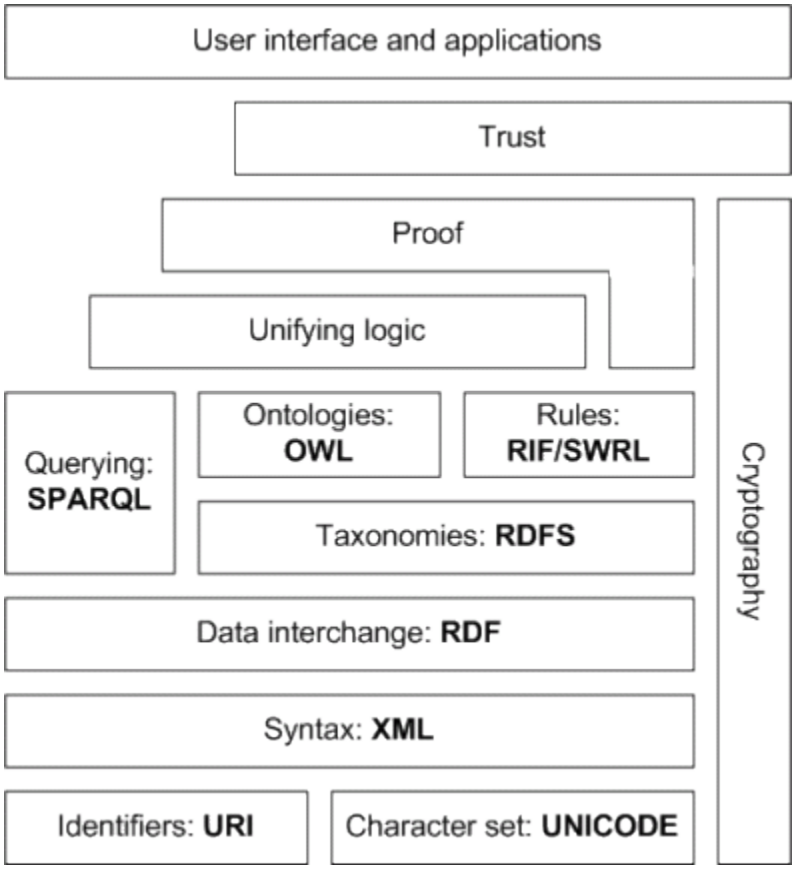
Exploiter sémantiquement les données.

Applications

Besoins

- Représenter la sémantique
- Établir des liens entre les données
- Exploiter ces liens
  - Déduire de nouvelles informations
- Indexer et interroger

Les standards du Web sémantique



Modèle de données sémantique

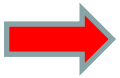
- RDF (Resource Description Framework) : description des données, annotations sémantiques
- RDFS : description des ontologies
- SPARQL : interrogation des données RDF

Changement d'échelle (Big Data)

- **Données**
  - Web2.0 : réseaux sociaux, news, blogs,…
  - Graphes, ontologies
  - Flux de données : capteurs, GPS,…

Très gros volumes, données pas ou peu structurées

- **Traitements**
  - Moteurs de recherche
  - Extraction, analyse
  - Recommandation, filtrage collaboratif



Transformation, agrégation, indexation

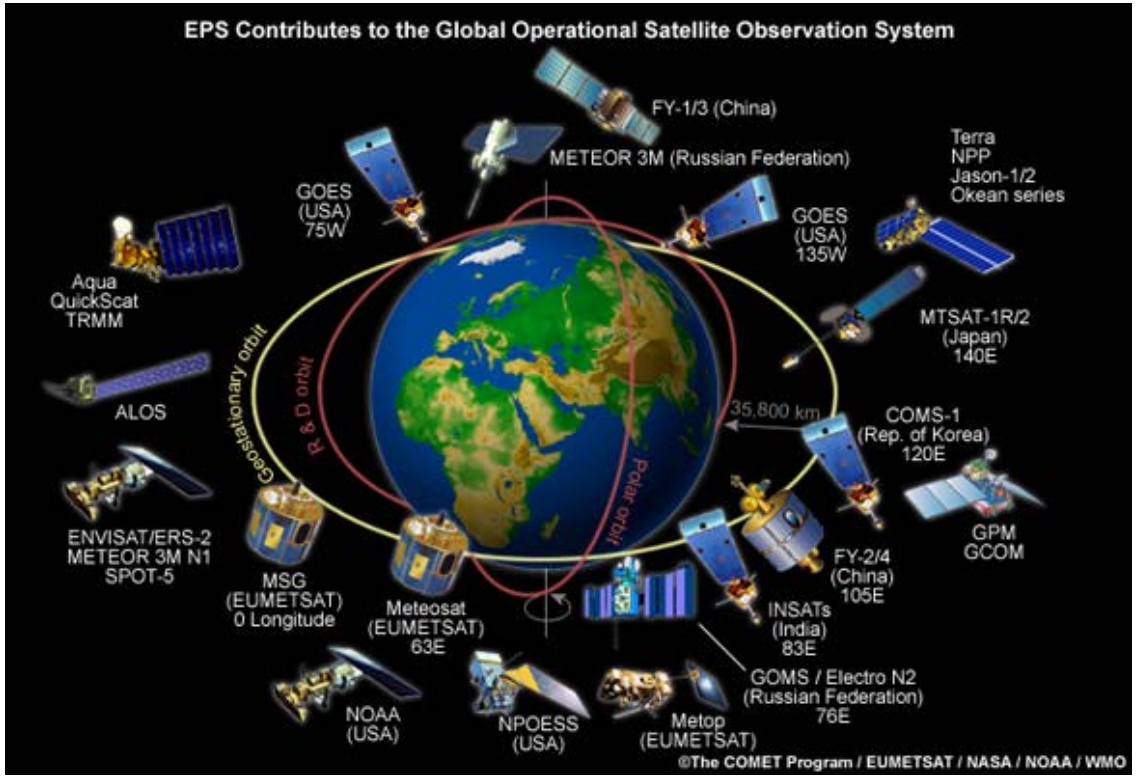
- **Infrastructures**
  - Clusters, réseaux mobiles, data centers, microprocesseurs multicoeurs

Distribution, redondance, parallélisation

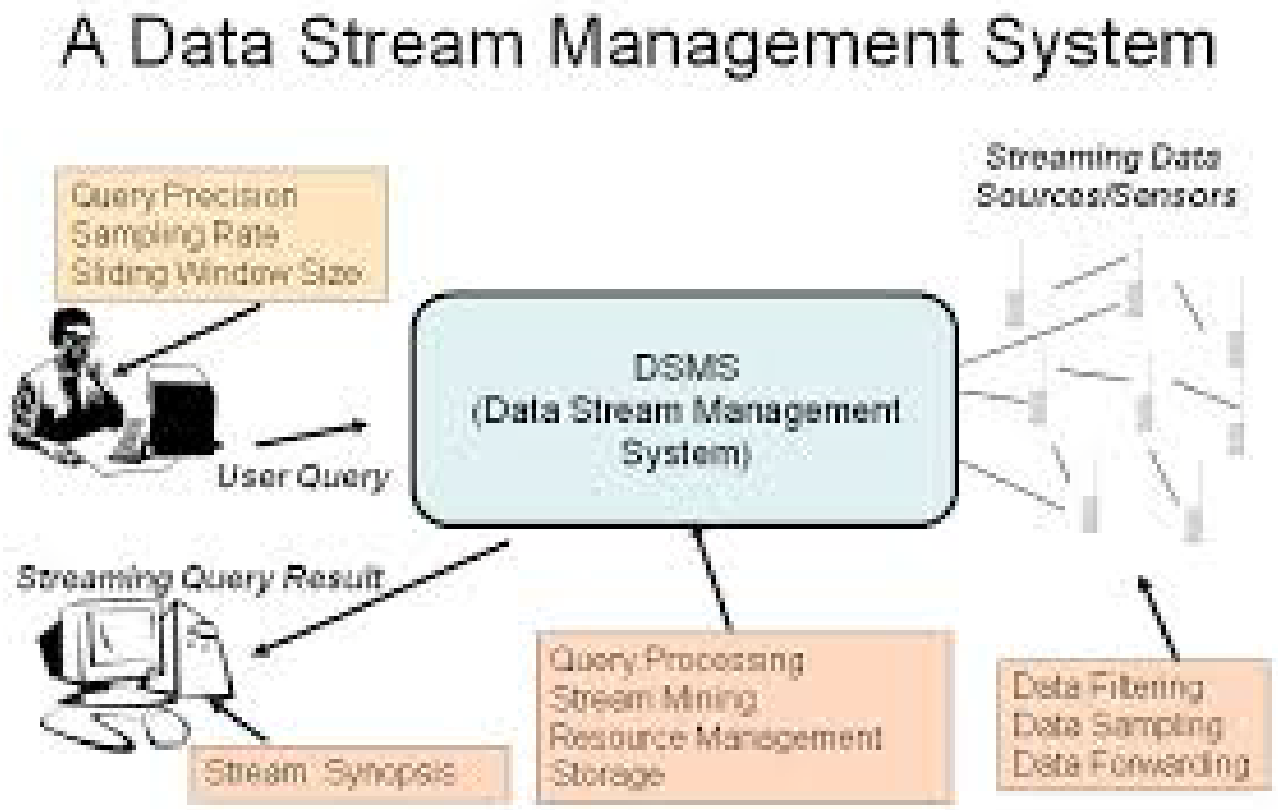
Comment faire ?

- Solutions spécifiques
  - Ex : Gestionnaire de flux de données
- Solution plus générale:
  - NoSQL

Flux de données



Gestionnaire de flux de données



Caractéristiques

- Très grosses quantités de données, très évolutives
- Données volatiles
- L'ordre d'arrivée des données est important
- Taille mémoire limitée

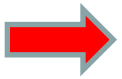
Comment interroger ?

Requêtes continues

- Interrogation par échantillonage
- Interrogation par fenêtrage
  - Les n derniers éléments du flux
  - Les éléments des n dernières secondes
- Mise à jour incrémentale des résultats en temps réel

Systèmes NoSQL (not only SQL)

- Systèmes qui abandonnent certaines propriétés des SGBDR (one size does not fit all):
  - Le langage d'interrogation
  - Le contrôle du schéma
  - La concurrence d'accès



Flexibilité, passage à l'échelle

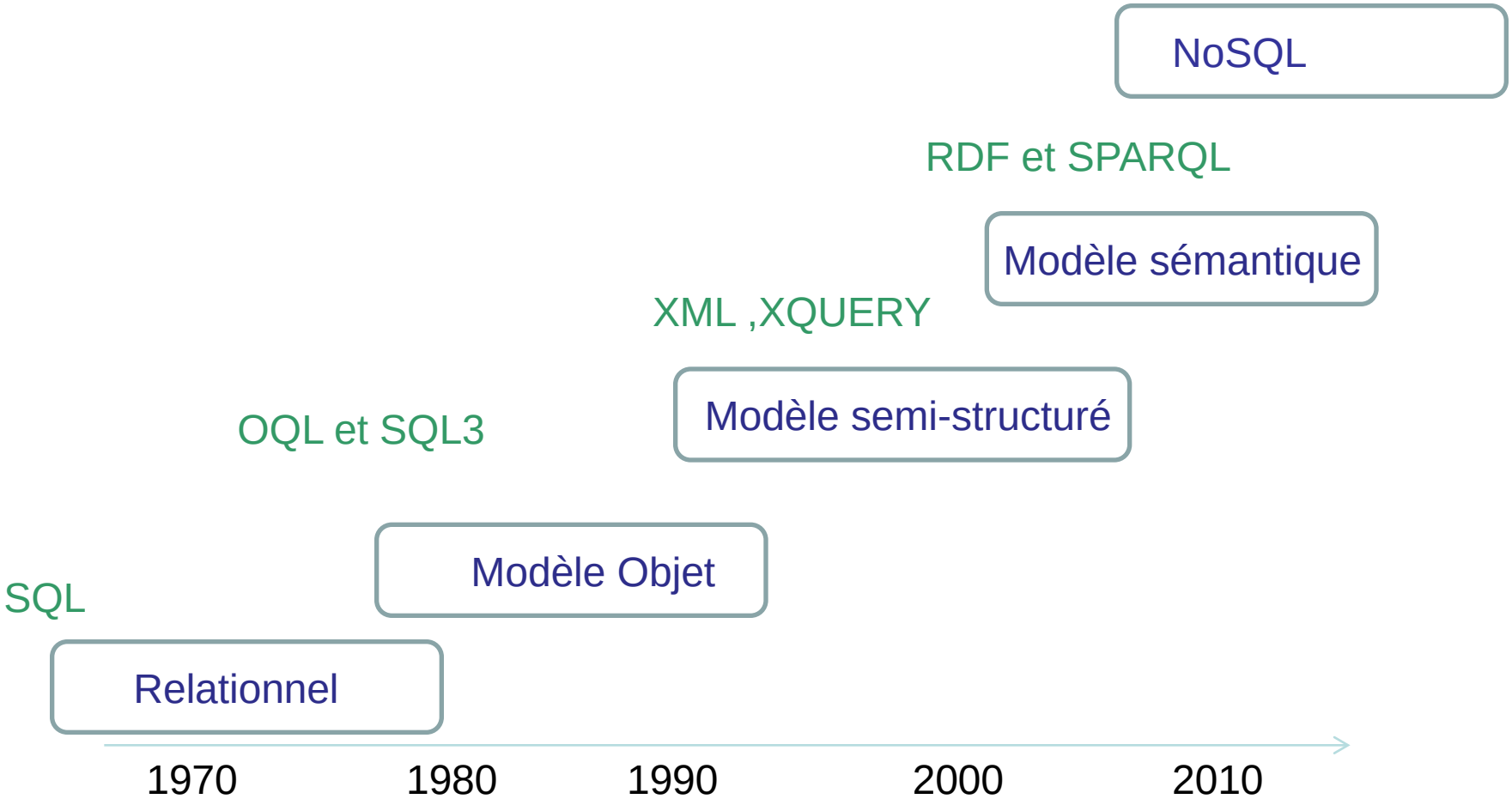
- Comment ?
  - Très forte distribution des données et des traitements (nombreux serveurs)
  - Adaptation élastique à la charge et au volume (clouds)

Systèmes NoSQL

- Modèle de données plus flexible, plus intuitif (le plus simple étant clé-valeur)
- Exploitation de ce modèle pour distribuer plus facilement et plus efficacement (cloud, MapReduce)
- Langages spécialisés, API simples
- Abandon des contraints fortes des SGBDR (propriétés ACID)

- **4 catégories** :
  - Bases orientées colonnes,
  - Stockage de couples clé-valeur,
  - Bases de documents,
  - Bases de graphes

Evolution des modèles



Conclusion

- Evolution des modèles en fonction du type des données et des applications
- Rester proche du ‘monde réel’
- Evolution des langages
- Adaptation, relâchement des techniques des bases de données (ex: NoSQL)