Objets Structurés et Semi-Structurés: Perspective Bases de données

Nacer Boudjlida

LORIA, UHP Nancy 1

http://www.loria.fr/~nacer

(Strasbourg, Septembre 2002)

Objectifs du tutoriel

Citation de VLDB (*http://www.vldb.org/future.html)*, Dec. 1999:

"… the database role of providing a storage manager, i.e. **core DB technology, remains central**… The **focus** of our community should turn to the investigation of **how core technology can become more widespread and usable**, by concentrating on the description of **new application areas**, on the method and tools for data analysis, design and integration, on the **technologies for data deployment in modern architectures** (middleware, wireless technology, the Web), and in general on all problems and **challenges** which are due to the need of **using very large databases in new contexts**."

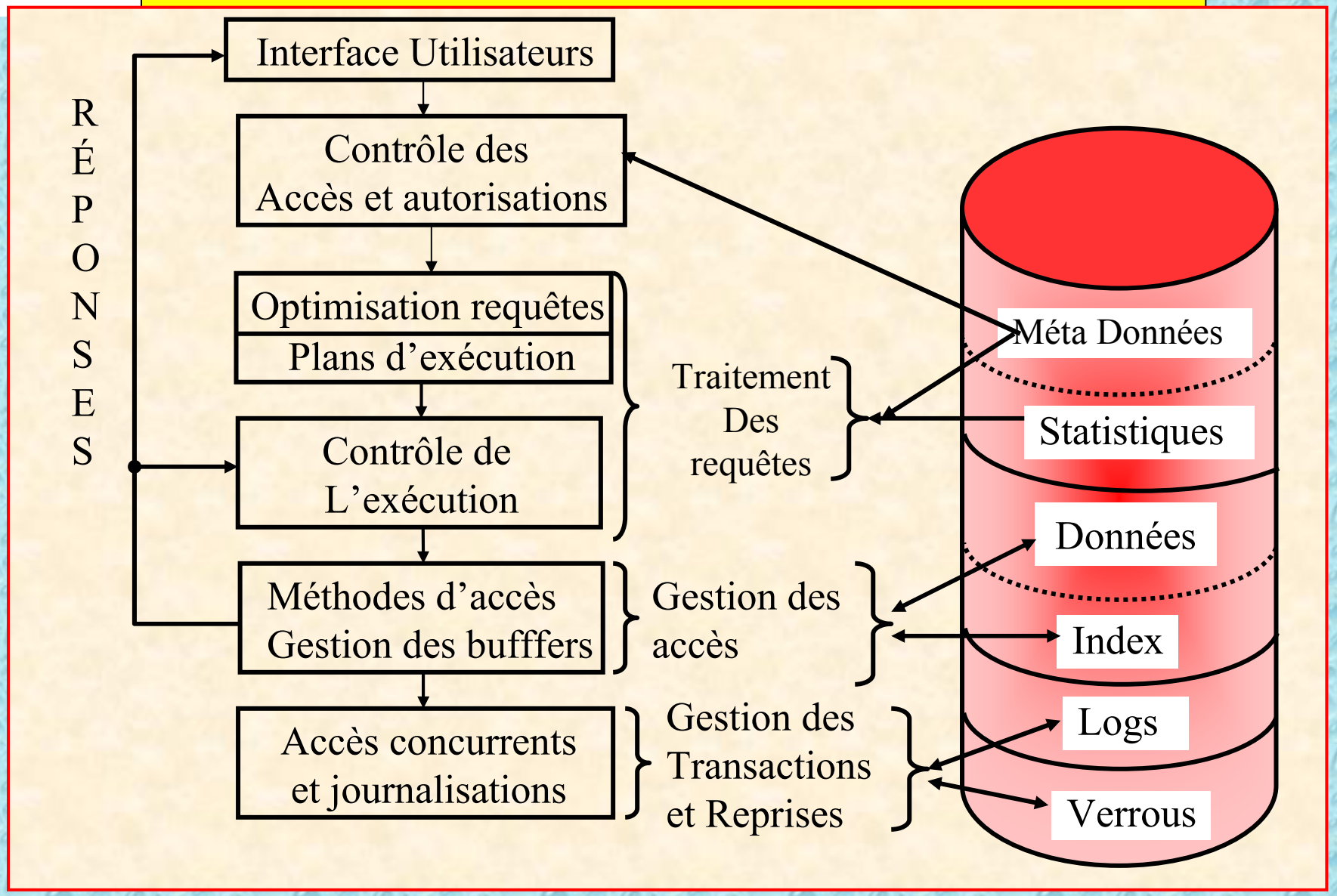
Objectif du tutoriel

- Représentation et manipulation
  - Données (fortement) structurées
  - Objets et données non ou semi-structurés
- Evaluer l'adéquation (d 'une partie) de la technologie Bases de données par rapport à de nouveaux types d'applications ou de nouvelles architectures logicielles (Client/Serveur + Web)

Fonctions d'un SGBD

- Description de données ("objets", attributs, relations)
- Manipulation: Langages de haut niveau (orientés ensembles)
- Confidentialité et Intégrité
- Accès concurrents
- Sécurité de fonctionnement
- Efficacité : Accès et Traitement des Requêtes

Architecture d'un SGBD



Classes de SGBD

- Un SGBD « comprend » et implémente un *modèle de représentation de données*
  - CODASYL (SGBD hiérarchiques/réseaux)
  - Relationnel (SGBD R)
  - Orienté Objets (SGBDO)
  - Données Non ou Semi-Structurées (Web-SGDB ?)
- Evolution du domaine :
  - % Autres domaines (Programmation, Réseaux, IA, etc.)
  - % Besoins des nouvelles applications (CAD/CAM, Wflow, Ingénierie du Logiciel, etc.)

Structure de la Présentation

Partie I : Survol de la technologie relationnelle

I.1- Modèle et Langages

I.2- Traitement des requêtes

- Bases Centralisées
- Bases Distribuées

Partie II : Modèles et systèmes pour objets complexes

Partie III : Web et Bases de données

I.1- Modèle et Langages Relationnels

- Concepts du Modèle de Données
  - *Domaine* : Ensemble de valeurs
  - *Attribut* : Nom associé à un domaine
  - *Relation* : Sous-ensemble du produit cartésien des domaines
  - Vision logique d'une relation : Table
    - *Colonnes* : Valeurs des attributs
    - Lignes : tuples, "faits''

Langages Relationnels (suite)

- Déclaratifs, Orientés ensembles

1/2) Algèbre Relationnelle : Opérandes et résultats = Relations

- Sélection/Restriction (σ), Projection (Π)
- Produit Cartésien (X), Union ($\cup$), Différence (\)
- Note: Join(R, S, Cond) = σ_{Cond} (R x S)
  - Condition : R.Attribut θ S.Attribut
  - θ : Opérateur de Comparaison (=, <, >, etc.)

$R \underset{\theta \text{ expression}}{\bowtie} S$

2/2) SQL: *Dialecte Standard* des SGBDR

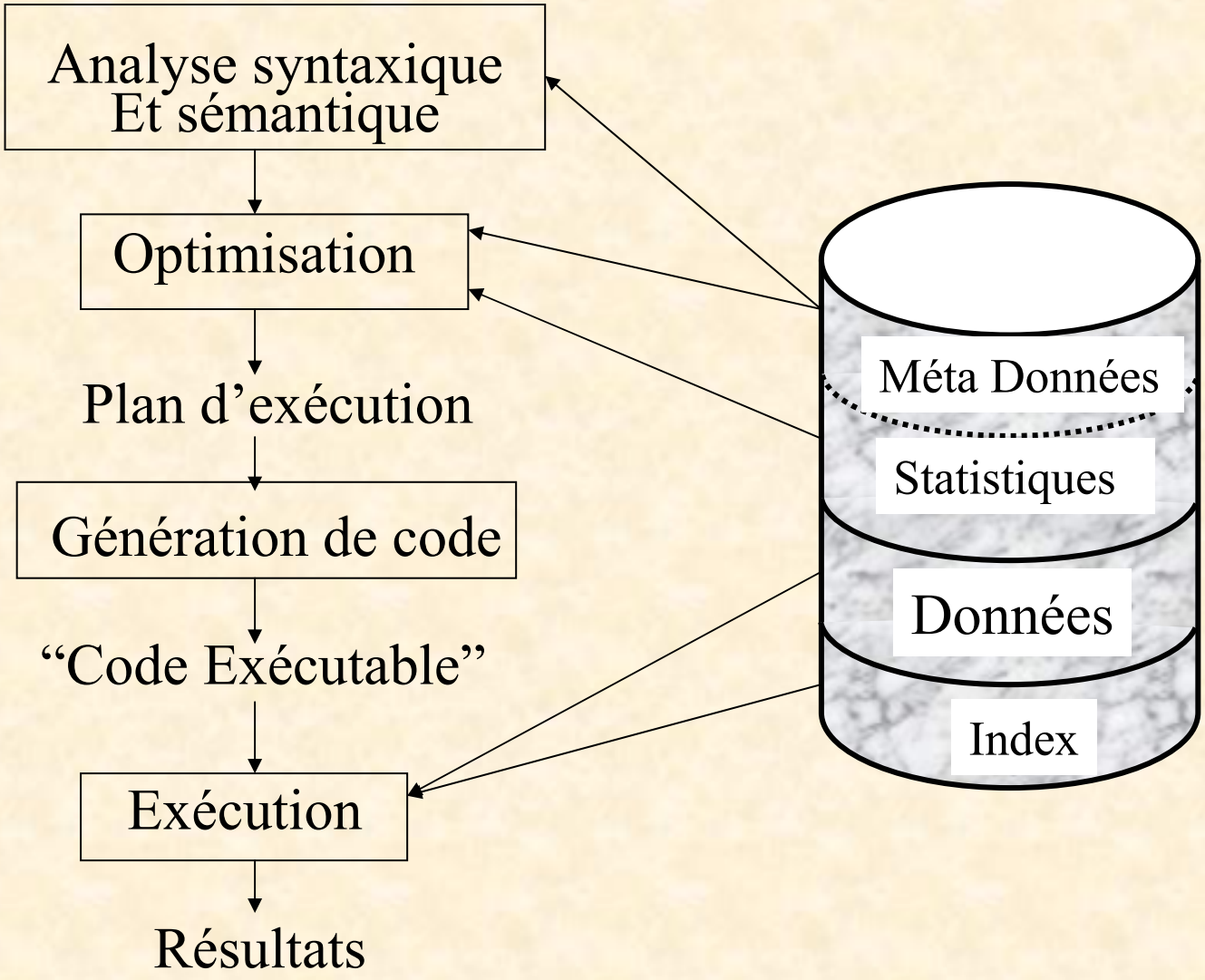
- **R ➲ select * from R**
- **$\Pi_{(A1, \ldots, An)}$ (R) ➲ select A1, …, An from R**
- **σ_C (R) ➲ select * from R where C**
- **R x S ➲ select * from R, S**
- **R $\cup$ S ➲ select * from R union (select * from S)**
- **R \ S ➲ select * from R minus (select * from S)**
- **Join$_{(\theta\text{-expression})}$ (R, S) = $\sigma_{(\theta\text{-expression})}$ (R x S)**

  ➲ select * from R, S where θ-expression

Langages Relationnels (suite)

- Traitement de requêtes:
  - Minimiser le coût d'évaluation
  - Optimisation (Arbre relationnel)
    - Syntaxique
    - Sémantique
    - Estimation de coûts

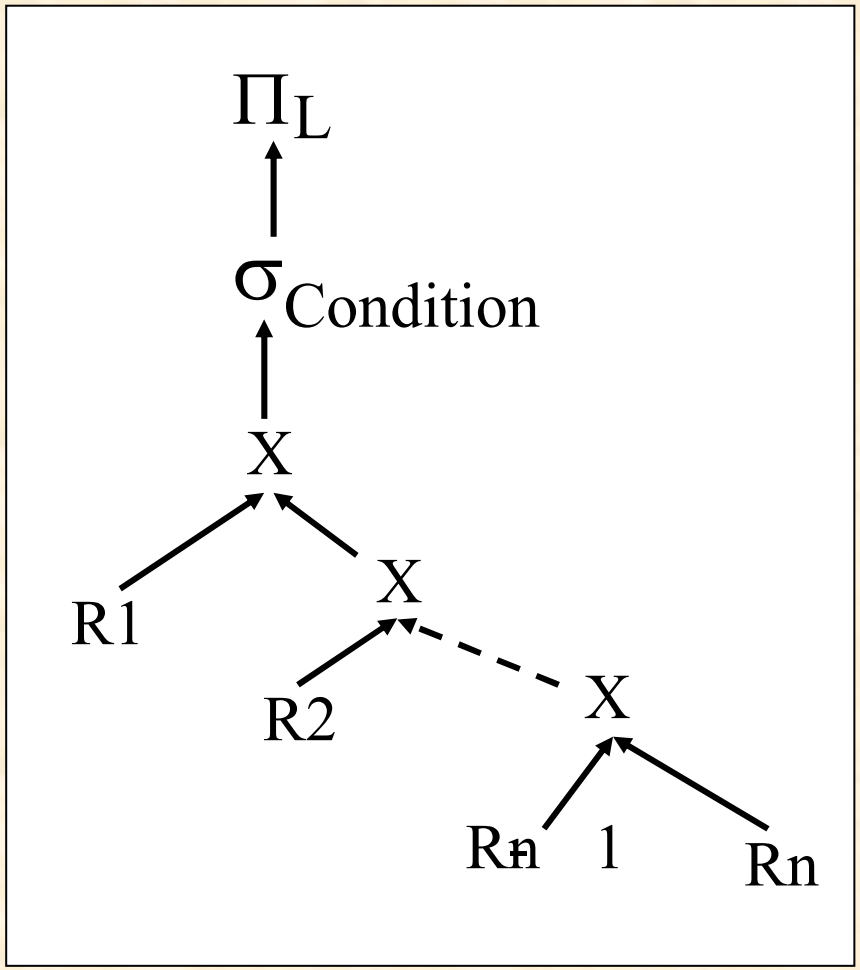
Traitement des Requêtes (suite) : Le processus



Traitement des Requêtes (suite) : Optimisation syntaxique

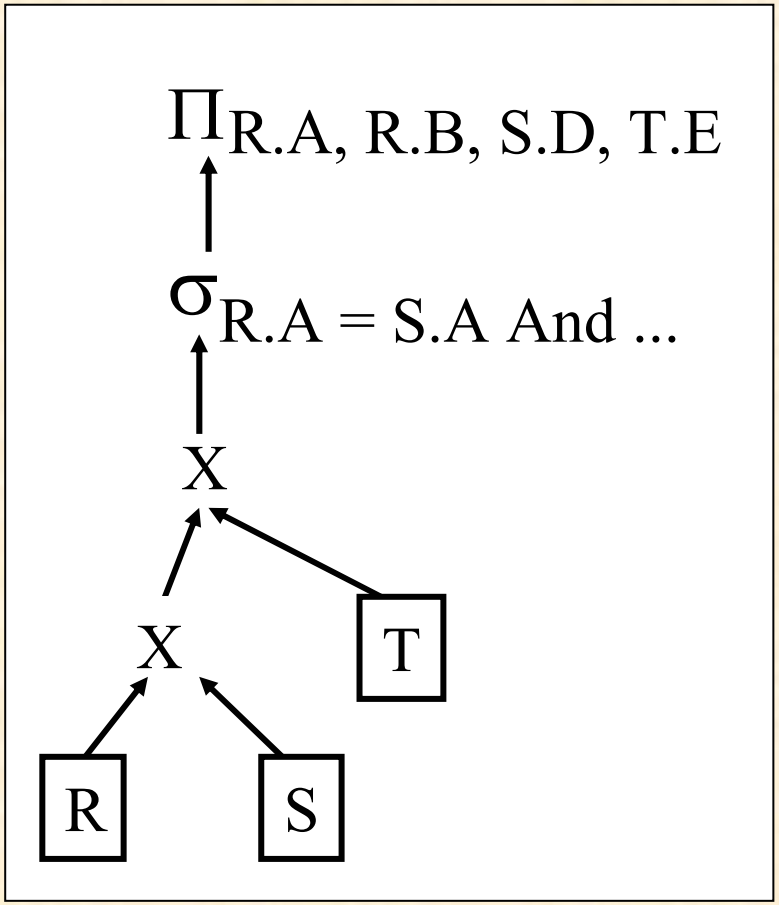
- Requête SQL → Arbre algébrique
- Transformer l'arbre en utilisant
  - Propriétés des opérateurs algébriques (Associativité, Commutativité, Distributivité, etc.)
  - Heuristiques: Evaluer en 1er les opérations les moins coûteuses
    - « Faire descendre » les Sélections (↘ nbr de tuples)
    - « Faire descendre » les Projections (↘ nbr de colonnes)

Select L
From R1, R2, …, Rn
Where Condition

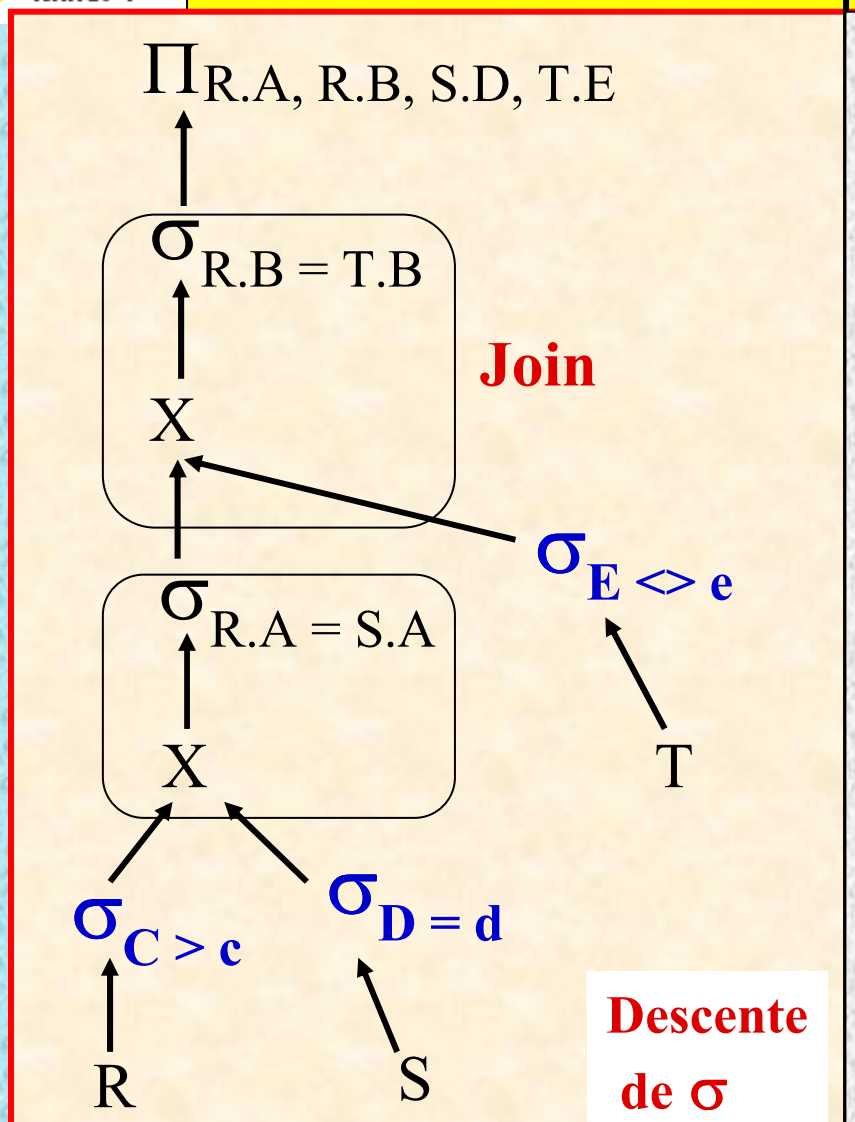


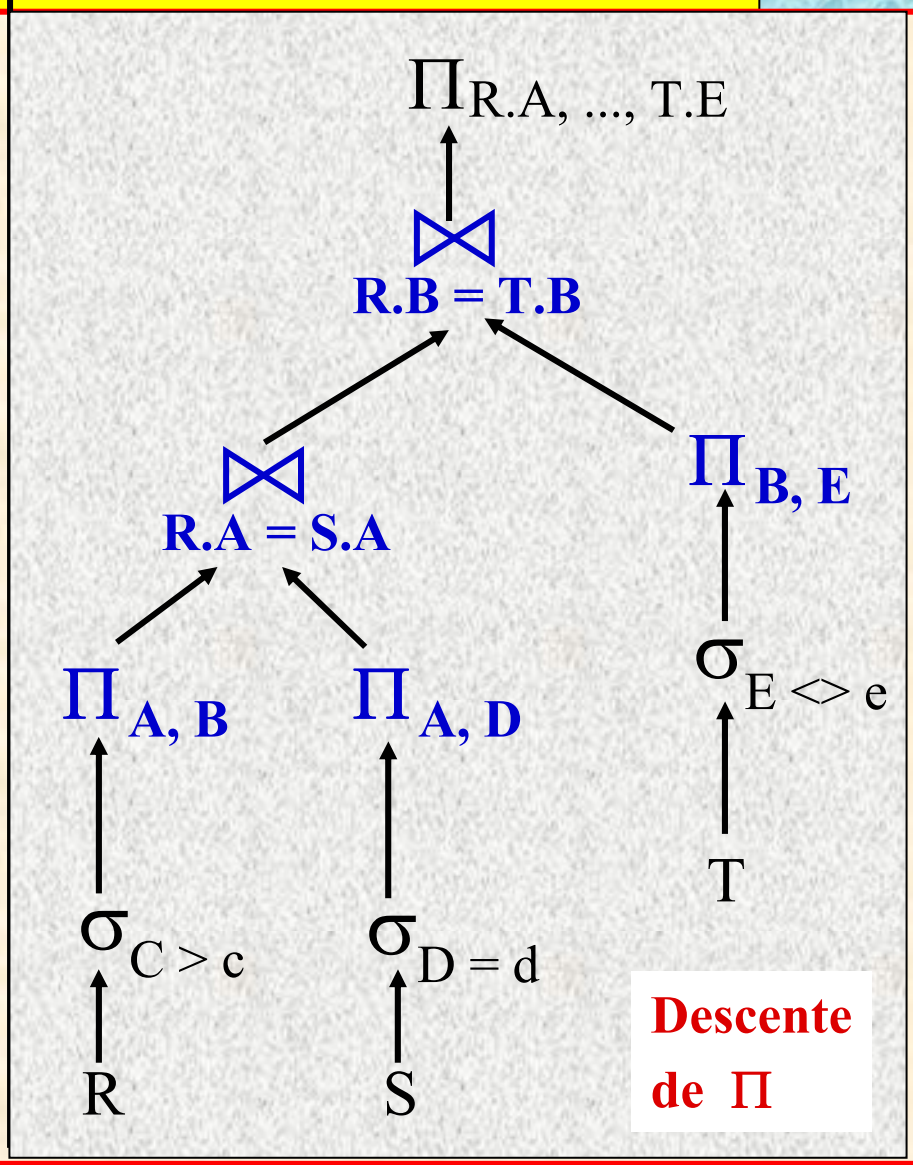
- Relations R(A, B, C, K), S(A, D, G), T(B, E, F)
- Requête

Select R.A, R.B, S.D, T.E
From R, S, T
Where R.A = S.A
And R.B = T. B
And C > c And D = d
And E <> e



Traitement des Requêtes (suite) : Optimisation syntaxique





Traitement des Requêtes (suite) : Bases distribuées

- *SGBD distribué* :

  SGBDs, ***autonomes***, éventuellement ***hétérogènes***, ***reliés par*** un réseau de communication et ***coopérants*** dans l'exécution de tâches.

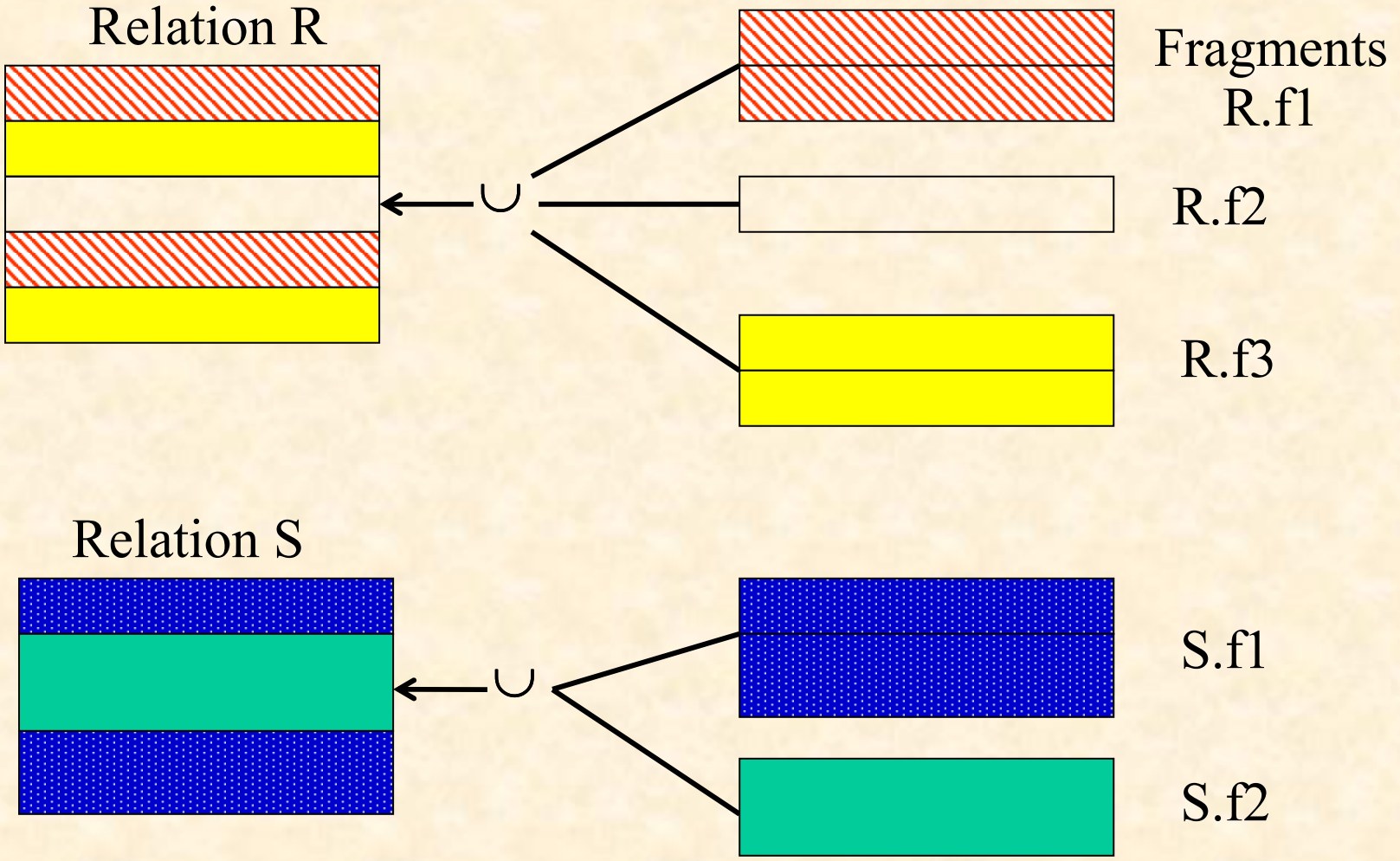
- *Base distribuée*:

  Collection de bases de données ***logiquement reliées*** et ***physiquement distribuées sur un réseau***

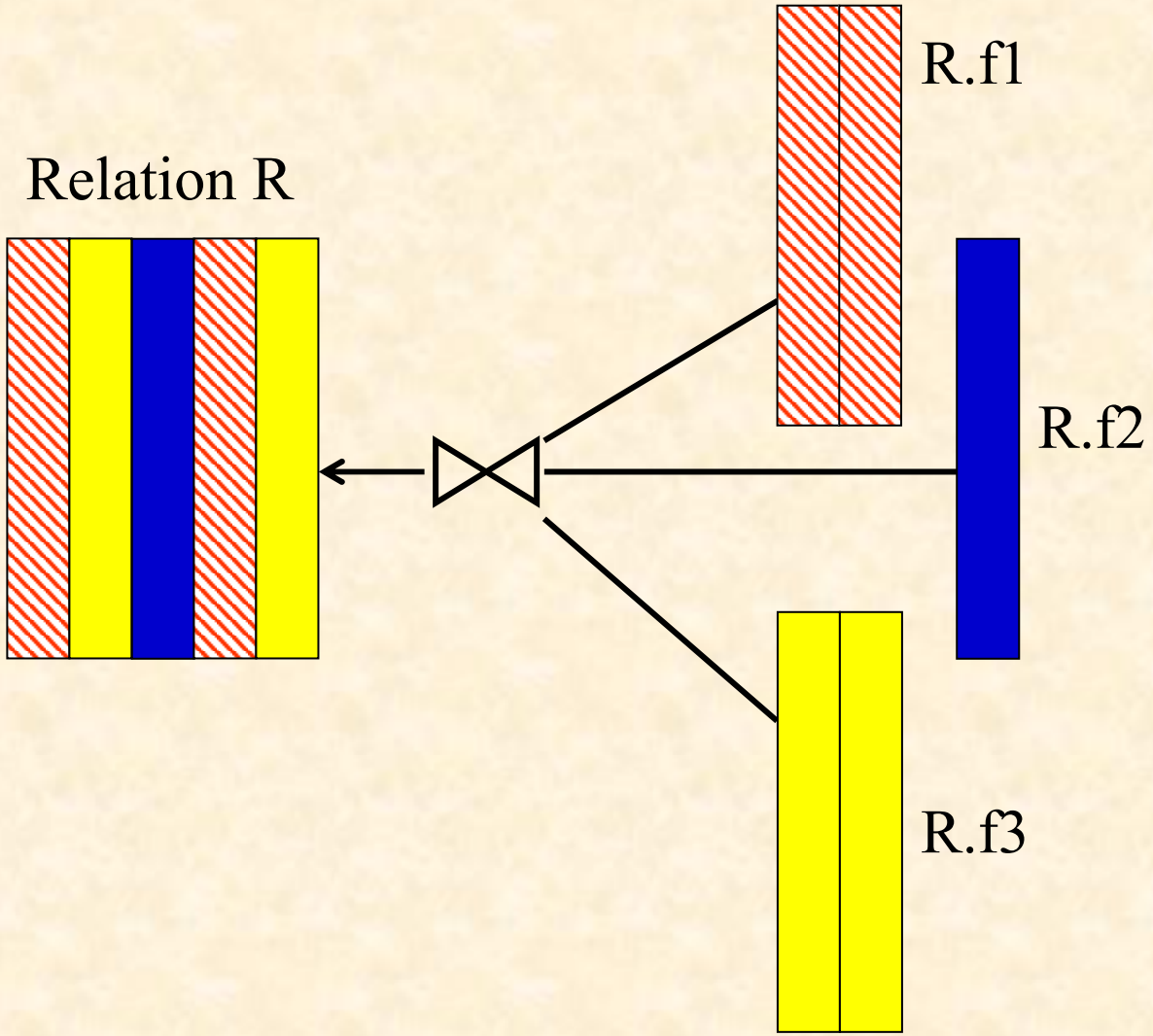
Bases distribuées : Notion de Fragmentation

- "Horizontale/Verticale"
- Avec/Sans duplication
- Duplication totale (cf. Replication Servers)
- Fragmentation et Duplication
  - Duplication d'un fragment
  - Fragmentation de duplicats, etc.

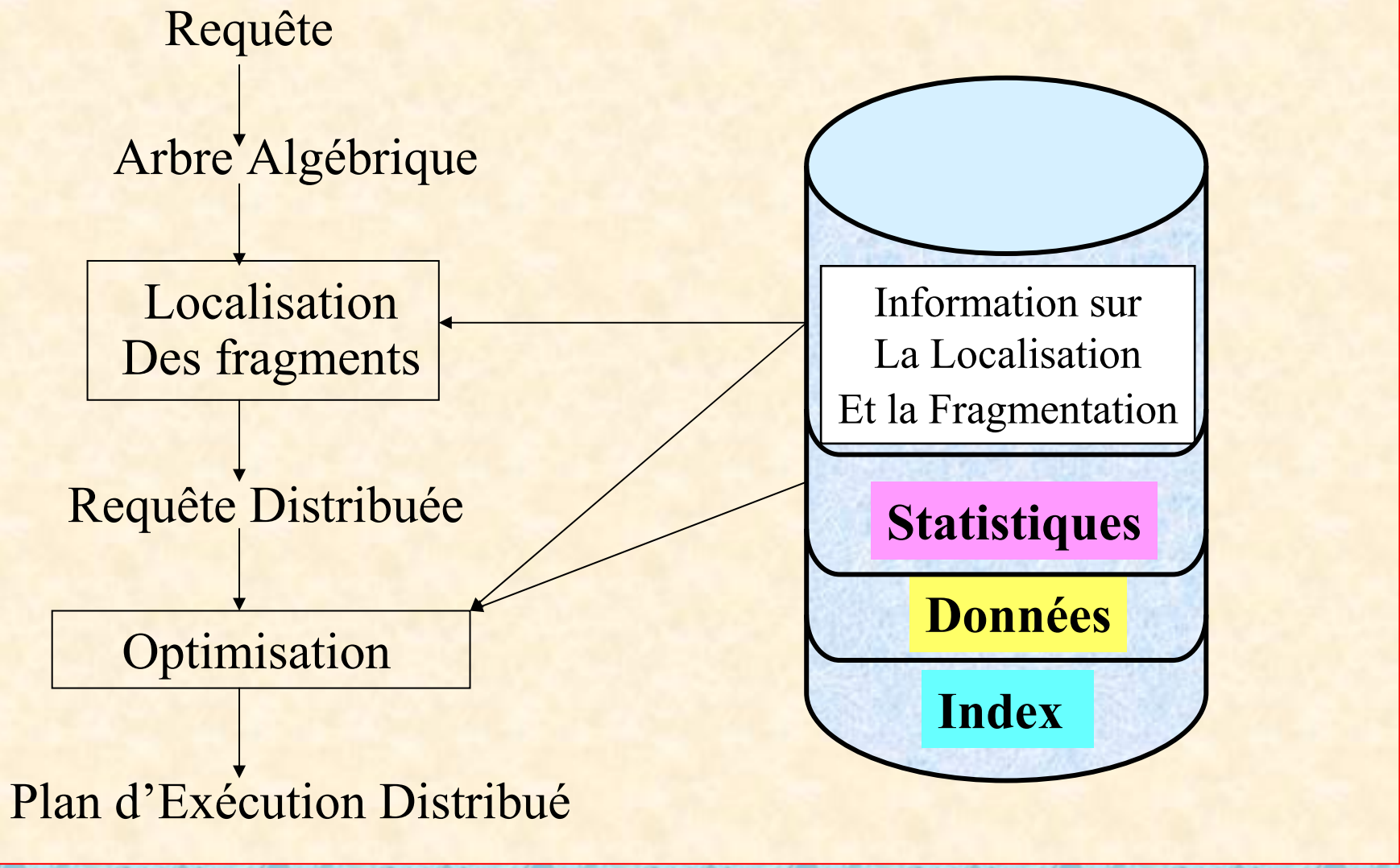
Bases distribuées : Fragmentation Horizontale



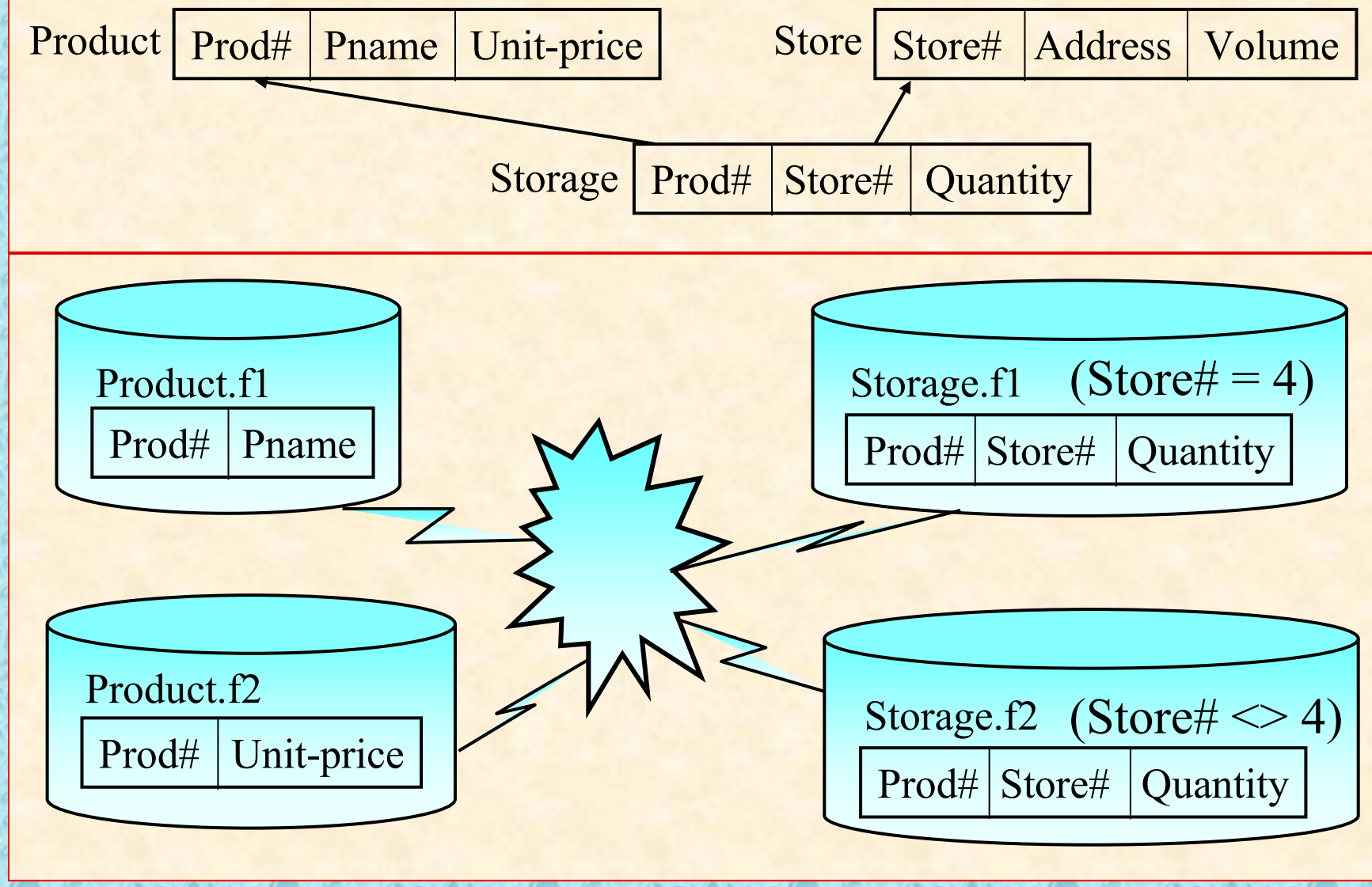
Bases distribuées : Fragmentation Verticale



Bases distribuées : Traitement des Requêtes



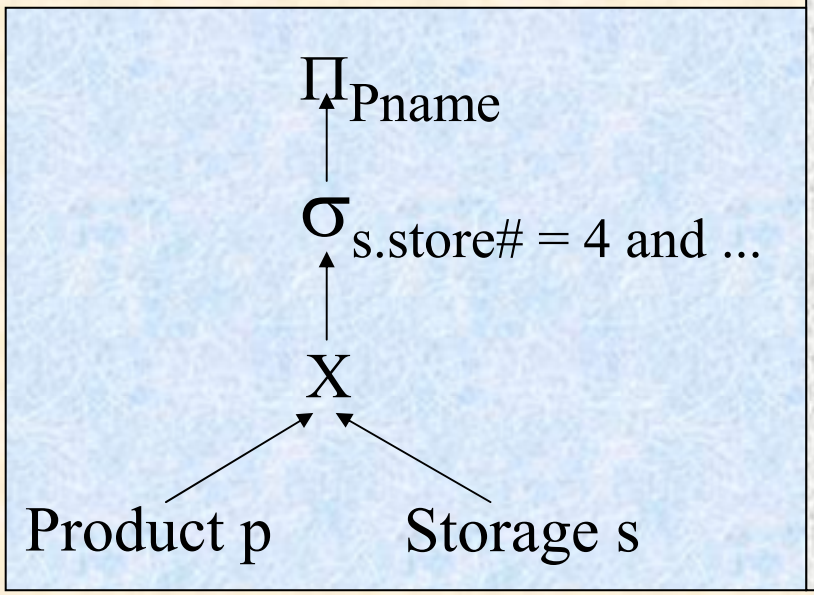
Bases distribuées : Traitement des Requêtes

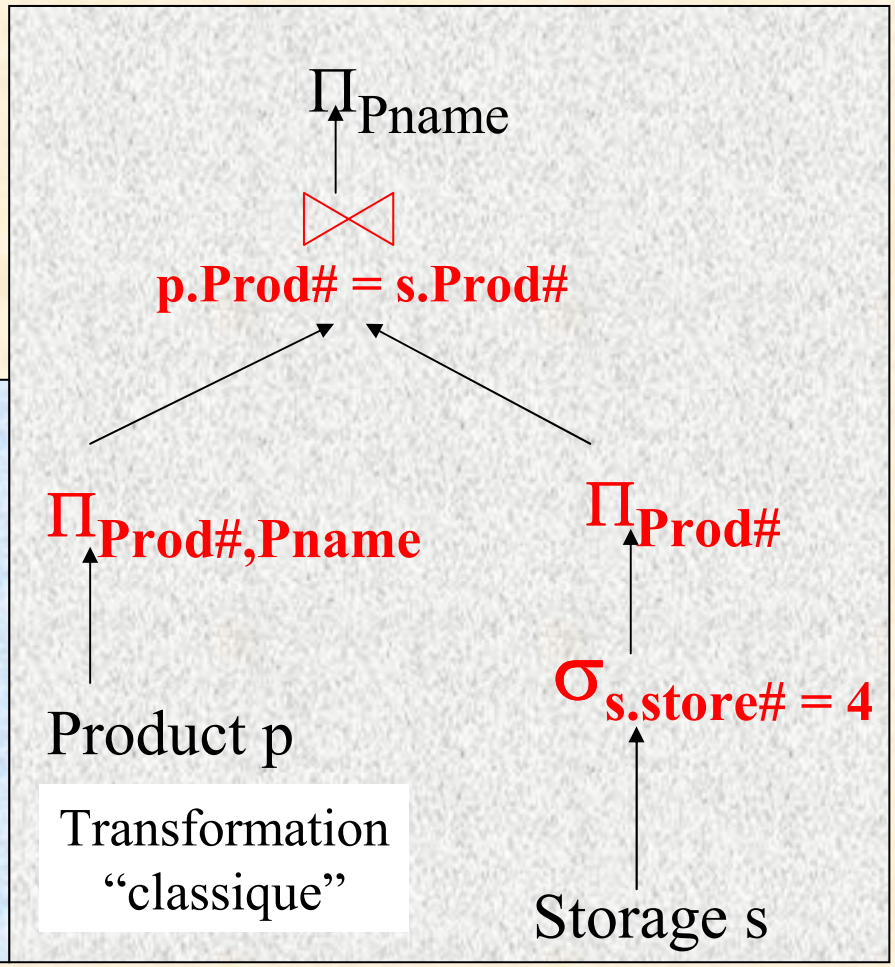


Bases distribuées : Traitement des Requêtes

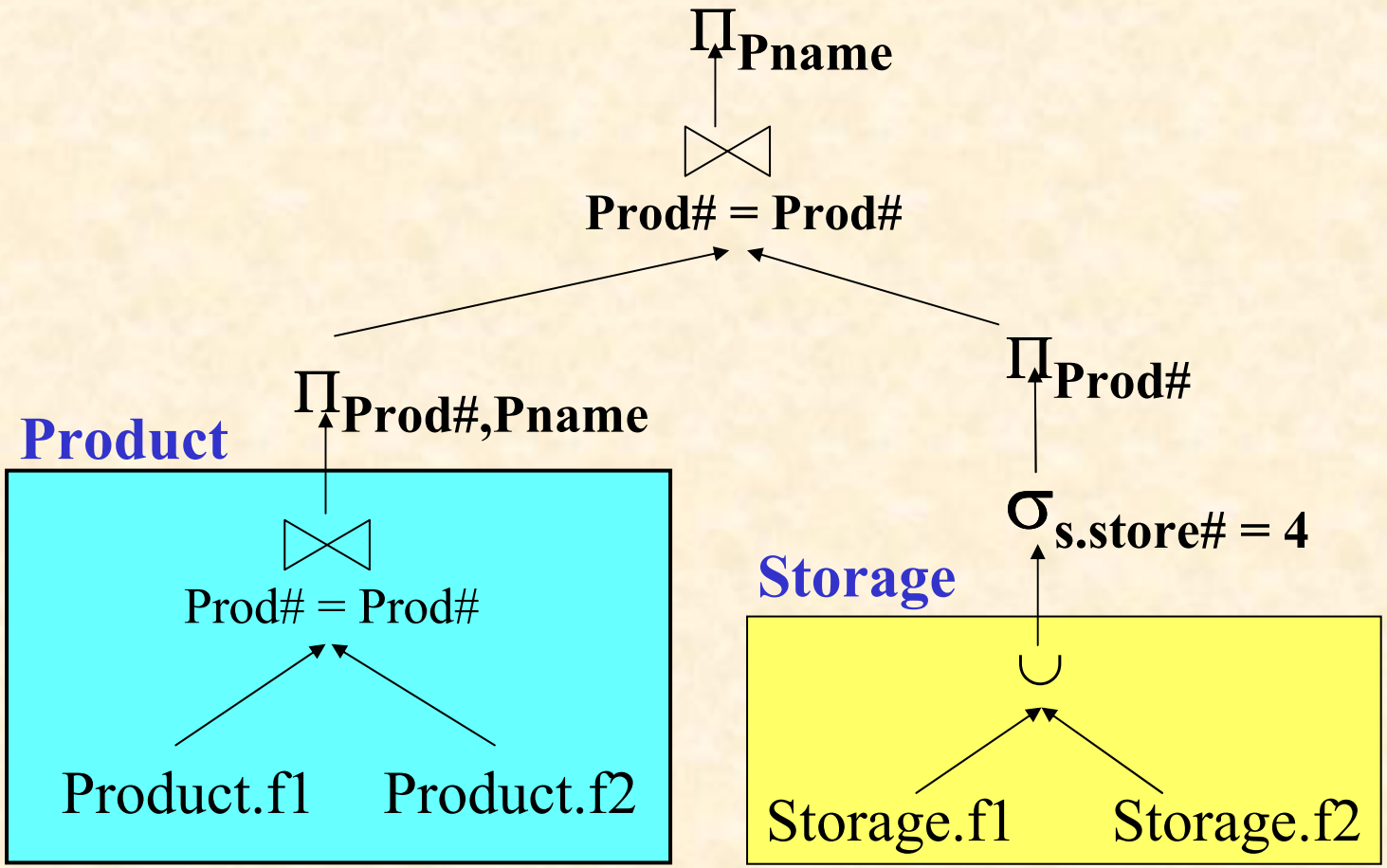
- Requête : Noms des produits du dépôt N° 4 ?

```
Select p.Pname
From Product p, Storage s
Where s.store# = 4
And     p.Prod# = s.Prod#
```

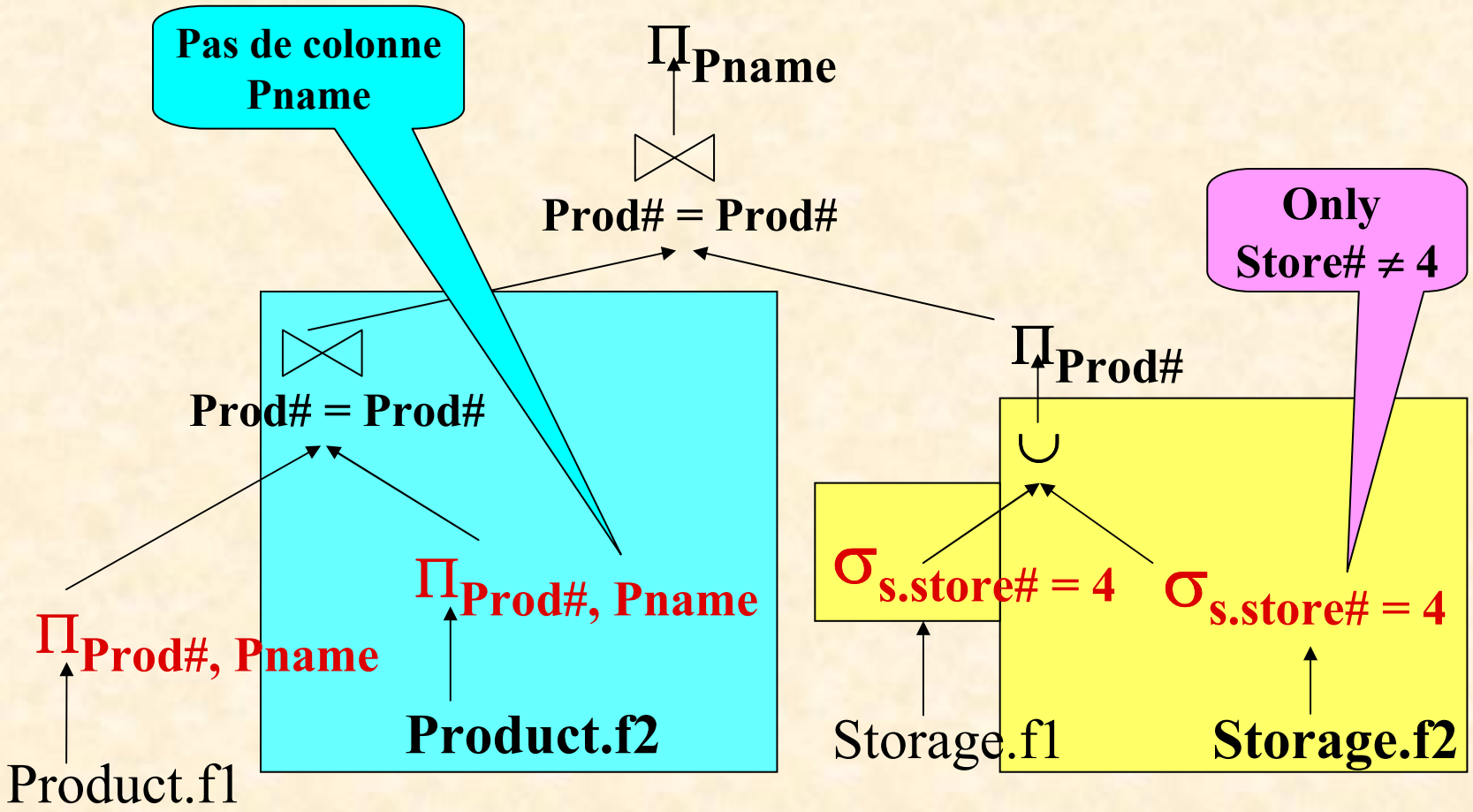




- Seuls les fragments existent ! (⇒ sous-arbres)

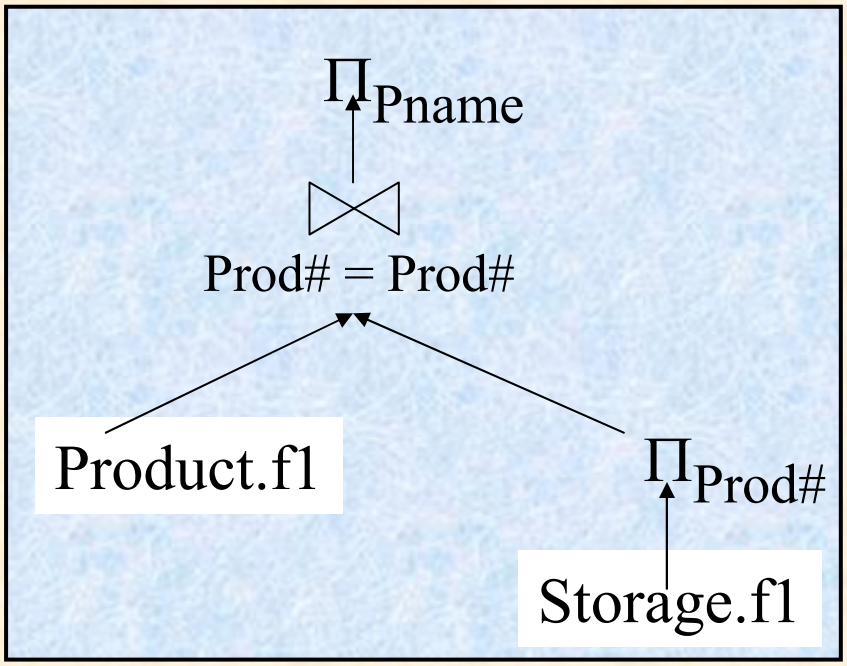


- Descente des sélections et des projections



Bases distribuées : Traitement des Requêtes

- Requête finale :



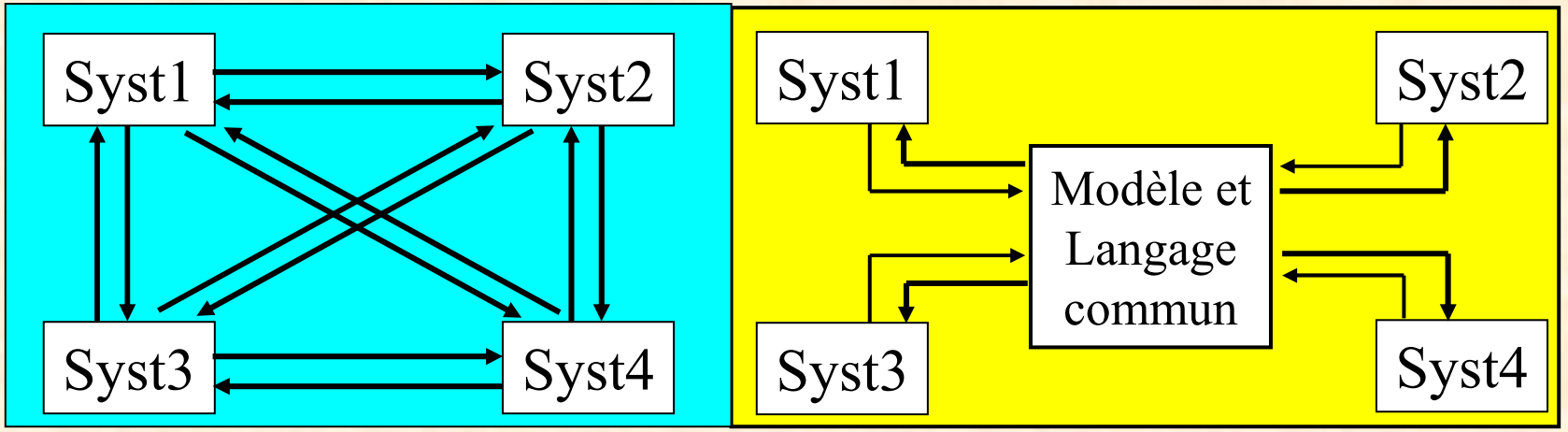
- Accès à 2 sites au lieu de 4 !

Bases distribuées : Traitement des Requêtes

- Choisir une stratégie d'exécution (// joins, semi-join, transfert de données)
- Envoyer les sous-requêtes aux sites impliqués
- Chaque site
  - Application optimisation locale
  - Evaluation
  - Envoi des résultats au site émetteur

Bases distribuées : Traitement des Requêtes (fin)

- Cas SGBD hétérogènes
  - *Transparence* → Requête : langage du SGBD local
  - Traduction dans les langages des SGBD cibles
  - Conversion format des réponses
- Traduction de requêtes; Conversion de données



I.2- Traitement des Requêtes (fin)

- Résultats théoriques et appliqués
- Autres travaux d'intérêt :
  - Inclusion de requêtes
    - $Q1 \subseteq Q2 \Leftrightarrow Repse(Q1) \subseteq Repse(Q2)$
  - Equivalence de requêtes
    - $Q1 \subseteq Q2 \wedge Q2 \subseteq Q1$

I- Conclusion : Forces et Faiblesses du Relationnel

- Ensemble Simple et Restreint de concepts
- Bases Formelles (Algèbre, Logique)
- Indépendance Données-Programmes
- Langages de Haut Niveau
- Résultats Théoriques
- Systèmes Opérationnels

Au-delà de la technologie relationnelle

- Limites:
  - « Faible » modèle de représentation de données
    - Contrainte de 1ère forme normale : Relations « plates »
    - Difficile de modéliser des objets complexes
  - Langages de Manipulation
    - Puissance d'expression (→ Langages Hôtes)
    - Types de données User-defined (cf. SGBD extensibles et Objet-Relationnel)

Au-delà de la technologie relationnelle (suite)

- Nouvelle(s) génération(s)
  - Préserver les acquis
  - Pallier les insuffisances
  - Intégrer
    - Technologies émergentes (réseau, client/serveur, Web)
    - Résultats d'autres domaines (programmation, IA, etc.)
  - Répondre aux besoins de nouvelles applications
    - Activités Non Monotones (e.g. conception)
    - Travail en groupe (Groupware) et Coopération

Post-Relationnel : Quelques tendances

- Connaissances et bases de données
- Système de type extensibles
- Objets complexes et paradigme objet
- Nouvelles technologies: C/S, Internet

Plan de la Présentation : Où sommes-nous?

Partie I : Rapide survol de la technologie relationnelle

Partie II : Modèles et systèmes pour objets complexes

Partie III: Web vs SGBD

II: Modèles et systèmes pour objets complexes

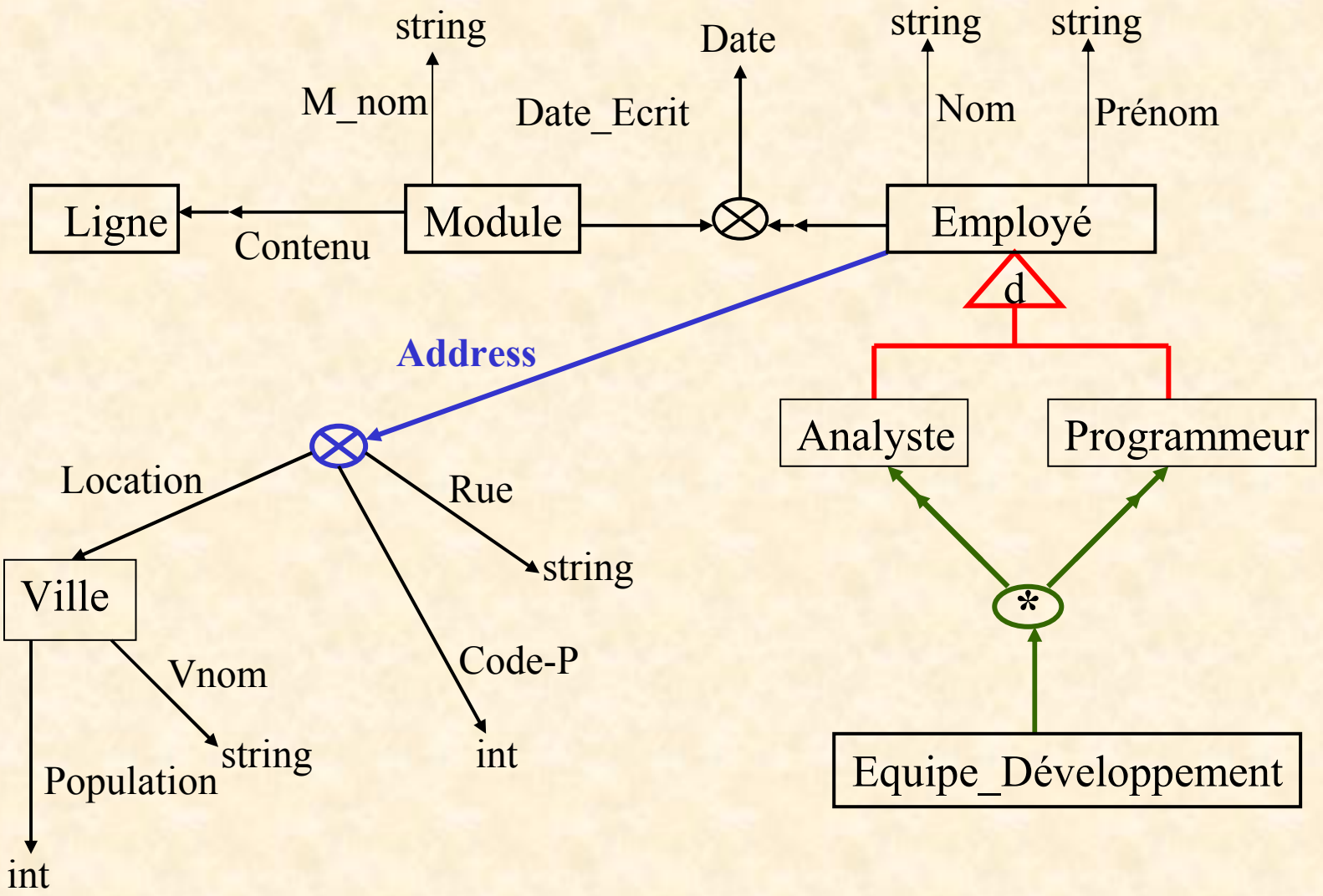
- Approches non « cloisonnées »
  - Logique (DATALOG^{Set})
  - Objets Complexes Structurés :
    - Relationnel : RM/T, NF^2
    - Non relationnels : Modèles Sémantiques
  - Modèles « totalement » Orientés Objets (OMG, ODMG)

- Contenu de la partie II :
  1. Modèles sémantiques
  2. Modèles et systèmes à objets

II.1- Modèles Sémantiques

- Milieu des années 70
- Entity-Relations [Etendu] (Chen)
- Semantic Data Model (Hammer, Mc Leod)
- Functional Data Model
- DAPLEX, etc.
- Objectifs:
  - Etendre le pouvoir d'expression des modèles
  - Réduire/Eliminer l'écart entre le niveau conceptuel et le niveau physique

Modèle sémantique : Exemple



Modèles Sémantiques : Langages

- Schéma Conceptuel = Schéma Physique

- Paradigmes:
  - *Impératif*: TAXIS, Dial, Galileo
  - *Functionnel*: Functional Query Language (FDM)
  - *Relationnel*: GEM, DAPALEX, ARIEL

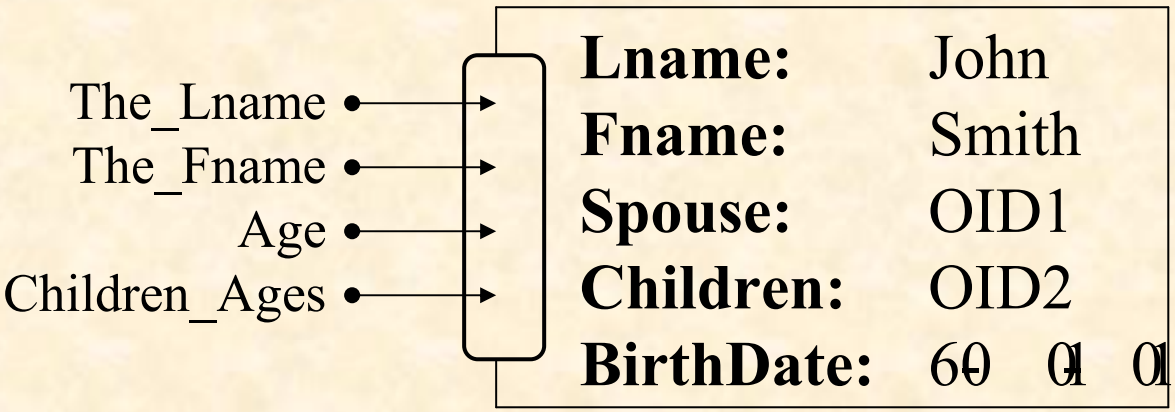
Modèles Sémantiques : Implémentations

- Ré-utiliser l'existant
  - SGBDR + Langage de Programmation Persistant
  - EFDM: Persistent-Algol
  - TAXIS: Pascal-R
  - GEM: Machine Relationnelle
- Développement de mécanismes ad-hoc
  - ADAPLEX: Ada + SGF « classique »

II.1- Modèles Sémantiques : Conclusion

- Pas de fondement formel
- Peu d' implémentations commerciales
- Transition vers les Objets (?)
- Sémantique vs Orienté Objet
  - *Similitude* : Objets complexes imbriqués
  - *Différences* : Modèles et systèmes à objets
    - Comportement des Objets (Opérations/Méthodes)
    - Encapsulation
    - Héritage

- Application du paradigme Objet aux BdD
- Objet =
  - Structure (éventuellement complexe):
    - Identité (OID) indépendante de la valeur
    - Etat privé (Valeurs des attributs): atomiques ou OIDs
  - Interface: ensemble de sélecteur de méthodes



Paradigme Objet et Bases de données : Classes

- Groupement d'objets ayant même structure et même comportement
- ≈ Extension d'un type d'Object :

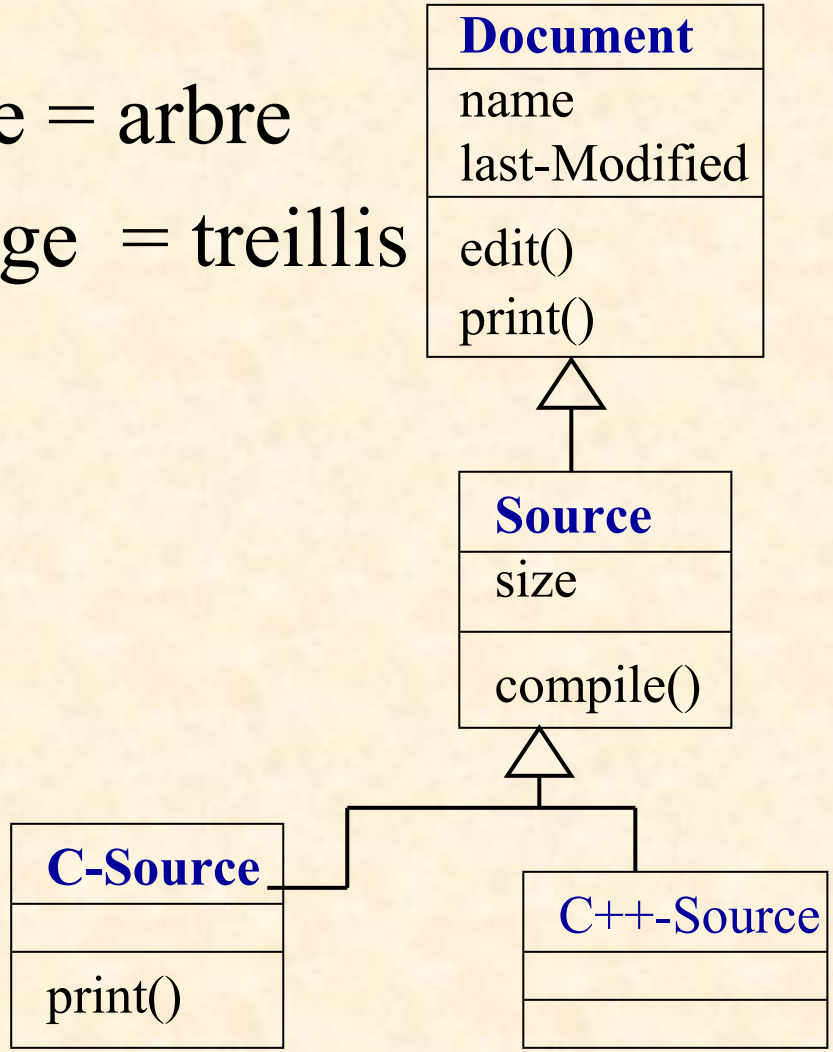
Class Person

Attributes Lname
Spouse
Children
……..

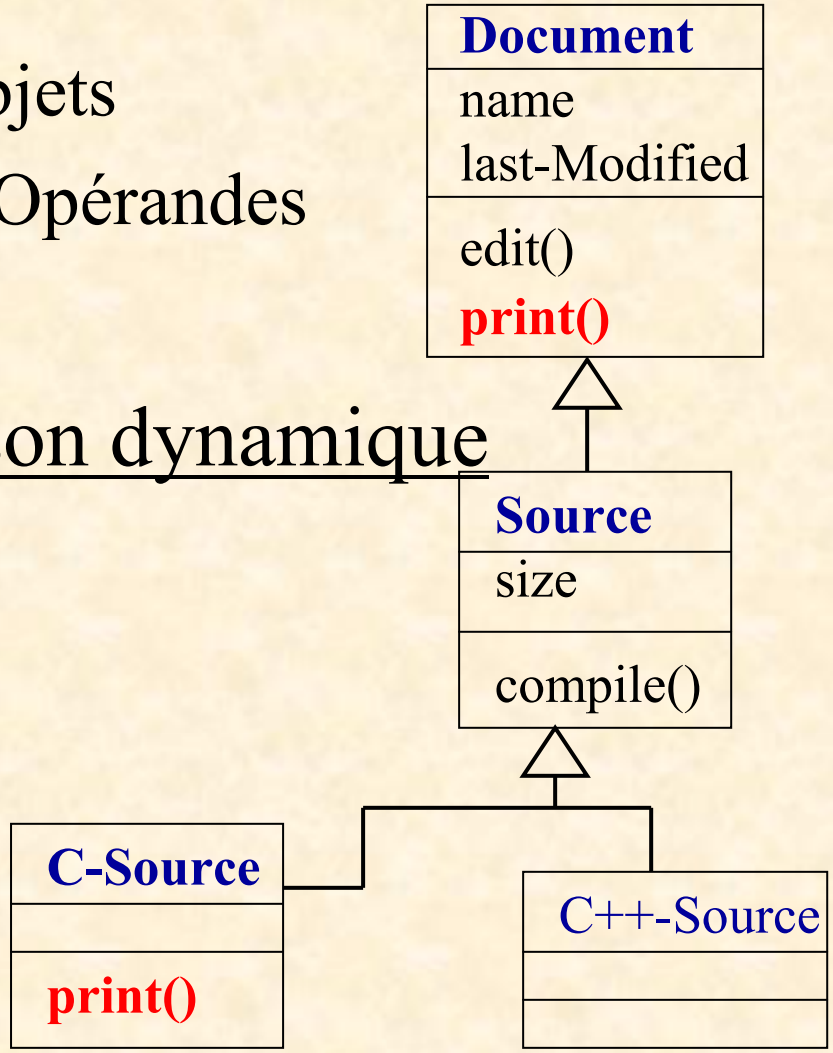
Methods The_Lname() *returns* (self.Lname)
Spouse_Fname *returns* (self.Spouse Lname)

Paradigme Objet et Bases de données : Héritage

- Attributs et Méthodes
- Simple: Graphe héritage = arbre
- Multiple: Graphe héritage = treillis



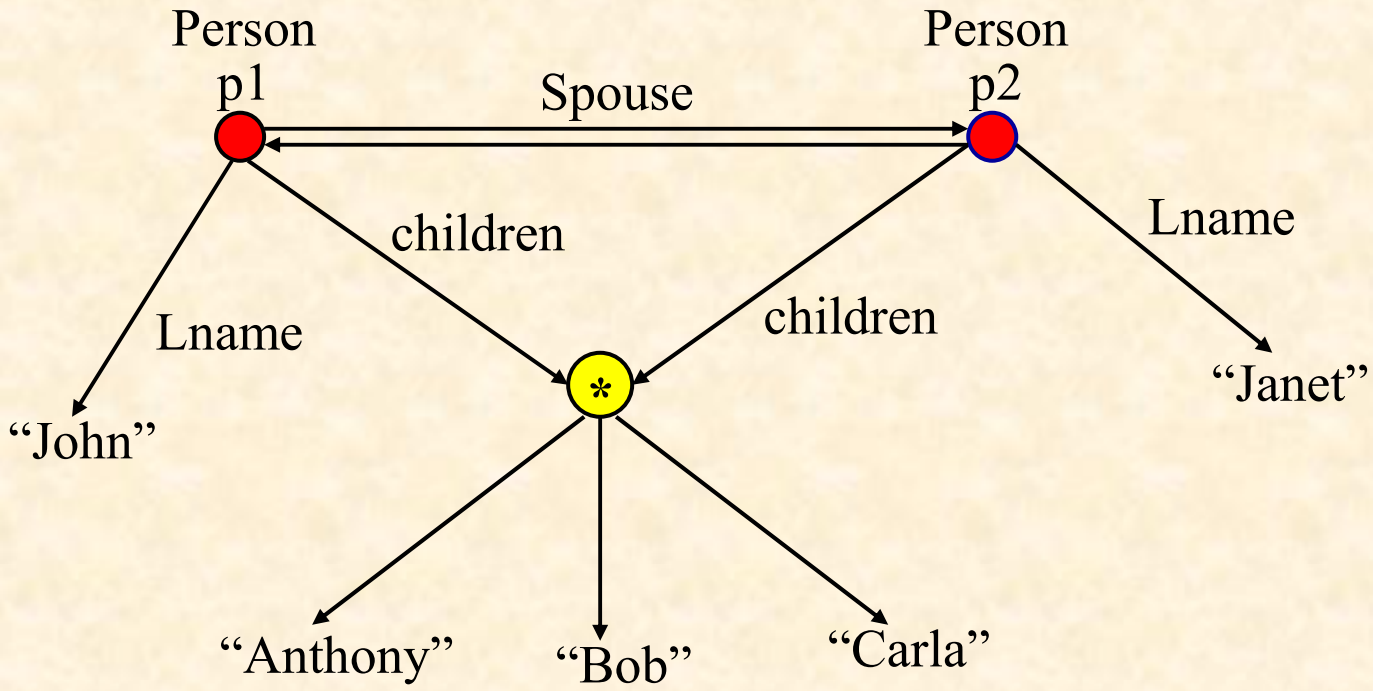
- Passage de Messages
  - Communication entre objets
  - Message ≈ Opération + Opérandes
- Polymorphisme et Liaison dynamique
  - print C-source
  - print C++-source



Paradigme Objet et Bases de données : Egalité

- « Surface » (shallow/value equal)
  - O1: < Name: 206, Manufr: Peugeot, Year: 1999>
  - O2: < Name: 206, Manufr: Peugeot, Year: 1999>
  - Objets Distincts, Egaux en Valeur
- « Profonde » (deep equal)
  - C1: < Name: 206, Manufr: Peugeot, Options: **O1** >
  - C2: < Name: 206, Manufr: Peugeot, Options: **O2** >
  - **O1**: < Type: A1, Value: Air conditioning >
  - **O2**: < Type: A1, Value: Air conditioning >

- Personnes p1 et p2 ont le même ensemble d'enfants



Paradigme Objet et Bases de données (fin)

- Orientation Objet "totale" :
  - Gestion d'Objets Complexes
  - LDD, LMD
  - Sécurité, Intégrité, Confidentialité
  - Accès Concurrents
  - Reprise, etc. $\Rightarrow$ OODBMS

II.2- SGBDO

1. Langage de Manipulation
2. Représentation des objets
3. Indexation et Accès aux objets
4. Propositions de l'ODMG
5. [+ Concurrence, reprise, …]

II.2.1: Langages de Manipulation

- Stratégies de Développement :
  - Nouveaux modèle et langage : Galileo
  - Extension langage ou système: OSQL, Postgres
  - Extension langage à objets: Gemstone
  - Bibliothèques pour SGBDO : Starburst
  - Immersion dans un langage hôte : C++, O2
- Pas de consensus: → Standard (ODMG) ...

II.2.2: Représentation des Objets

- Objets et leur description
- Taille des objets
  - Grands ensembles
  - Objets longs
  - Atomes de grande taille (e.g. image, son)
  - Données graphiques, etc.
- Imbrication d'objets
- Index pour objets complexes ?

II.2.2: Représentation des Objets

1/3. Directe : objet || contenus || descendants

(+) Pas de jointure pour la valeur profonde

(-) Redondance ou références vers objets partagés

2/3. « 3ème forme normale » : Aplatir l'objet

(+) Pas de redondance

(-) Jointures

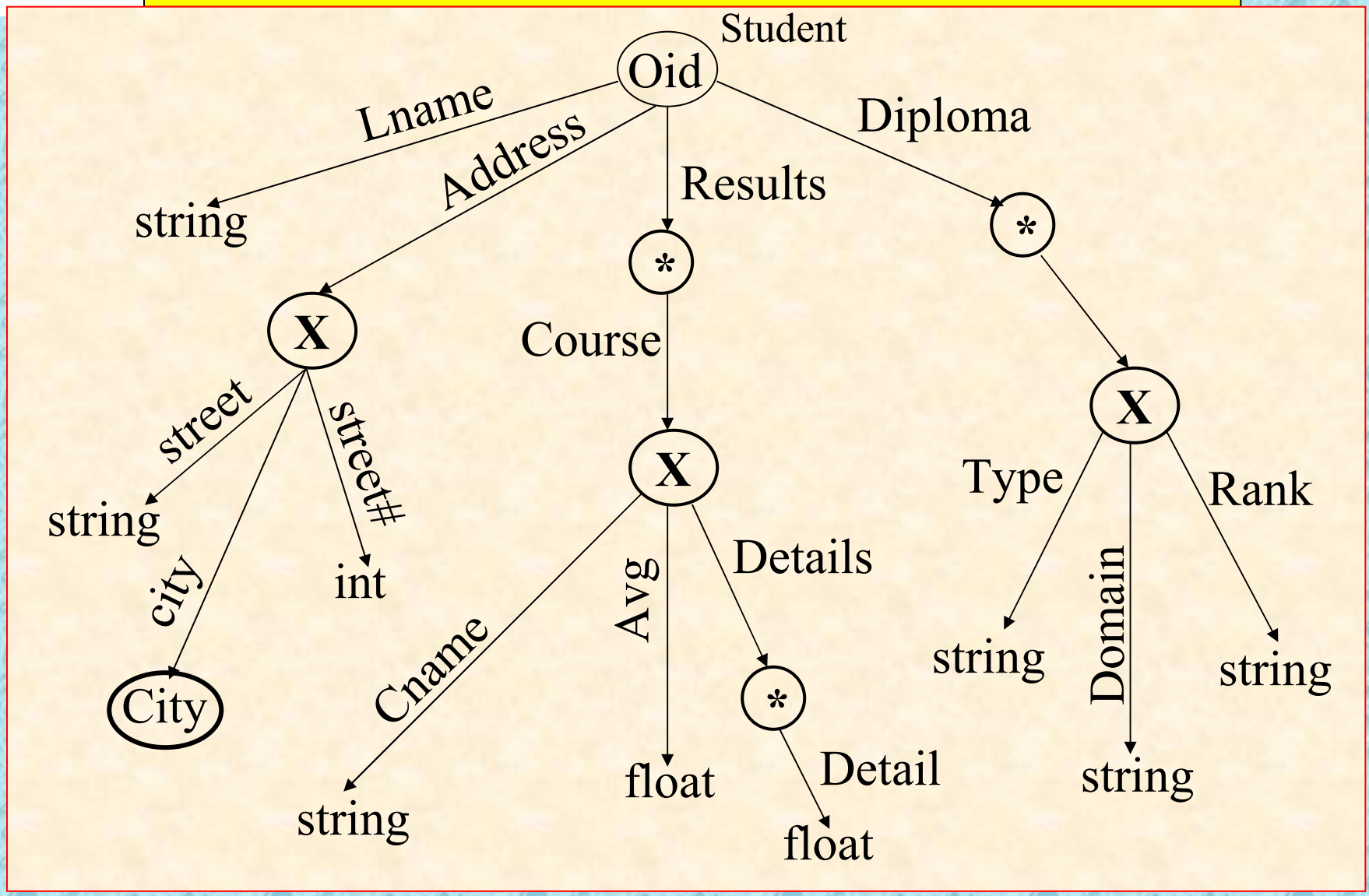
3/3. Multi-Graphe

- Feuilles : Valeurs ou Oids
- Nœuds internes : Constructeurs de types
- Racine : Object ID
- Arcs labellés par les noms d'attributs

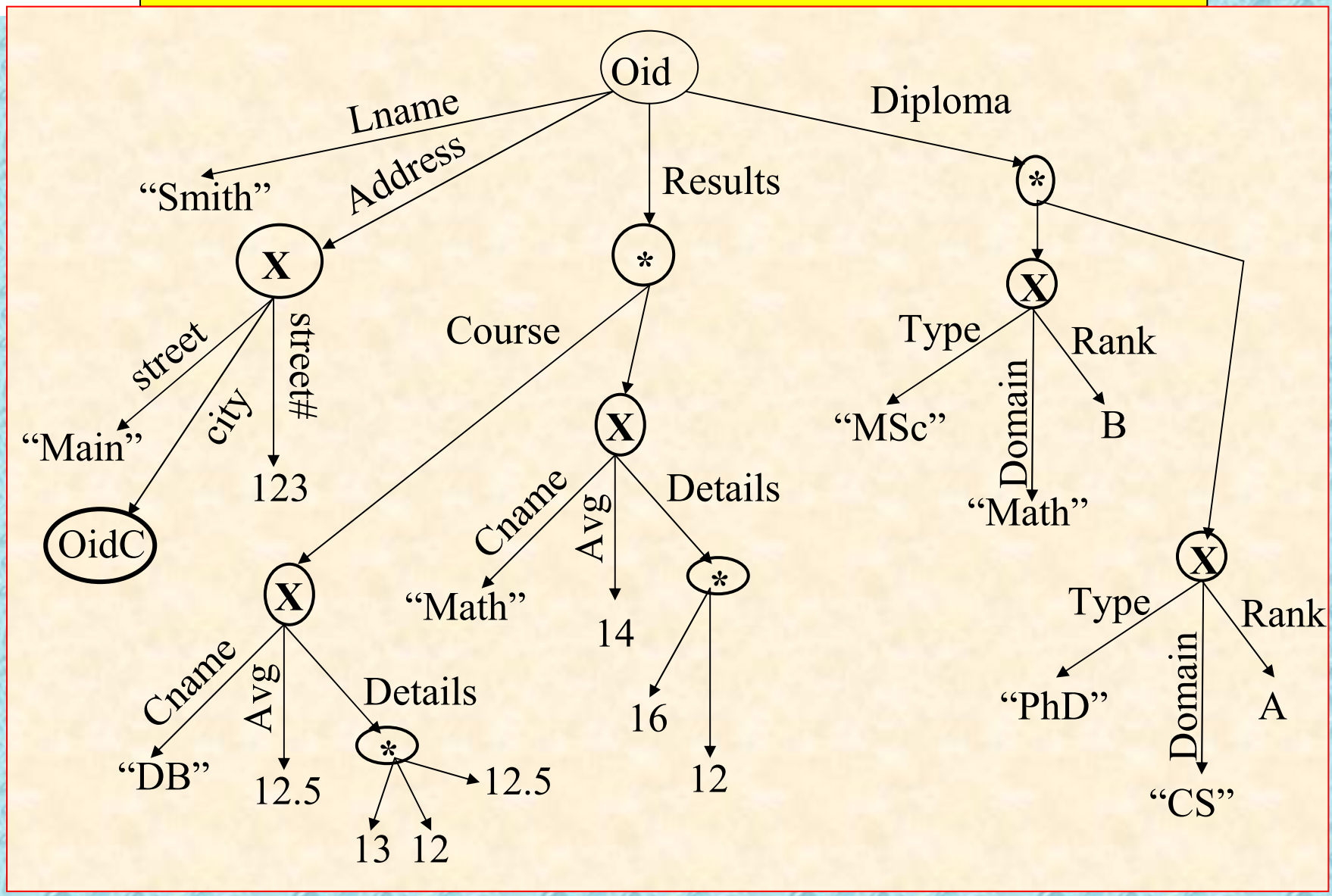
(+) Représentation unique : schéma, instance

(-) Transformation Graphe ↔ « Record »

II.2.2: Représentation des Objets

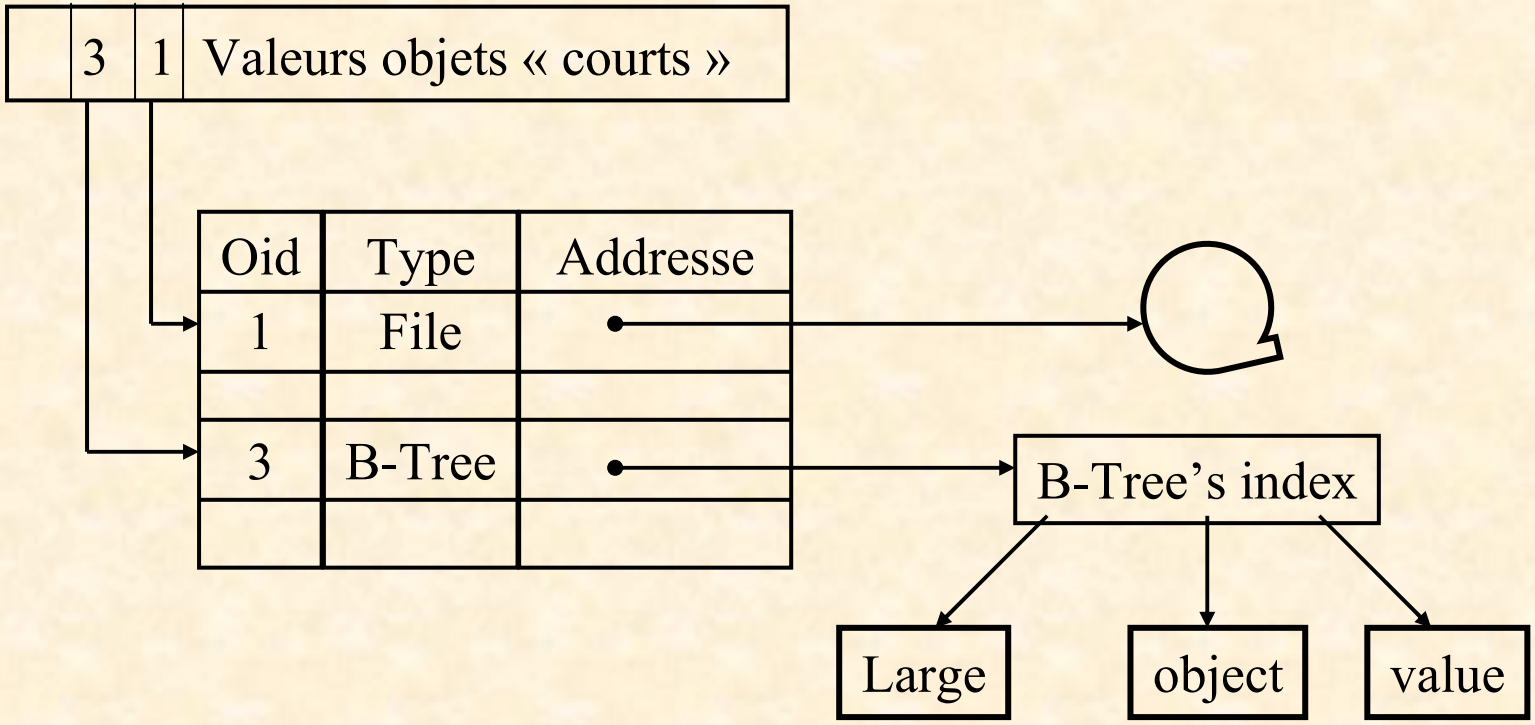


II.2.2: Représentation des Objets



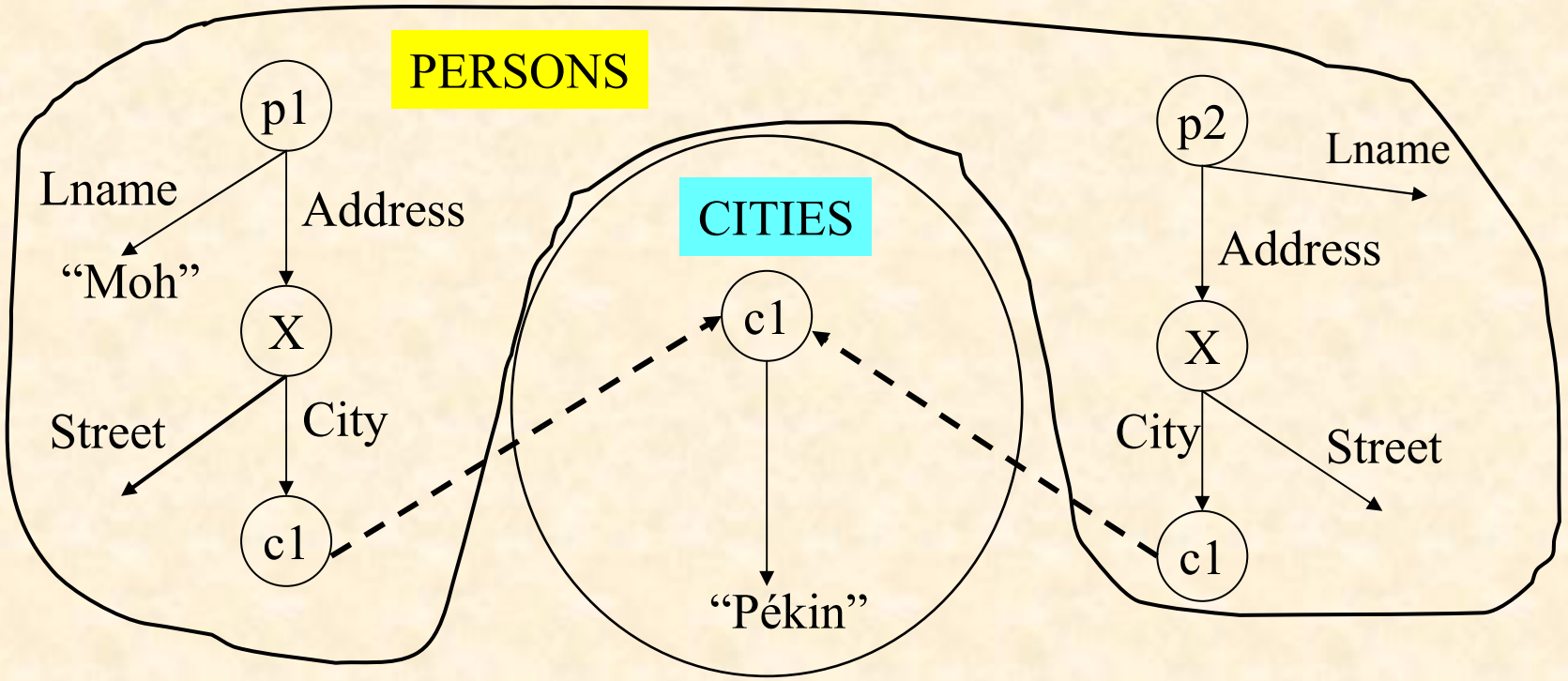
II.2.2: Représentation des Objets Longs

- Objets de petite taille : Valeur immédiate
- Objets longs : « par référence »



II.2.2: Représentation des Objets et Index

- Index composé : Pékin → {p1, p2}
- Index de chemins (traversée de classes)
  - Pékin → {p1, p2}.Address.c1.Name.”Pékin”



II.2.3- Accès aux objets

- Valeur d'un objet :
  - Valeurs des attributs de 1er niveau ?
  - Valeur profonde incluant attributs des sous classes ?
- Lire objet : Représentation base → mémoire
- Ecrire objet : Représentation mémoire → base
- Objets persistants
  - Atteignables partant de racine(s) de persistance (e.g. O2, Opal)
  - Localisés dans un espace de persistance
  - Instances Persistantes (e.g. ObjectStore), etc.

II.2.3- Propositions de l'ODMG

- Object Database Management Group (1991) : Définition de standard pour l'interopérabilité de SGBDO (http://www.odmg.org)
- ObjectDesign, Ontos, Objectivity, HP, Poet Software, Digital Equipment, O2, etc.
- Approche Langage de programmation BdD i.e. extension d'un langage avec :
  - Transparence % persistence
  - Accès Concurrents, etc.
- Interfaces pour stocker des données dans des bases relationnelles

Propositions de l'ODMG

- Composants Standards :
  - Object Model: basé sur le modèle de l'OMG
  - Object Definition Language: basé sur IDL (OMG)
  - Object Query Language: SQL'92, quand possible

Skip ODMG

ODMG : Object Query Language

- Etend SQL-92 avec des concepts OO : Objets complexes, OiD, polymorphisme, etc.
- Langage Clos % Modèle d'objets
- Déclaratif; Fonctionnel; Requêtes nommées
- Peut être immergé dans un langage « bindé »
- Peut invoquer des opérations de ces langages
- Pas d'opérations explicites de modification (opérations des objets)

II.2- SGBDO (fin)

- Produits Commercialisés
- Support de persistance (back-ends)
- Tendance :
  - Pas de migration vers les SGBDO
  - Mais de l'objet « dans » le relationnel

Plan de la Présentation : Où en sommes-nous?

Partie I : Survol du relationnel

Partie II : Modèles et systèmes pour objets complexes 

Partie III: SGBD vs Web 

Partie III : SGBD vs Web

1. Notions de Client/Serveur
2. Typologie des serveurs
3. Web et Bases de Données
    1. Le Web en tant que Base de données
    2. Introduction à XML et sa galaxie

III.1 : Notion de Client/Serveur

- Définitions
  - Application Cliente : invoque des services
  - Application Serveur : offre des services
  - *Machine Cliente* : résidence application cliente
  - *Machine Serveur* : résidence application serveur
  - Machine Client et Serveur peut être la même
  - Un client peut aussi être serveur

Notion de Client/Serveur (suite)

- Middleware : la “glue” entre Client et Serveur
- Portée du Middleware:
  - Des APIs, côté client
  - Au retour des réponses, côté serveur
- Classes de Middleware :
  - Général :
    - Piles de Communication, Synchronization Horloges
    - Services d’Authentification, de Nommage, etc.
  - Spécifique (à un service, une architecture)

Notion de Client/Serveur (suite)

- Middleware Spécifique :
  - Base de Données : ODBC, JDBC, DRDA, EDA/SQL
  - Transaction : TxRPC, Transactional RPC (Encina)
  - Objets Distribués : CORBA, OLE/DCOM
  - Internet: HTTP, S(ecure)-HTTP

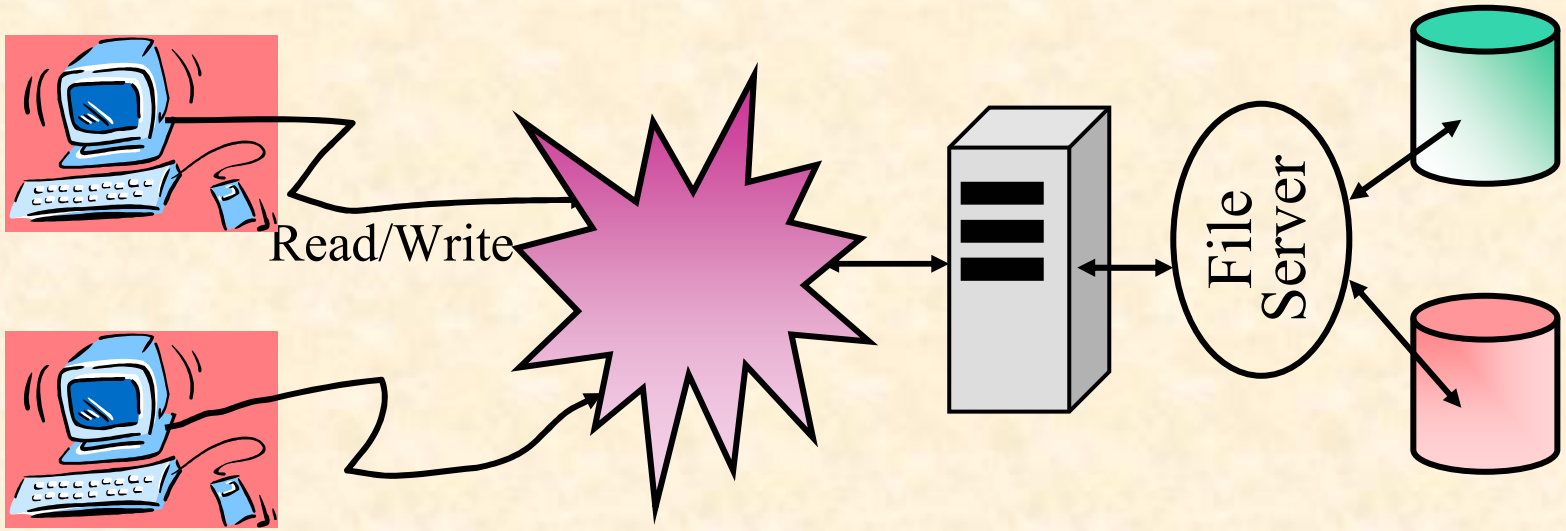
Notion de Client/Serveur (fin)

- Communication Client/Serveur :
  - Synchronisation Requêtes-Réponses
  - Traitement de l'hétérogénéité éventuelle des données
- Mécanismes de Communication :
  - *peer-to-peer* (égal à égal, ≈ sockets)
  - ***R**emote **P**rocedure **C**all*
  - *Message Queues* (**M**essage **O**riented **M**iddleware)
  - ***O**bject **R**equest **B**roker*

III.2- Typologie des Serveurs

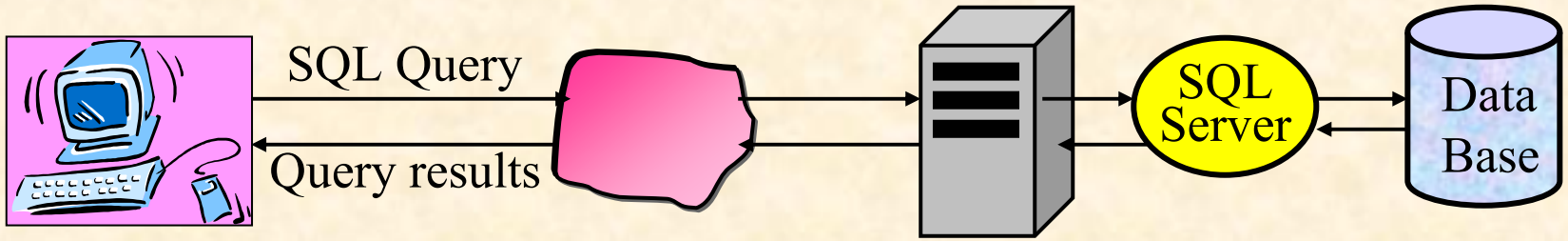
1/7) Serveurs de Fichiers

- Transparence % Localisation → Service de Nommage

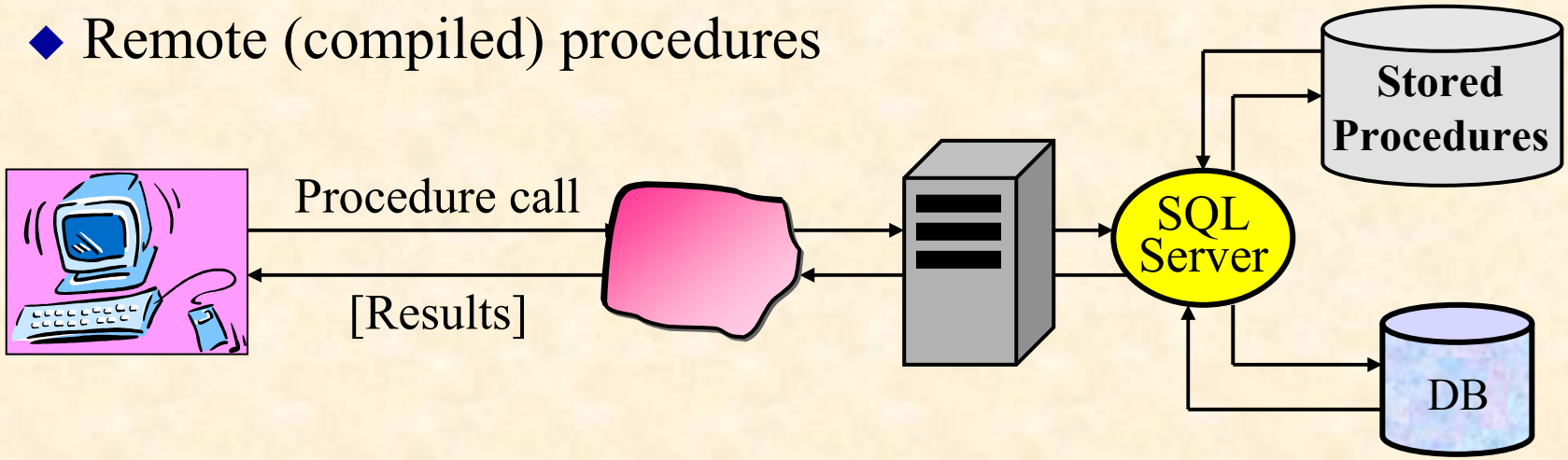


2/7) Serveurs de Données (SQL)

- SQL Queries



- Remote (compiled) procedures

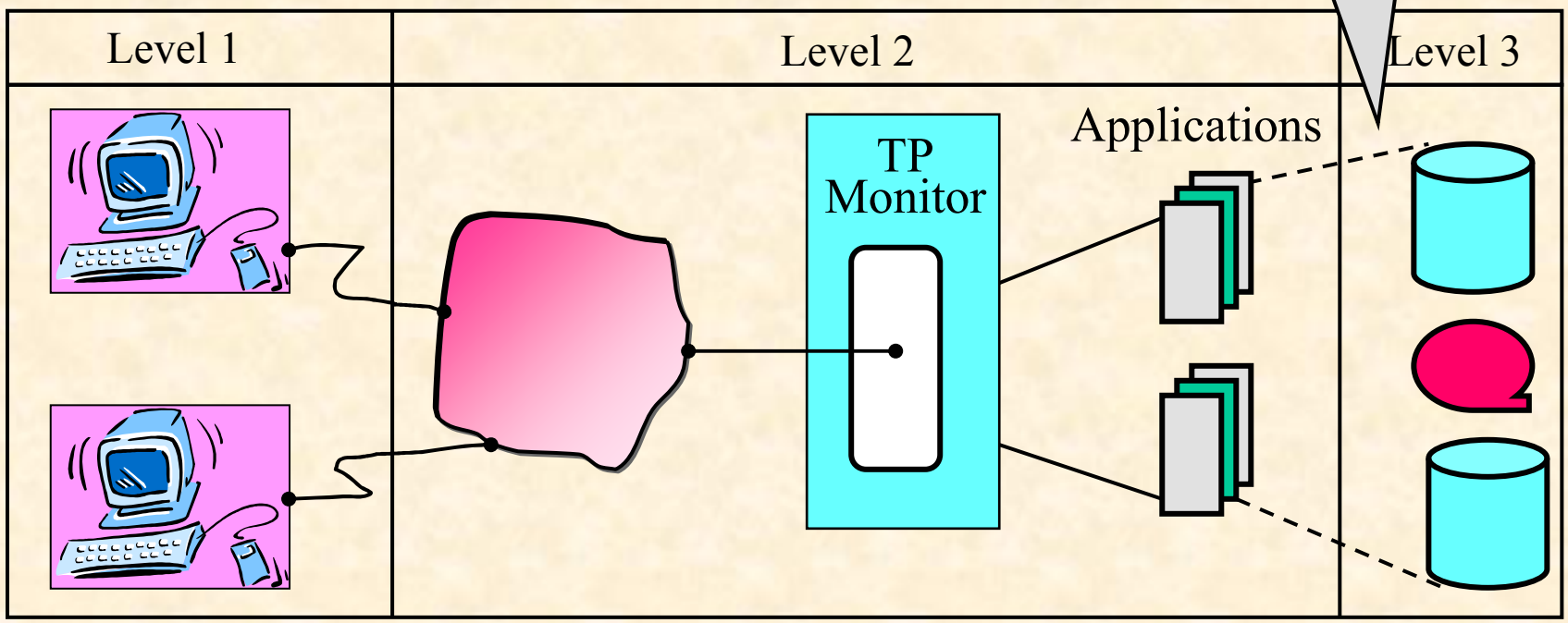


3/7) Serveurs Groupware

- Communication entre personnes
- Données Non structurées : mails, textes, images, etc.
- Exemple : Lotus-Notes, W4, etc.

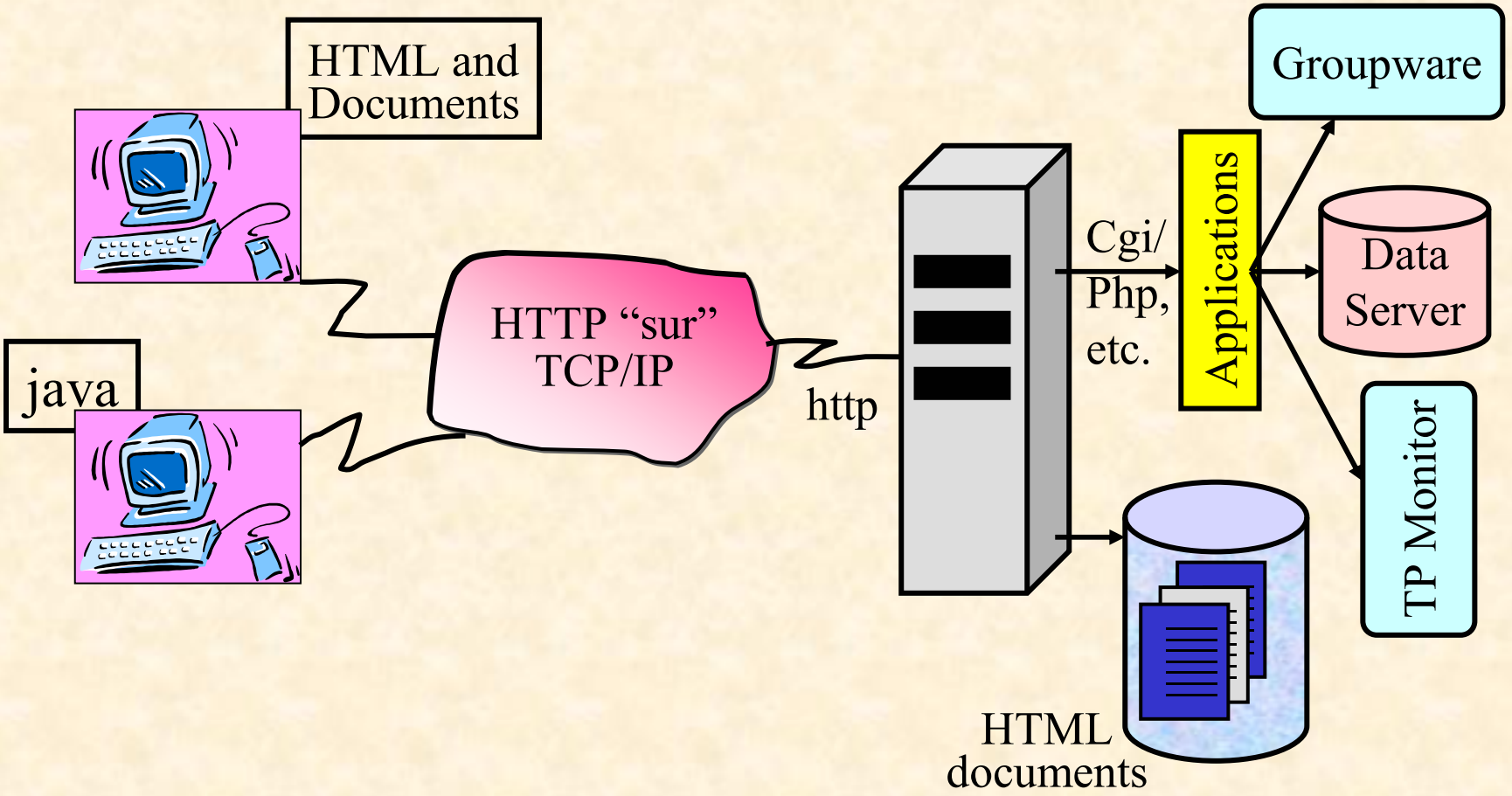
4/7) Serveurs de Transactions : comportent

- Ensemble de transactions ou applic
- Moniteur de Transactions ≈ systèm ion



5/7) Web Serveurs

- Conversion : Relational/Object ↔ HTML (XML)

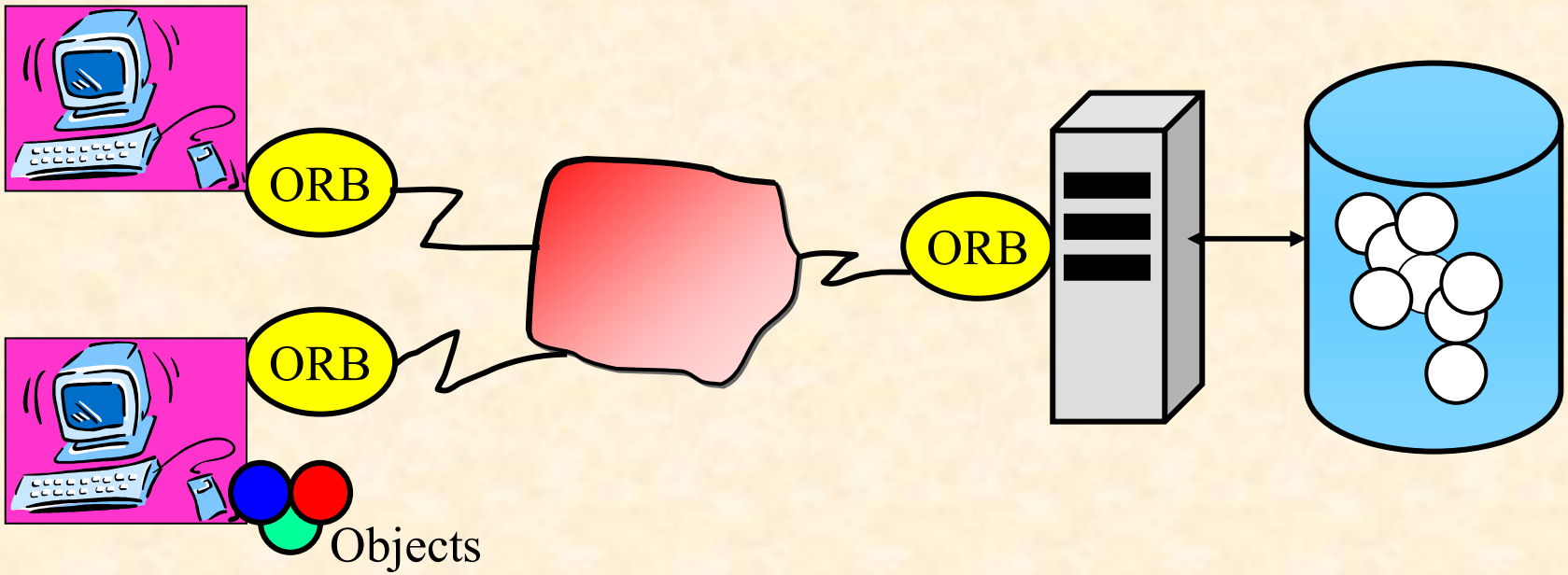


Serveurs Web (suite)

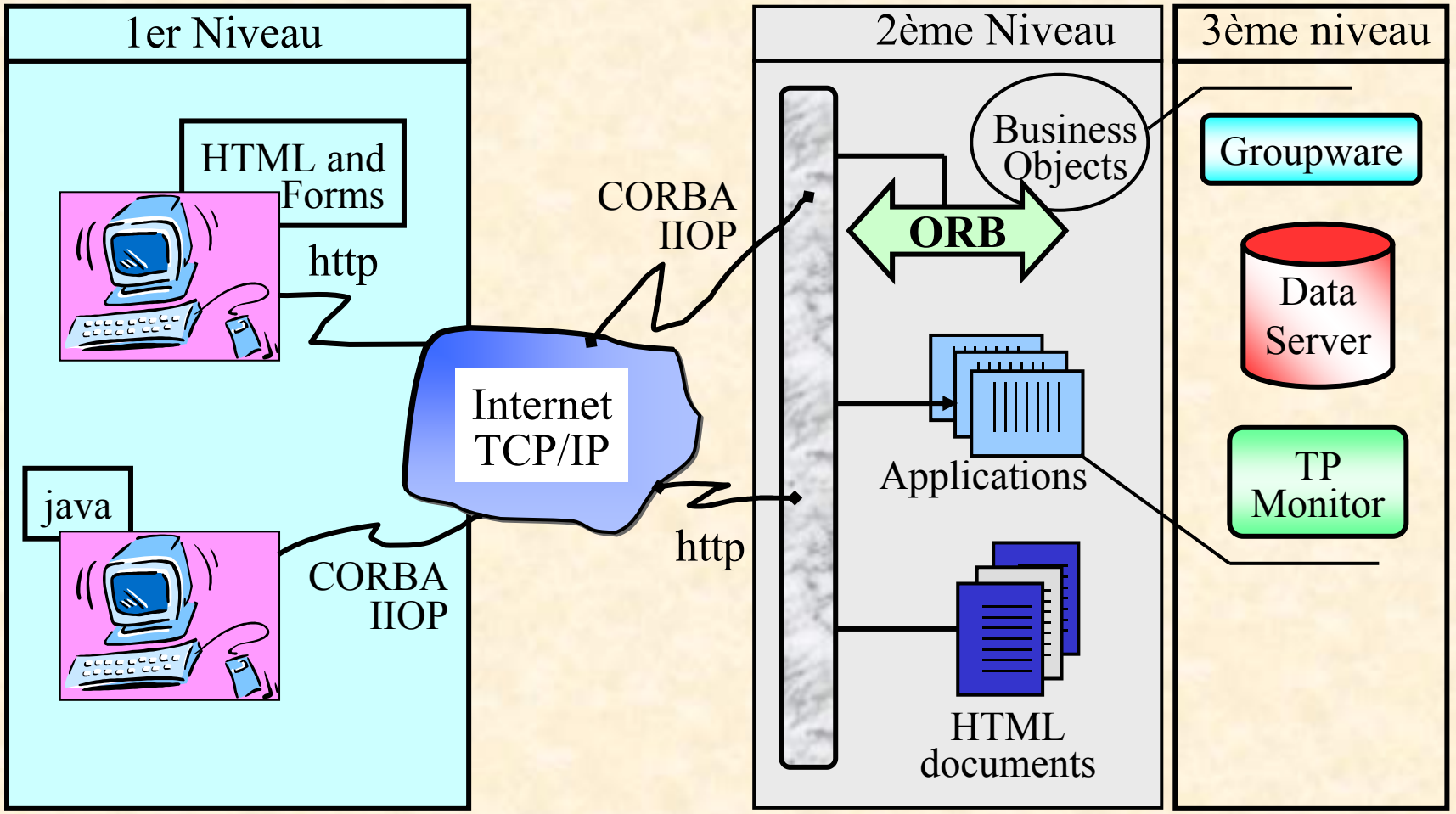
- Unité d'information : fichier (page) adressé par une URL (Uniform Resource Locator)
- HTML : HyperText Mark-up Language
  - *esperanto* des fureteurs Web (browers)
  - Données non typées, (presque) non-structurées
  - Données hétérogènes (texte, images, etc.)
  - Pour la présentation visuelle
  - Inclut des liens vers « d'autres sources » (similitude avec objets composites)

6/7) Serveurs Objets

- Application = {Objets Communicants}
- Communication par un Object Request Broker (bus logiciel)



7/7) Serveur Web-Objet : un modèle d'architecture

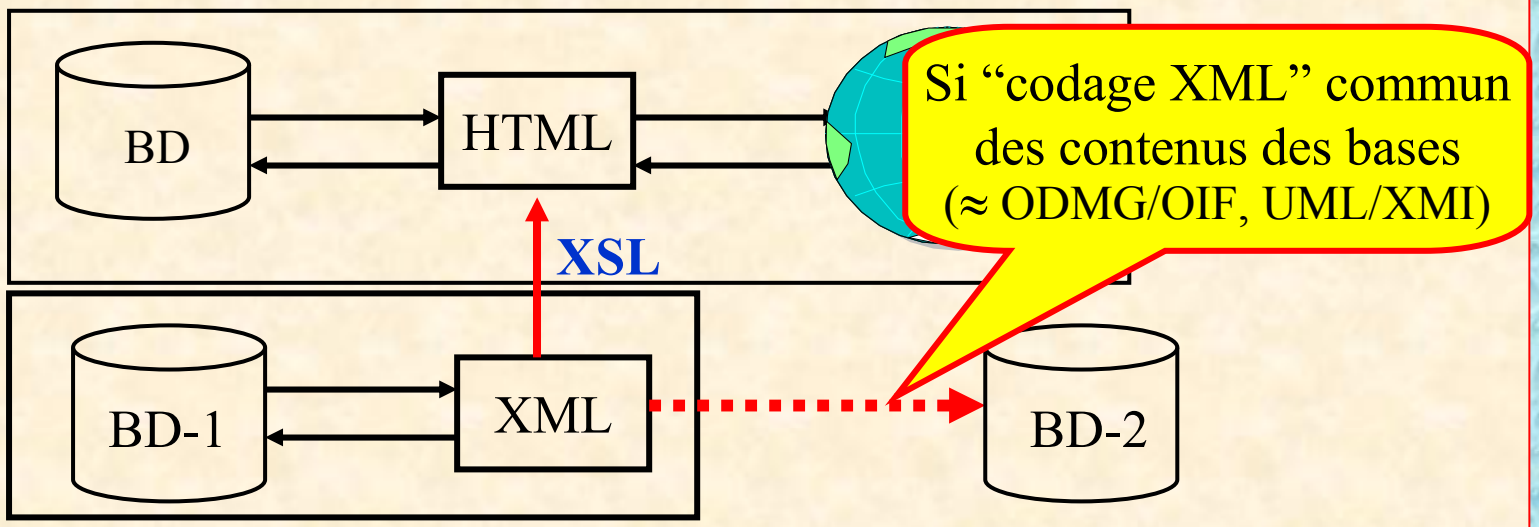


Partie III : SGBD et Architectures Client/Serveur

1. Notion de Client/Serveur
2. Typologie des serveurs
3. Web et Bases de Données
    1. Le Web en tant que Base de données
    2. Notion de données semi-structurées
    3. Introduction à XML et sa galaxie

III.4-1. Web en tant que BdD

- ≠ Accès à des bases via le Web (cf. types d'architectures Client/Serveur)
  - Langages de scripts (perl, php, etc.)
  - Convertisseurs de formats (médiateurs):
    - Peer-To-Peer vs Représentation Commune
    - Ce qui est ou n'est pas (commercialement) disponible :

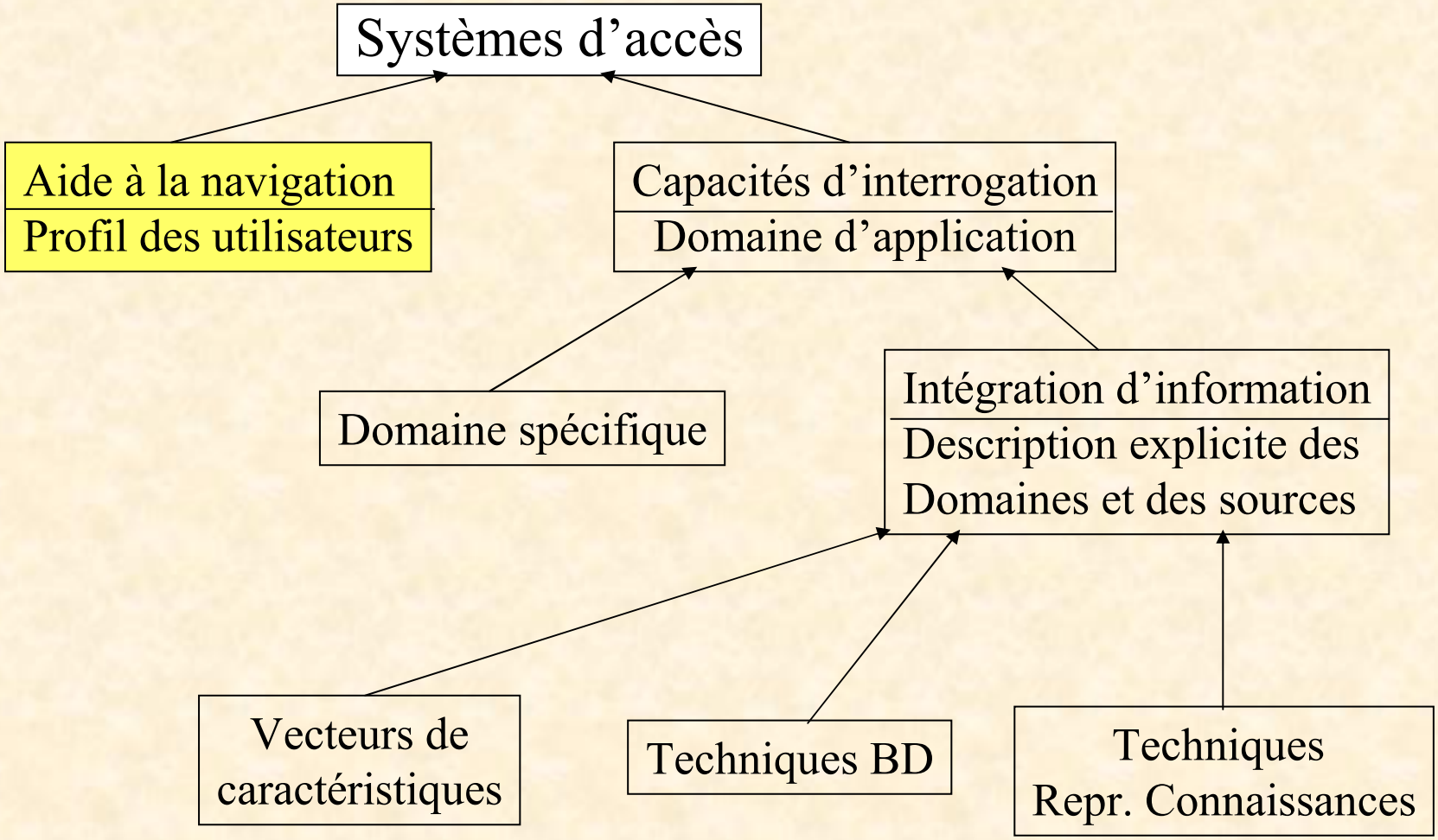


1. Web en tant que BdD

- Le Web comme base de données : pourquoi ?
  - Sources d'informations
  - Introduction d'un peu « d'organisation »
- Interrogation base de données : schéma nécessaire (!?)
- A priori, aucune structure (Web ≈ gigantesque graphe) : quel(s) modèle(s) de données ?

1. Web en tant que BdD

- Systèmes d'accès (T. Catarci, ER'99, Paris)



1. Web en tant que BdD

1. Systèmes d'aide à la navigation (« *surfers* »)
   - Détection profil utilisateur pour l'aider à naviguer

2. Systèmes avec capacités d'interrogation
   - Offrir une vue intégrée de différentes sources
   1. Dédiés à un domaine («*Information Brokers*»)
   2. Généraux : distinction fondée sur le mode de représentation et d'exploitation des informations
      a) Vecteur de caractéristiques
      b) Modèles de données (≈ Bases de données)
      c) Représentation des connaissances

1. Web en tant que BdD

a) Systèmes à vecteur de caractéristiques

- Document caractérisé par un vecteur d'attributs
- Cas des moteurs de recherche (*Altavista, Lycos, etc.)*
- Exploration et indexation des documents
- Recherche par mots-clés

(+) Facilité d'apprentissage

(-) (im)Précision des réponses

b) Systèmes avec modèle de données

- Web ≈ base contenant des informations sur le domaine d'application
  - Intégration des sources : cf. fédération de bases de données (*Tsimmis*)
  - Classification et stockage local des informations (*Araneus*)
- Base « virtuelle » (cf. notion de vue), physique ou les 2
- Médiateurs distribuent les requêtes aux sources
- « Wrappers » se chargent des formats des données

( ) Modélisation des sources
( ) Extraction non automatique de la structure des sources
( ) Evolutivité

c) Systèmes fondés sur la connaissance

- Techniques des SBC pour :
  - Caractérisation de sources
  - Réponses aux requêtes
- Web ≈ Réseau sémantique
- Base de connaissances = Structure de concepts liés par des relations sémantiques

(-) Temps de réponse

(-) Vocabulaire commun (→ Ontologies)

1. Web en tant que BdD : Quelques problèmes

- Accès « intelligent » aux informations intimement liés à leur représentation
- Représentations simples (ex: moteurs de recherche) : Efficaces mais inappropriées
- Représentation et structuration des connaissances difficiles mais peuvent contribuer à de meilleures résultats
- Dynamicité du Web (vie/mort des sites)

1. Web en tant que BdD : Quelques problèmes

- Optimisation des requêtes (cf. inclusion de requêtes et « caching »)
- Hétérogénéité et non structuration des sources
- Structures évolutives (cf. dynamicité)
- La suite :
  - Notion de données semi-structurées
  - XML:
    - Langage de description et d'échange
    - Langages d'interrogation pour XML

Web en tant que BdD : 2. Données semi-structurées

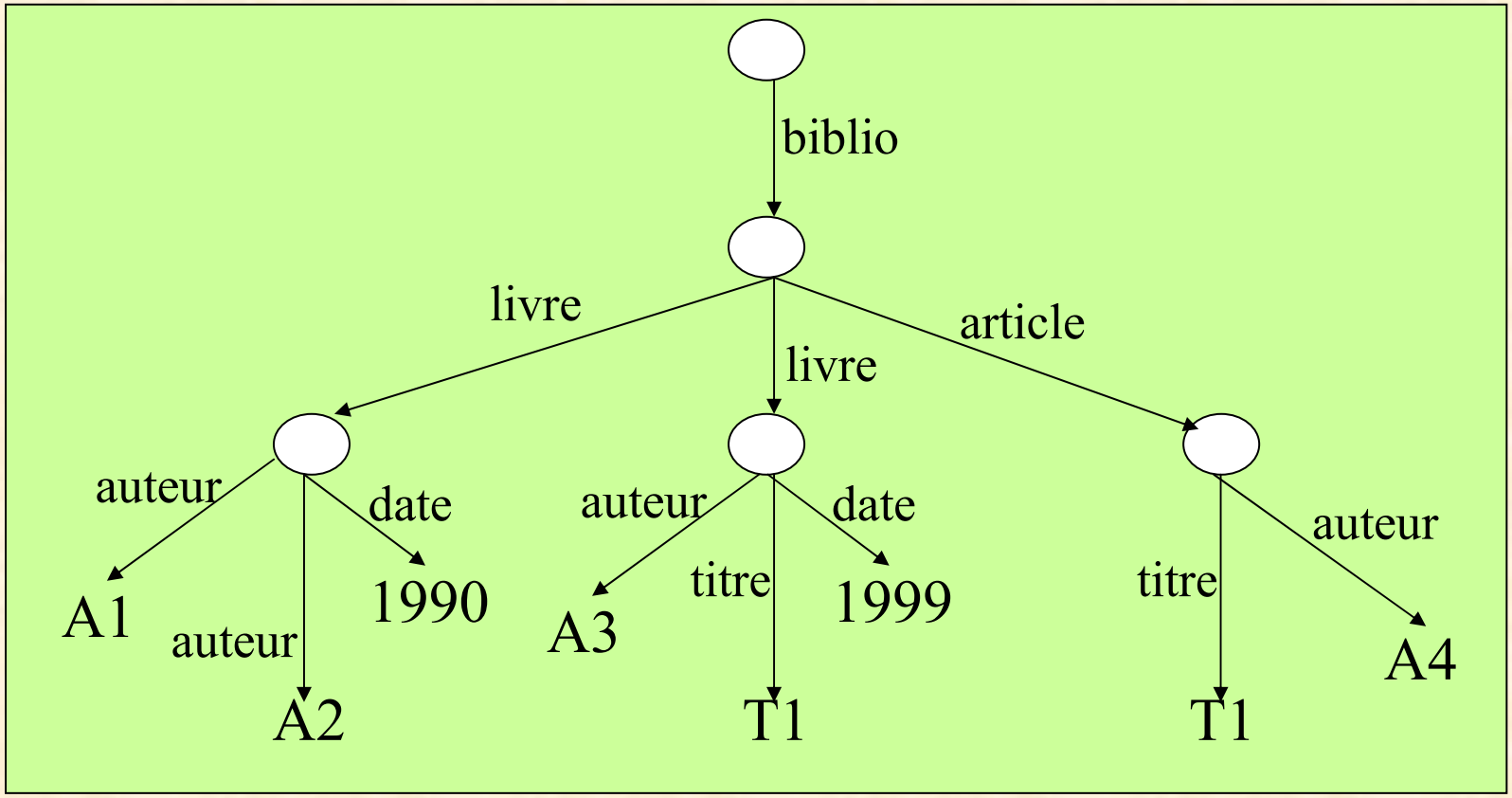
- Notion de données semi-structurées
  - ≈ « Tout ce qui n'est pas relationnel »
  - Pas de langage de description de schéma
  - Données auto-décrites

- Exemples :
  - Bibtex (« rubriques » author, title, year, etc.)
  - Données sur le Web (HTML, XML)
  - Formats d'échange de données (ASN.1, Step/Express, etc.)

2. Données semi-structurées : Caractéristiques

- Pas nécessairement la même structure
  - Données manquantes (ex: Annotation BibteX)
  - Type varié (ex: date, adresse)
- Structure peut être implicite
- Délibéremment
  - « Oublier » le type de l'instance
  - Sérialiser les valeur en annotant chaque élément par sa description

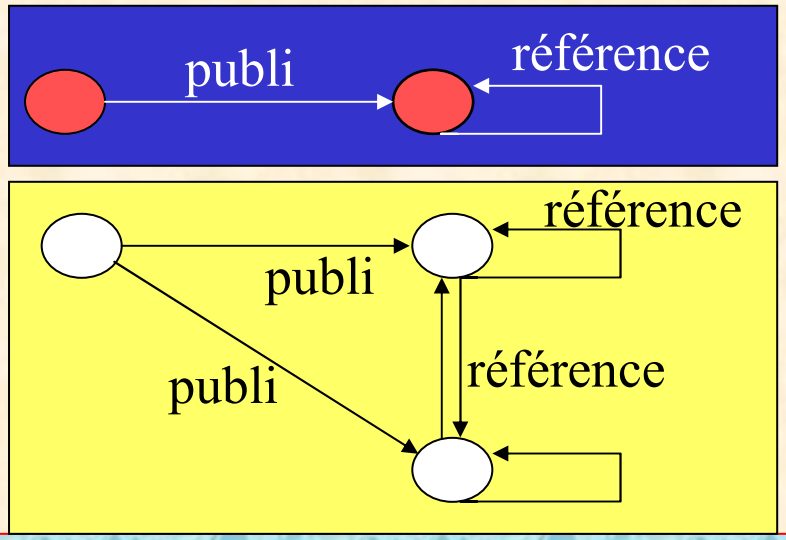
- Parcours/Transformation de graphe de données (Récursion structurelle) :

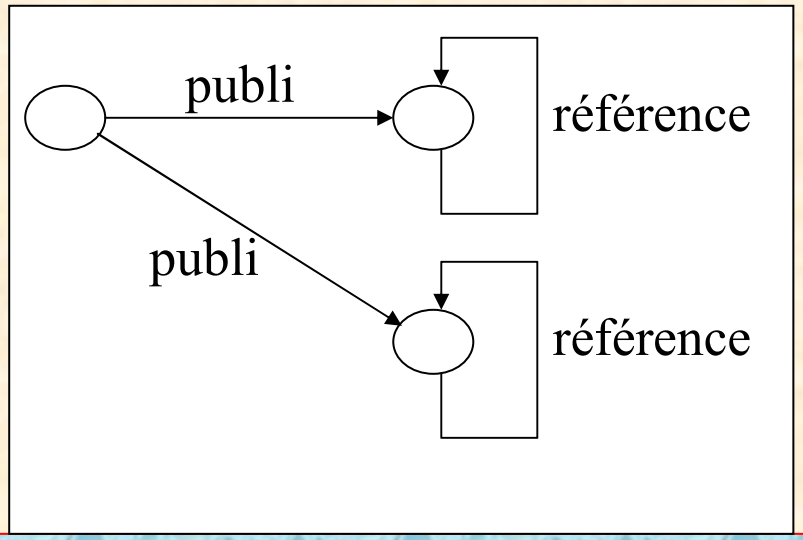


- Transformer tous les entiers en chaînes
  - F(v) = si Entier(v) alors Ent2Chaine(v) sinon v
  - F({}) = {} \\ {} : graphe à un nœud (feuille)
  - F({l : t}) = {l : F(t)}
  - F(t1 ∪ t2) = F(t1) ∪ F(t2) \\ t1, t2 : sous-arbres

2. Données semi-structurées : Egalité

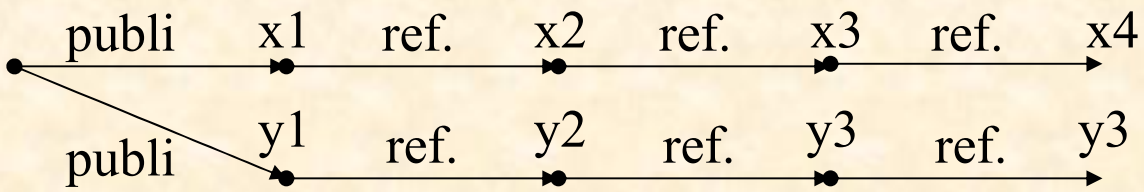
- t1: {livre: {auteur: “A”, auteur: “A”, titre: “T1”}}
- t2: {livre: {auteur: “A”, titre: “T1”}}
- t1 et t2 dénotent le même objet ?
  - Oui si élimination de la redondance
- Cas des données cycliques:





2. Données semi-structurées : Egalité

- « Déplier » et comparer les arbres



Si x1 = y1, …, xn = yn alors les objets sont égaux

- Comment comparer sans déplier :
  - Calcul d'une relation de bi-simulation

2. Données semi-structurées : Conclusion

- Structures
  - Irrégulières
  - Implicites (extraction ?)
  - Incomplètes
- Absence ou faible typage
- Exemples : OEM, LOREL
- XML : même veine ?

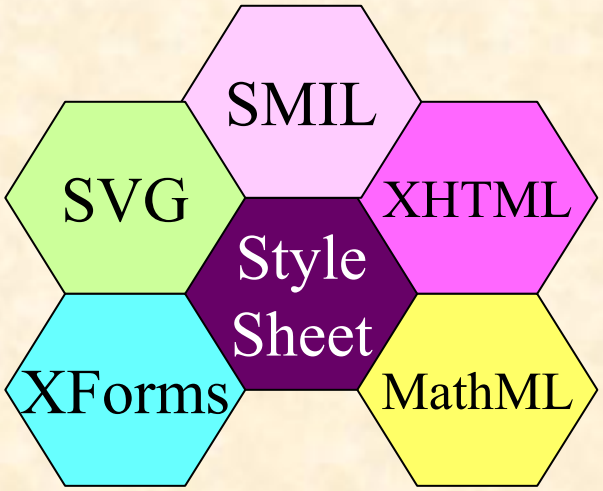
Web en tant que BdD : 3. Introduction à XML

- XML: eXtendible Mark-up Language
  - World Wide Web Consortium (www.w3c.org)
  - Sous-ensemble de SGML (Standard Generalized Mark-up Language) : Gestion Electronique de Documents
  - Description du contenu des documents
  - Objectif Principal : Echange de données
  - Marquage (balisage) « libre » et extensible
  - Hypothèse : Compréhension commune des balises (e.g. XMI pour l'échange de diagrammes UML)

Web en tant que BdD : 3. Introduction à XML

- Données *a priori* non interprétées (**P**arsed **C**haracter **DATA**)
- Langage facilement analysable
- Mise en forme du contenu (XML → HTML)
  - **CSS** (Cascading Style Sheet) : cf. HTML
  - **XSL** (eXtensible StyleSheet Language) : non reconnu comme standard
  - **XSL**Transformation ≈ XSL Microsoft (Internet Explorer)
- XHTML1.0: Reformulation de HTML4 en XML
  - Modules, Document profiles, DTDs (→ XML Schémas), …
  - Prochaine Génération de browsers Web : fondée sur XML ?

- XHTML1.0 : Recommandation, Jan. 2000
  - But : être utilisable avec des balises provenant d'autres « vocabulaires » XML, comme
    - SMIL : Synchronized Multimedia Integration Language
    - SVG : Scalable Vector Graphics
    - XForms : Formulaires Web
    - OFX : Open Financial eXchange

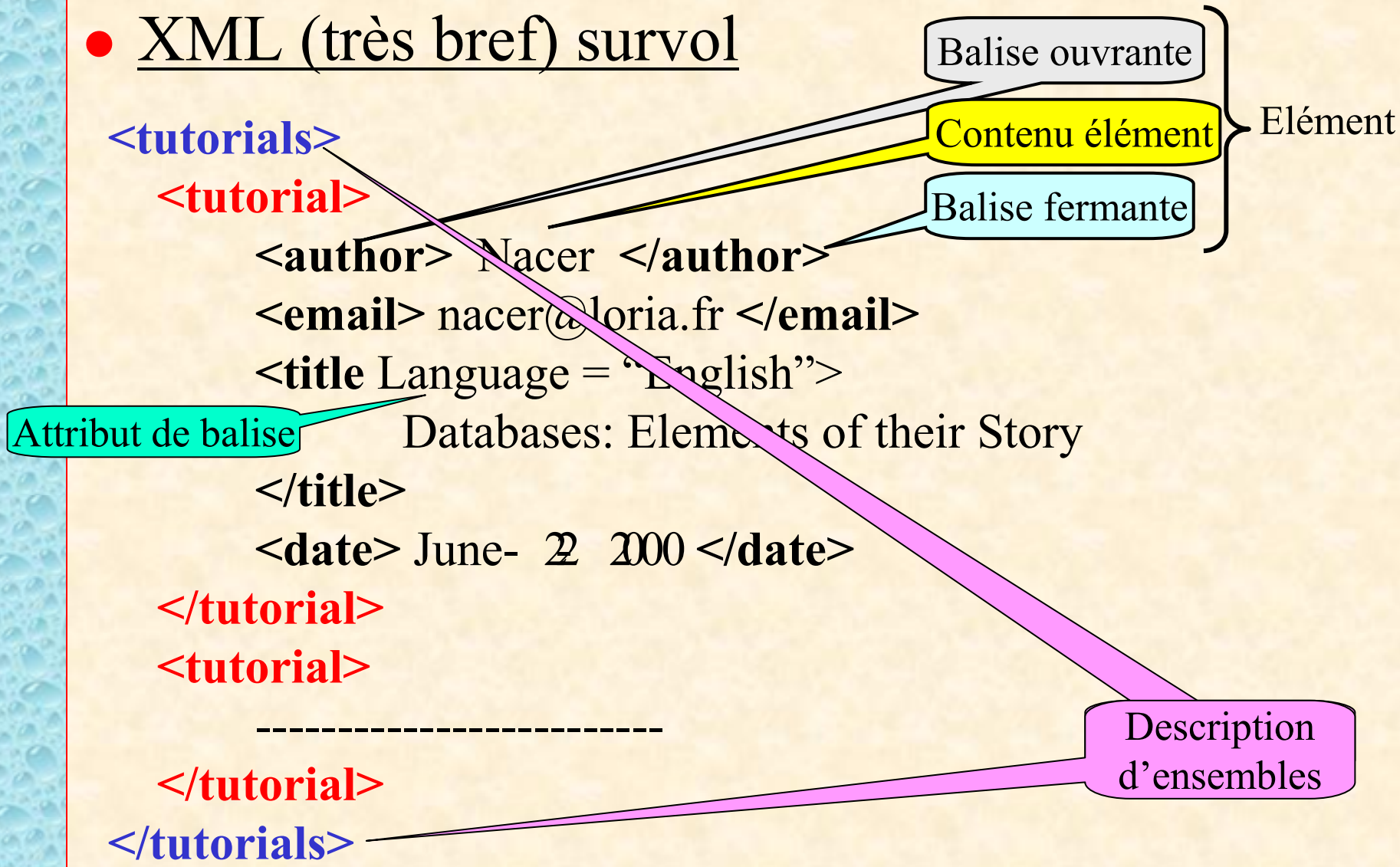


3.1- Introduction à XML

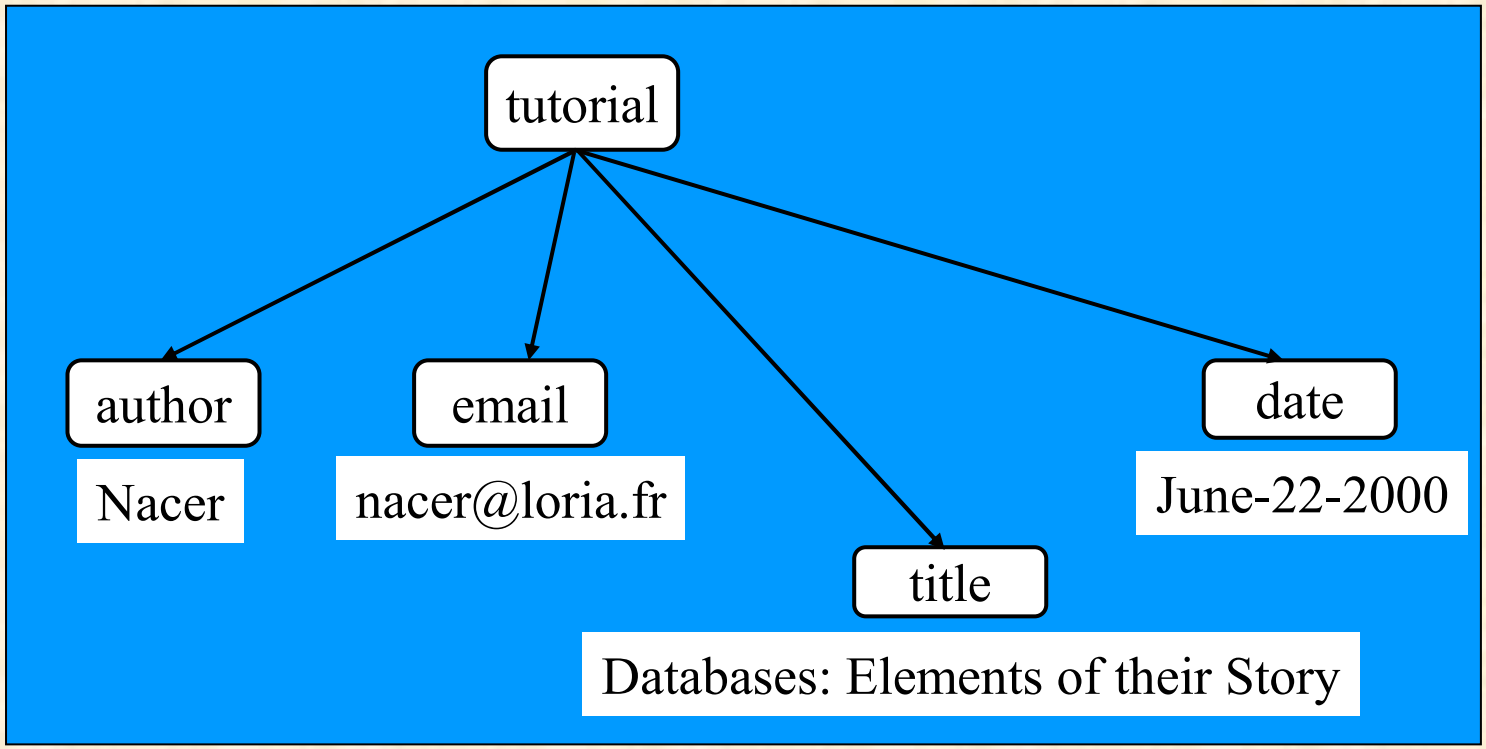
- XML :: HTML
  - Description du contenu
  - Pas d'ensemble prédéfini de balises
  - Support de structures imbriquées ≈ objets complexes (Xlink et Xpointer) :
    - Liens intra et inter documents
    - Navigation dans des documents XML
  - Données éventuellement auto-décrites :
    - **D**ocument **T**ype **D**efinitions
    - **D**ocument **C**ontent **D**escription (≈ Typage de document)
    - **R**esource **D**escription **F**ramework (pour Méta données)
    - XML Schema

3.1- Introduction à XML

- XML (très bref) survol



- Représentation :



3.1- Introduction à XML : Survol de XML (suite)

- Document Object Model (DOM) : API
  - Document vu et manipulé comme un arbre
  - Indépendance % langages, plateformes
  - Permet à une application de parcourir/d'agir sur un document
- Simple API for XML (SAX) : Déclenchement d'événement pendant l'analyse
  - DocumentHandler Interface
  - ErrorHandler Interface
  - DTDHandler Interface
  - EntityResolver Interface

3.1- Introduction à XML : Survol de XML (suite)

- Commentaires: <! Ceci est un commentaire >
- Processing Instructions (PI)

  <?xml-stylesheet href = "…." type = "…" ?>
  - xml-stylesheet: cible
  - Pas de sémantique associée aux PIs
- DTD: <!DOCTYPE nom [Définitions] >
- Structure d'un document XML

```
<?xml  ……. ?>                    <! Optionnel>
<!DOCTYPE  ….. >                  <! Optionnel>
{<baliseI>        …….  </baliseI>}*
```

- Exemple de structure de document (DTD)
  - Relations person (fn, ln, id), car(plate#, ownerId)

```
<?xml  version "1.0" ?>
<!DOCTYPE  rDB [
<!ELEMENT rDB (person*, car*)>
<! ELEMENT person (fn, ln, id) >
<! ELEMENT car (platenumber, ownerId) >
<! ELEMENT fn (#PCDATA) >
<! ELEMENT ln (#PCDATA) >
<! ELEMENT id (#PCDATA) >
<! ELEMENT platenumber (#PCDATA) >
<! ELEMENT ownerId (#PCDATA) >          ]>
```

- XML : Stockage et partage

  <!DOCTYPE rDB SYSTEM “http://www.../rDB.dtd”>

- Documents « composites » :

```
<?xml version “1.0” ?>
<!DOCTYPE tutorial [
        <!ENTITY %Scope SYSTEM “D:\TUT1\Tscope.xml” >
        <!ENTITY %Corps SYSTEM “D:\TUT2\Tcorps.xml” >
] >
<tutorial> <meta author = “ …… “, date = “ ……. “ />
        <title> Le Titre </title>
        %Scope;        <! -- Non inclus dans ce document >
        %Corps;        <! -- Non inclus dans ce document >
</tutorial>
```

- **XML Schema** : Définition d'une classe de documents

```
<?xml version ="1.0" encoding="ISO-8859-1"?>
<schema xmlns = "http//www.w3.org/2000/10/XMLSchema"
        targetNamespace =
        "http//www.w3.org/2000/10/XMLSchema">
<element name = "doc" type = "Document"/>
<complexType name = "Document">
        <attribute name = "Requete" type = "varchar"/>
        <element name = "MaTable1" type = "Table1"/>
        <element name = "MaTable2" type = "Table2"/>
</complexType>
</schema>
```

3.1- Introduction à XML : Survol de XML (fin)

- XML :: Données semi-structurées:
  - ≈ Graphe de données auto-décrites
  - ≠ Labels dans les nœuds (:: sur les arcs)
  - ≠ Ordre sur les éléments
  - ≠ Processing Instructions, Commentaires, etc.
- XML : pour l'échange (Constante évolution)
- Quel(s) langages d'interrogation pour le Web ?
  - Génération 1 : type SQL (ex: WebSQL, W3QL)
  - Génération 2 : Orientation objet (ex: WebOQL)
  - Génération 3 : Langages pour XML ?

3.2- Langages d'interrogation pour le Web (1/3)

- WebSQL: Web modélisé comme une base relationnelle avec 2 relations (virtuelles) i.e. graphe d'objets atomiques
  - Document(url, title, …)
  - Anchor (Base, label, href)
- Exemple: « Documents sur Nancy dans les serveurs en Algérie »

```
SELECT d.url, d.title
FROM Document d SUCH THAT d MENTIONS "Nancy"
WHERE d.url CONTAINS ".dz"
```

3.2- Langages d'interrogation pour le Web (2/3)

- WebOQL: Web modélisé comme un graphe d'objets structurés
- Interrogation :
  - Le Web
  - Une Page
  - Un ensemble de pages liées entre elles
- Restructuration :
  - HTML-HTML
  - HTML-Base de données
  - Base de données-HTML

3.2- Langages d'interrogation pour XML (3/3)

- XQuery : Influences
  - Quilt : langage de requêtes pour XML
  - SQL : Modèle « Select from where »
  - Xpath 1.0 (*) et XQL : Expressions de chemins
  - XML-QL : « binding » de variables
  - OQL : aspect fonctionnel du langage avec expressions imbriquées

(*) Notation pour naviguer dans un document XML

3.2- Langages d'interrogation pour XML (3/3)

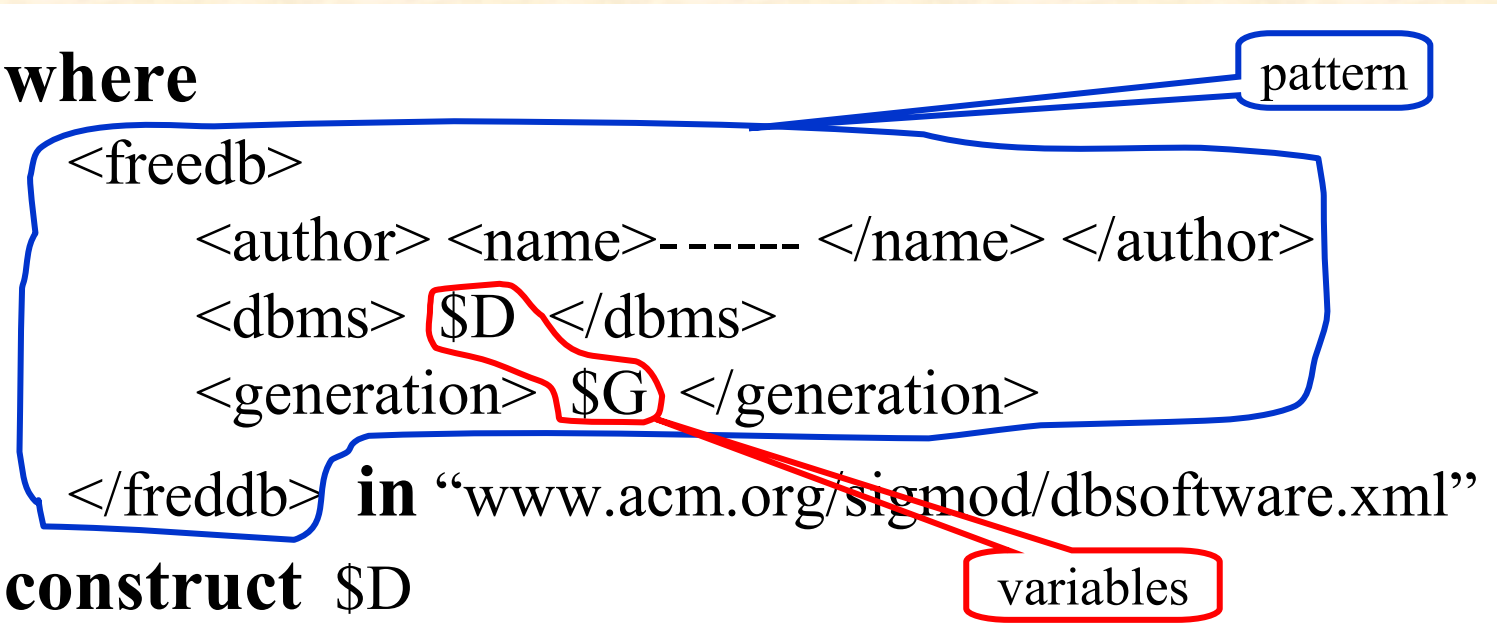
- XQuery : les exigences (requirements)
  - Non procédural
  - Syntaxe en XML et Syntaxe « human readable »
  - Cloture du langage
  - E/S de XQuery : instances de XQuery Data Model
  - Combinaison d'opérateurs y compris des requêtes comme opérandes
  - Possibilité de combiner des données de sources différentes
  - Agrégation et tri

3.2- Langages d'interrogation pour XML (3/3)

- XQuery : les exigences (requirements)
  - Accès aux méta-données (Schéma, DTD)
  - Extensibilité (types, fonctions)
  - Applicabilité à tout type de sources XML i.e.
    - Documents stockés en mode natif ou
    - Documents vus comme des documents XML (grâce à des médiateurs ou des convertisseurs de formats)
- La suite :
  1. XML-QL : une des influences de XQuery
  2. XQuery : des exemples

3.2- Langages d'interrogation pour XML: XML-QL

- Relationnellement complet, pour des données de type relationnel
- Principaux composants du langage:
  - **where** ≈ clauses from + where de SQL
  - **construct** ≈ select



3.2- Langages d'interrogation pour XML: XML-QL

- Intérêt des « patterns » :
  - Spécification du « modèle » de résultat
- Traitement des requêtes :
  - « Comparer » le pattern aux données
  - Lier les variables
  - Retourner les variables de *construct*

- Spécification de la structure du résultat :

```
<query>
  where
    <freedb>
        <author> $A          </author>
        <dbms>   $D </dbms>
        <generation> OO </generatio
    </freedb>  in "www.acm.org/…"
   construct
      <result> <author> $A </author>
               <dbms> $D </dbms>
      </result>
</query>
```

```
<result>
   <author>
     <name> --- </name>
   </author>
   <dbms> --- </dbms>
</result>
        ……...
<result>
        -------
</result>
```

3.2- Langages d'interrogation pour XML: XML-QL

- Prise en compte des différences de structure
  - Pas le même ensemble de balises
- Exemple: <release> non présent partout
  - SGBD et leur release, quand elle existe

```
where <freedb> $F </freedb> in "www.acm.org/…",
      <dbms> $D </> in $F
construct
      <result> <dbms> $D </dbms>
            where <release> $R </> in $F
            construct <release> $R </release>
      </result>
```

- Requêtes imbriquées: "SGBD par génération"

```
where
    <freedb>  <generation>  $G  </generation>
    </freedb>  in "www.acm.org/…"
construct
     <result>  <generation>  $G  </generation>
       where  <freedb>
                  <dbms>  $D  </dbms>
                 <generation>  $G  </generation>
               </freedb>  in  "----"
       construct    <dbms> $D </dbms>
     </result>
```

- Interrogation des attributs : "SGBD sous Unix"

```
where <freedb platform = "Unix" >
                <dbms> </>  element_as  $D
        </freedb>  in "www.acm.org/…"
construct  $D
```

- Associer des variables aux balises :

"Dupont auteur et chef de projet en 1999"

```
where  <$Tag> <dbms>  $D  </dbms>
                <released>  1999  </released>
                <$A>  Dupont  </>  </>  in "…",
construct  <$Tag>    <dbms>  $D  </dbms>
                <$A>  Dupont  </>      </>
```

- « Drafts » : juin 2001, 30 avril 2002
- Principaux types d'expression XQuery
  - Expressions de chemin
  - Expressions For Let Where Return (*FLoWeR*)
  - Expressions avec opérateurs et fonctions
  - Expressions conditionnelles
  - Expressions avec quantificateurs
  - Expressions testant ou modifiant les types de données

- Expressions de Chemins: Prédicats filtrant un ensemble de nœuds
  - document("fichier.xml") : nœud racine
  - . équivaut à self::node()
  - .. équivaut à parent::node()
  - nom équivaut à child::nom (/)
  - @nom équivaut à attribut::nom
  - // équivaut à /descendant-or-self::node()/

- ***document****("zoo.xml")//chapter[2]//figure[caption = "Tree Frogs"]*
  - Nœud racine : zoo.xml
  - 2ème descendant du nœud racine : <chapter[2]>
  - Recherche des éléments <figure> n'importe où dans l'élément <chapter>
  - Sélectionner ceux ayant un attribut <caption> avec la valeur "Tree Frogs.".

- ***document****("zoo.xml")//chapter[2* ***TO*** *5]//figure*

3.2- Langages d'interrogation pour XML: XQuery

- Chemins de localisation

  Syntaxe : **indicateur de relation::filtre[prédicat]**

- Troisième enfant : **child::*[3]**

- Dernier descendant:

  descendant::[position()=last()]

- Films sans acteur :

  film[count(child::acteur)=0]

- Film dont l'attribut titre est égal à Brazil :

  film[child::@titre='Brazil']

```
<bib>
   <book year="1994">
        <title>TCP/IP Illustrated</title>
        <author>  <last id=12>Stevens</last>
                  <first>W.</first>      </author>
        <publisher>Addison Wesley</publisher>
    </book>

    <book year="1998">
        <title>TCP/IP Illustrated< /title>
        <author>  <last id=12>Stevens</last>
                  <first>W.</first></author>
        <publisher> Morgan Kaufmann </publisher>
    < /book>
</bib>
```

- **[FOR | LET] … WHERE … RETURN**

FOR *$b* ***IN document***(*"bib.xml")//book*
WHERE *$b/publisher = "Morgan Kaufmann"*
AND *$b/year = "1998"*
RETURN *$b/title*

LET *$b :=* ***document***(*"bib.xml")//book*
RETURN *$b/title*

[unordered()|distinct()]

- Quantificateurs : exemple

```
<items>
    <item_tuple>
        <itemno>1001</itemno>
        <description>Red Bicycle</description>
    </item_tuple>
    <item_tuple>
        <itemno>1002</itemno>
        <description>Red Bicycle</description>
    </item_tuple>
    <item_tuple>
        <itemno>1003</itemno>
        <description>Red Bicycle</description>
    </item_tuple>
</items>
```

- Quantificateurs: **SOME | EVERY … IN … SATISFIES**

```
FOR $i IN document("data/R-items.xml")//item_tuple
WHERE NOT(
      SOME $b IN document("data/R-bids.xml")//bid_tuple
      SATISFIES $b/itemno = $i/itemno)
RETURN
      <no_bid_item>
            {$i/description}
      </no_bid_item>
```

- Jointure

FOR *$i* ***IN document***(*"catalog.xml")//item,*

$p ***IN document***(*"parts.xml")//part[partno = $i/partno],*

$s ***IN document***(*"suppliers.xml")//supplier[suppno = $i/suppno]*

RETURN

- etc.

3.2- XML (fin) : des adresses utiles

- Consortium W3 (tout sur XML et même plus):

  http://www.w3.org/

- Outils divers pour XML

  http://www.garshol.priv.no/dowload/xmltools

- Implantations de XQuery

  http://www.softwareag.com/developer/quip

  http://www.xhive.com/

Conclusion : Bases de données vs Web

- Bases de données
  - Distinction entre les niveaux logiques et physiques
  - Données structurées et fortement typées (schémas)
  - Langages de description et de manipulation
  - Concurrence, etc.
- Sources Web:
  - Données non ou semi-structurées (“typage faible”)
  - (Souvent) pas de schéma (méta-données)
  - Pas (encore) de langage d’interrogation

Conclusion : Bases de données vs Web (suite)

- Modèles de données
  - Stockage
    - Objets, ODMG
    - Fichiers texte
    - Base relationnelle/objet
    - Graphes/Arbres
- Indexation
  - cf. objets complexes
  - Maintenance des index?
- Schéma de données
  - Bases sans schéma ?
  - Extraire le schéma des données (cf. DB migration)?
  - Reverse engineering (HTML to XML schema) ?

Conclusion : Bases de données vs Web (suite)

- Langage de requêtes
  - ⇩ SQL like, OQL
  - ♋ Comparaison des Objets
  - ➔ Traitement des requêtes
  - ⇩ Portée des requêtes ?
  - ♋ Hétérogénéité
  - ⇩ XQuery (en cours)
  - ♋ Egalité profonde/de surface
  - ➔ Cf. requêtes distribuées
  - ➔ Avec/sans méta données ?
  - ➔ Cycles données/hyperliens ?
  - ⇩ Une fédération de sources ?
  - ⇩ World vide ?
  - ⇩ Site Web comme une vue ?
  - ⇩ Complexité et efficacité ?
  - ♋ Image, son, ...

Conclusion : Bases de données vs Web (suite)

- Optimisation de requêtes ?
- Mises à jour des sources ?
- Transactions ? Reprise ?
- Data mining vs Web mining
- Intégration Schémas/Vues (local vs global as view)
- Intégration de données hétérogènes

Conclusion : Bases de données vs Web

- Noyau de la technologie BdD comme base (conceptuelle) : Besoin d'adaptation et d'extensions
- Ne pas "ré-inventer la roue"
- D. Suciu (Keynote address, 7th Intern'l Conf. On Database Systems for Advanced Applications, "Web Data and the Resurrection of Database Theory", Hong Kong, 2001):

  "new artefacts are not concepts but standards"
- Nouveaux défis : hétérogénéité (données et structures), efficacité (stockage, requêtes), etc.

That's all Folks !

Merci de votre attention.

Questions ?

©Nacer.Boudjlida@loria.fr

Objets structurés et semi-structurés (fin)