

Approche « supply driven » : focus détaillé

Giuseppe Berio

giuseppe.berio@univ-ubs.fr

2024



Focus détaillé

- ▶ Transformation de schémas
- ▶ Échange de données
- ▶ Intégration de schéma
- ▶ Intégration de données

Approche « supply-driven » : perspective méthodes/techniques

3

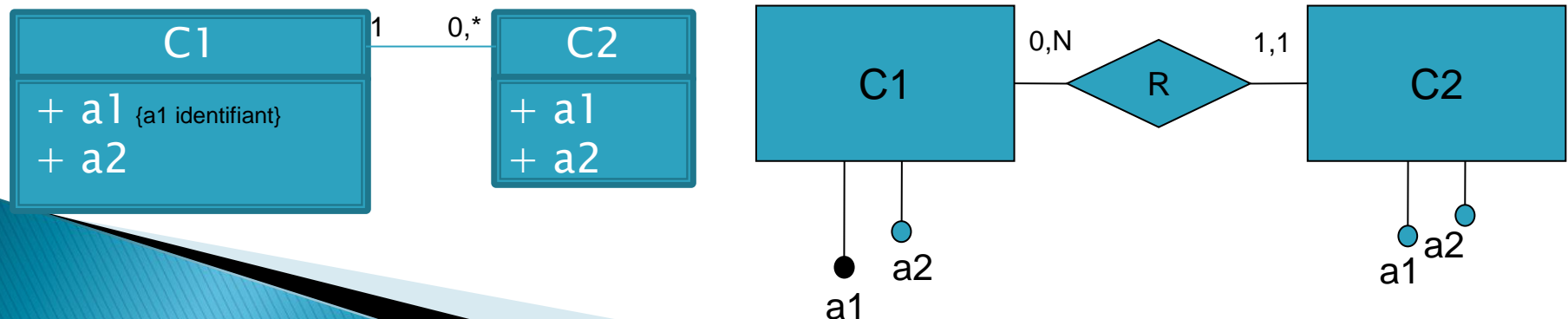
Identification sources	Conception SI	Conception SD	Conception Flux	Conception
Analyse (statique) de qualité : choix de sources	Transf Schéma ($C \rightarrow C$)	Transf Schéma ($C \rightarrow L; C \rightarrow C; C \rightarrow L$)	Échange de données : Mappings (conceptuel)	
Transf Schéma ($L \rightarrow C; C \rightarrow L$)	Intégration schéma (conceptuel) + Transf Schéma ($C \rightarrow L$)		Intégration de données (conceptuel)	Mise en œuvre
Réimplantation sources	Implantation SI	Implantation SD	Programmation flux	
Transf Schéma ($L \rightarrow P$ sources) + déploiement	Transf Schéma : $L \rightarrow P$ schéma intégré) + déploiement	Transf Schéma : $L \rightarrow P, L \rightarrow L$ schéma dimensionnel) + déploiement	Flux de chargement (SI-SD) + déploiement	
			Flux d'intégration (complétude + non redondance) (Sources-SI) + déploiement	
			Flux d'extraction et d'amélioration de la qualité (Source-Source) + déploiement	

Transformation des schémas (TS)

- ▶ Définition : produire un **schéma cible** à partir d'un **schéma source**
- ▶ Objectifs : **points de vue**, évolution et migration, implantation, rétro-ingénierie, génération de code
- ▶ **Notion d'équivalence ?**
 - N'est pas une nécessité formelle, même si elle peut avoir un intérêt (**traduction des schémas**, comme par exemple dans le cas de la normalisation relationnelle) car, si présentes, les données seront adaptées en fonction de la transformation (échange de données)
 - Souvent basée sur les données source qui doivent pouvoir être reconstruites à partir de données organisées selon le schéma cible (données source = F(données cibles))
- ▶ **Aspect critique** : la transformation doit être **justifiée** et **une liaison entre le schéma origine et le schéma résultat** doit être disponible (mais les données ne sont pas forcément en jeu dans cette liaison) ; le schéma résultat doit être testé (pour complétude et correction)

Transformation de schéma : Le langage commun

- ▶ Le langage commun est un **choix contextuel**
- ▶ Ce langage peut être un langage pour des **modèles logiques** de données (par exemple, le relationnel) et/ou pour des **modèles conceptuels** (par exemple, MERISE/EA/UML)
- ▶ Ce choix est typiquement un langage pour modèles conceptuels (car les autres sont généralement imposés) ; par exemple :



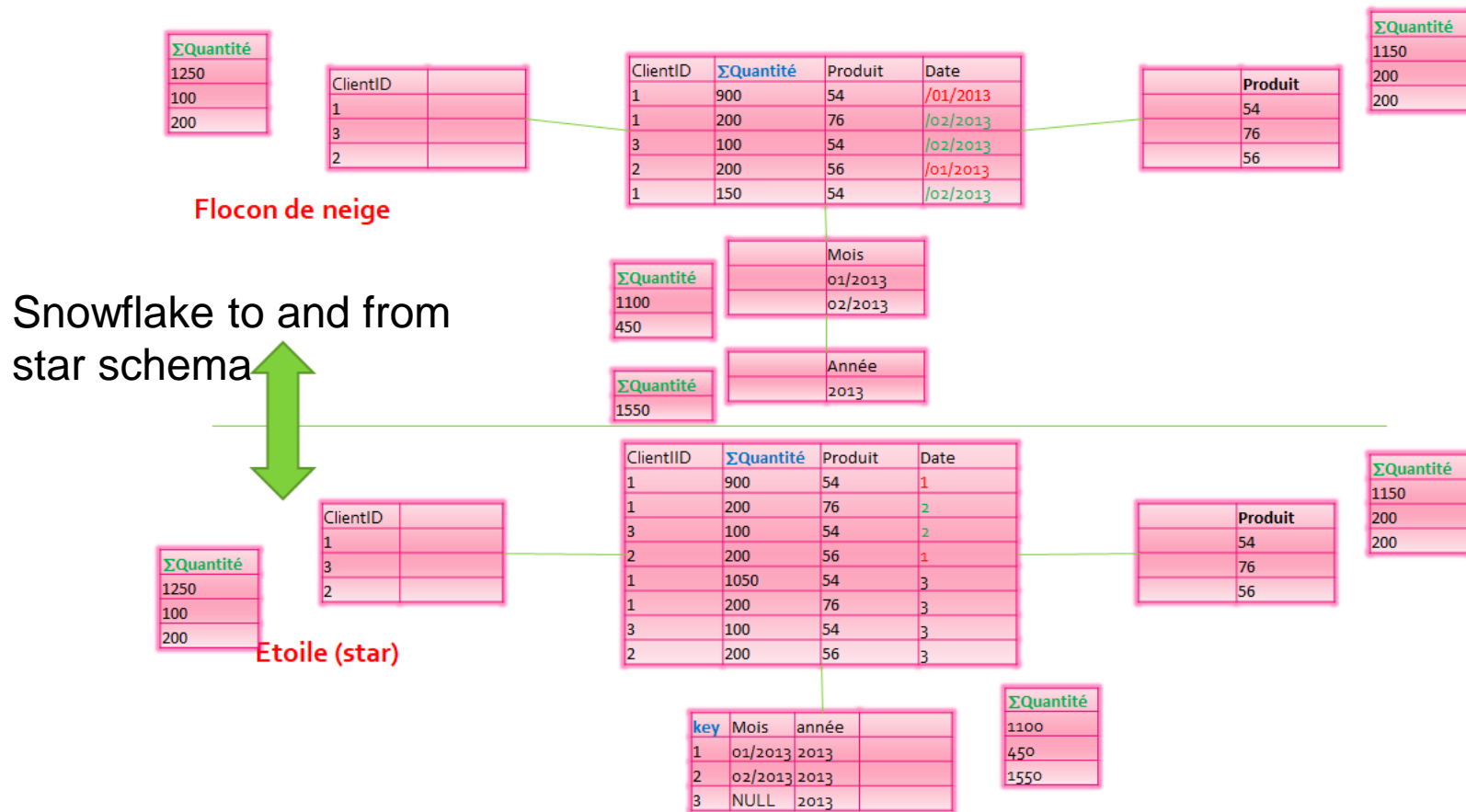
Types de transformation

- ▶ Types couramment utilisés :
 - Conceptual to logical (C to L) – exemple, mise en œuvre d'un schéma conceptuel via une bdd relationnelle transactionnelle
 - **Logical to logical (L to L)** – exemple, « normalisation relationnelle », « flocon → étoile », « colonnes/tables additionnelles »
 - **Logical to conceptual (L to C)** – exemple, « relationnel → diagramme ER »
 - **Conceptual to conceptual (C to C)** – exemple, « classes/attributs additionnels »
- ▶ En complément : Logical to Physical (L to P) et vice-versa (P to L)
- ▶ Ces types de transformation peuvent être composés, en particulier dans le cas des entrepôts
- ▶ Certains types sont « naturellement automatisables » : $P \rightarrow L, C \rightarrow L$; pour d'autres l'automatisation ne fournit qu'une base, parfois inutile

Transformation « conceptual to logical »

- ▶ Il s'agit de la transformation classique pour, par exemple, à partir d'un modèle plutôt conceptuel (comme un modèle UML de classes métiers ou un modèle ER et similaires) des tables relationnelles sont générées formant le schéma d'une base de données
- ▶ Ces transformations sont typiquement automatisables en grande partie

Transformation « logical to logical »



Transformation « logical to logical »

Professor (Id Name Sal)

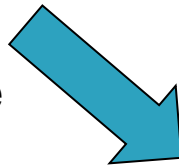
Clés primaires
surlignées

Student (Name Yr)

PayRate (Rank HrRate)

WorksOn (Name Proj Hrs ProjRank)

Abstraction (agrégation de
données, perte de détail)



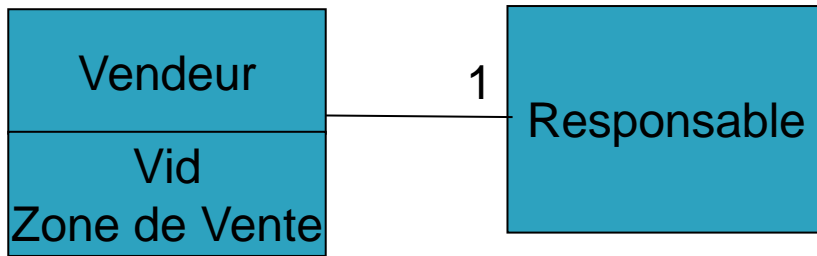
Personnel (Id Name Sal Addr)

Transformation « logical to conceptual »

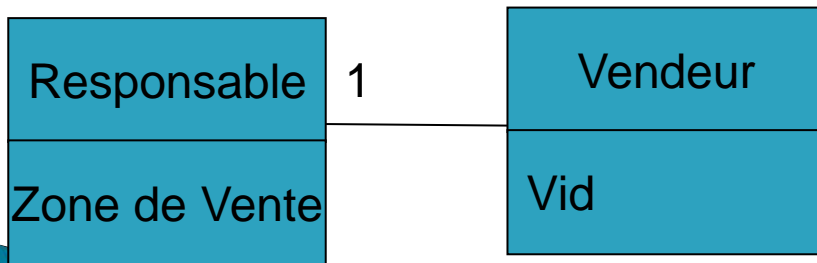
Vendeurs(Vid, Nom, Zone de Vente, Responsable)



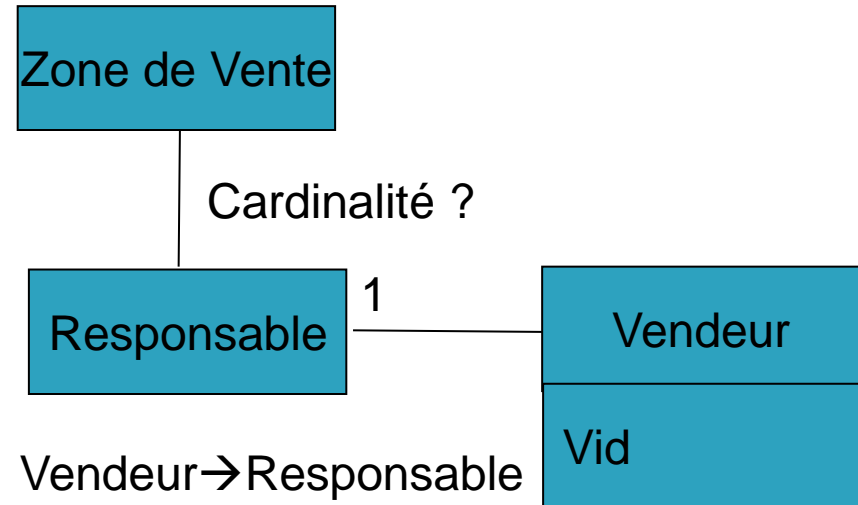
Transformations
possibles vers un
schéma UML_like
(enrichissement
semantique, distinction
relation/classe,
normalisation)



Vendeur→Responsable, ZdV



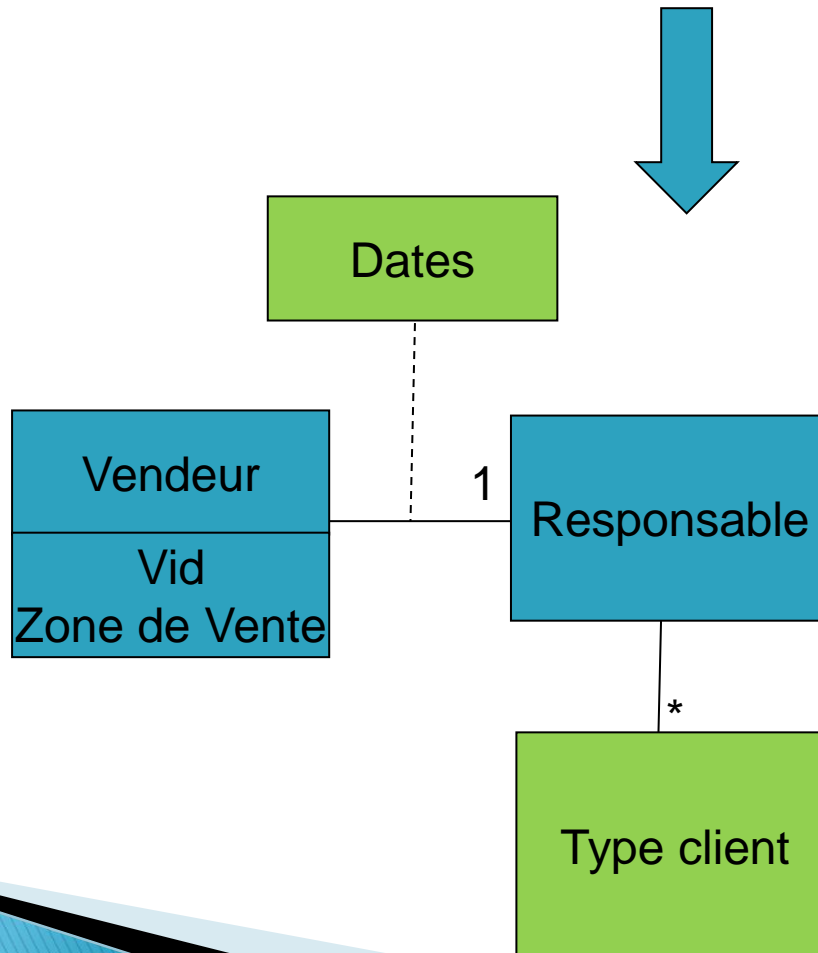
Vendeur→Responsable
Vendeur→ZdV



Vendeur→Responsable

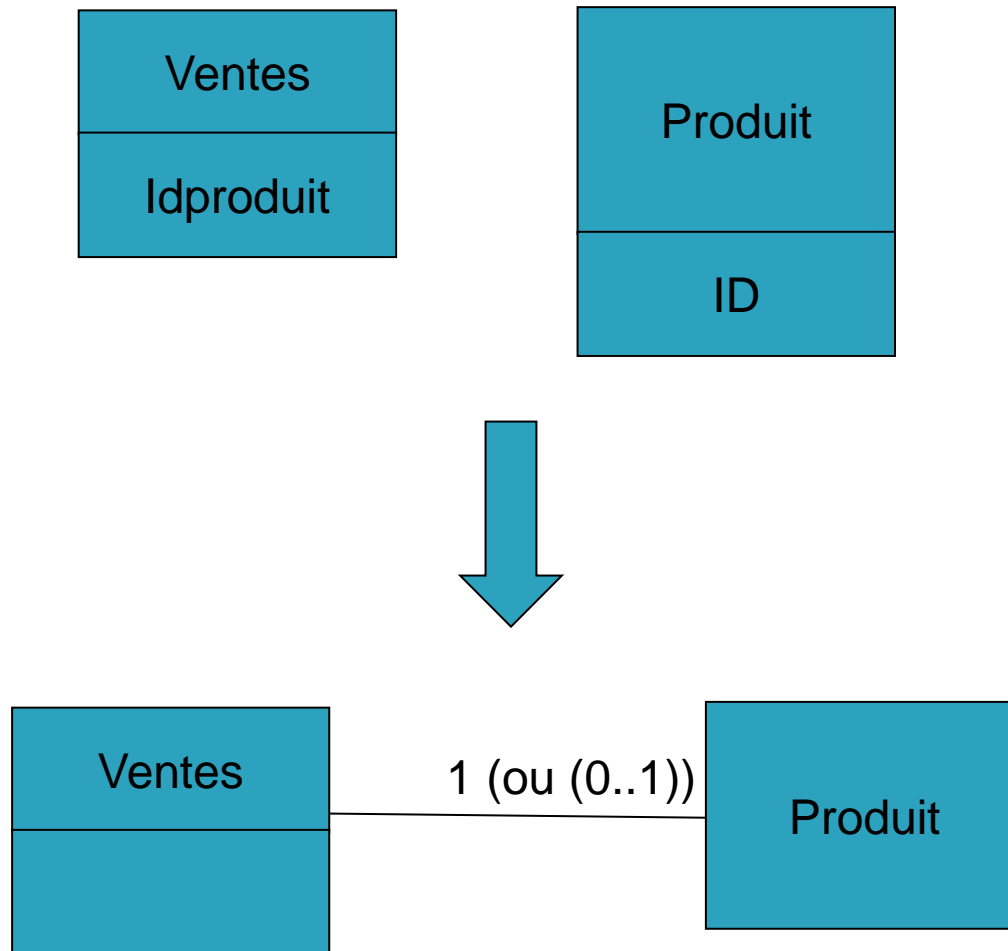
Transformation « logical to conceptual »

Vendeurs(Vid, Nom, Zone de Vente, Responsable)



Transformations possibles
vers un schéma UML-like
(enrichissement
semantique/nouvelles
données, même inconnues)

Transformation « logical to conceptual »

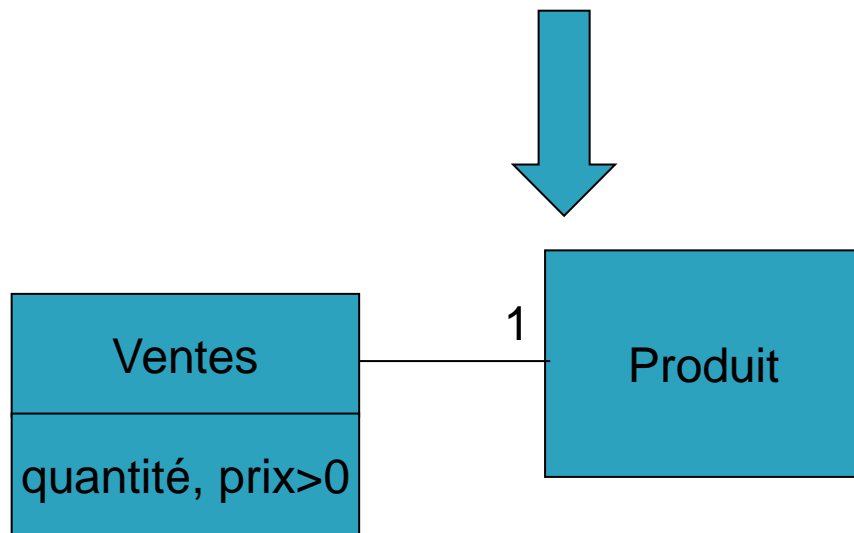


Enrichissement :
rajout des relations
(ou introduction de
clés étrangères)

Transformation « logical to conceptual »

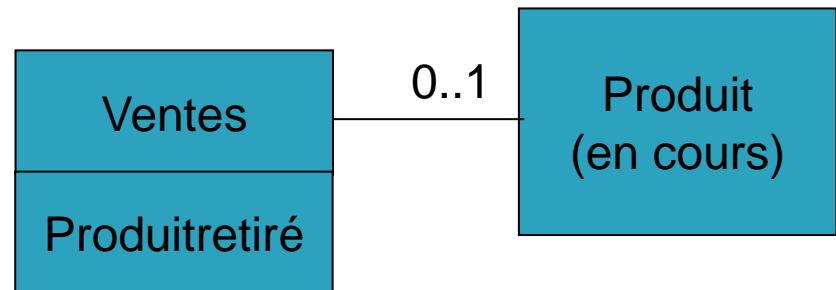
Produit(pid, prix, descriptif)

Ventes (produit not null, quantité, prix) sans clé étrangère sur Produit



Enrichissement :
nouvelles contraintes

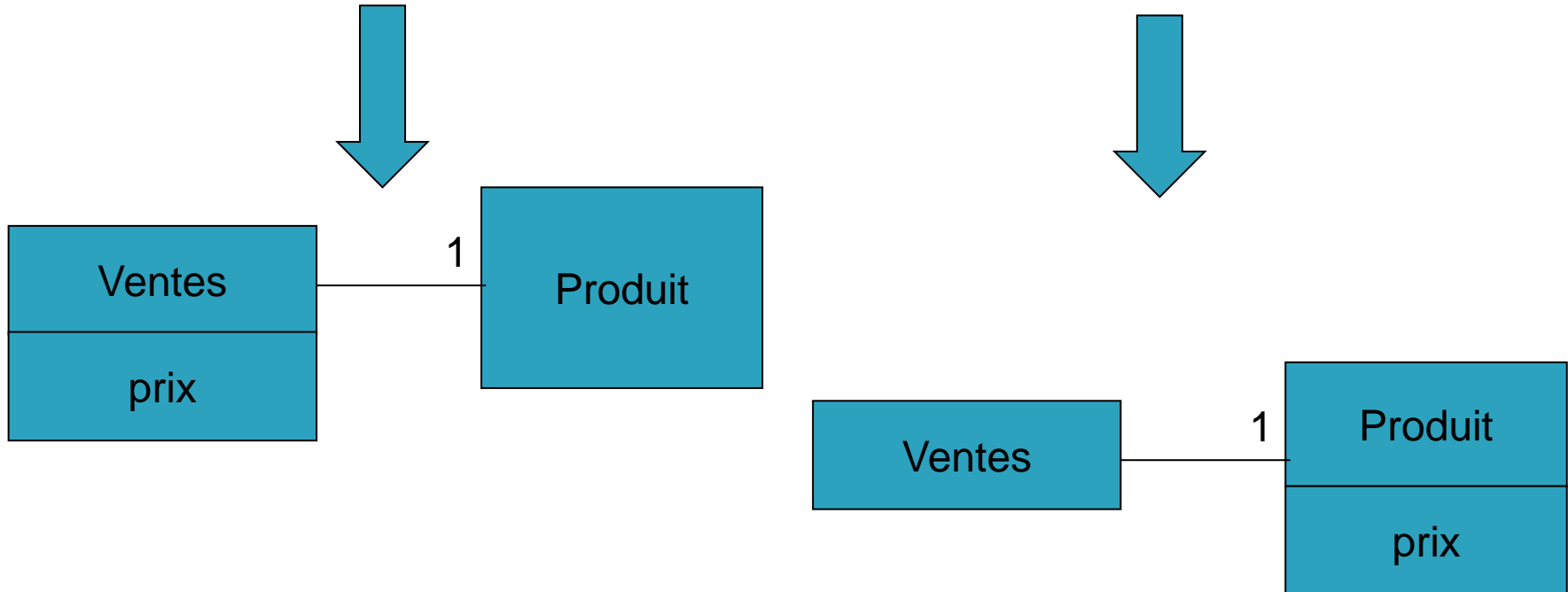
(Ventes.Produit is not Null and
Vente. Produitretiré is null) or
(Vente.Produit is null and
Produitretiré is not null)



Transformation « logical to conceptual »

Produit(pid, descriptif)

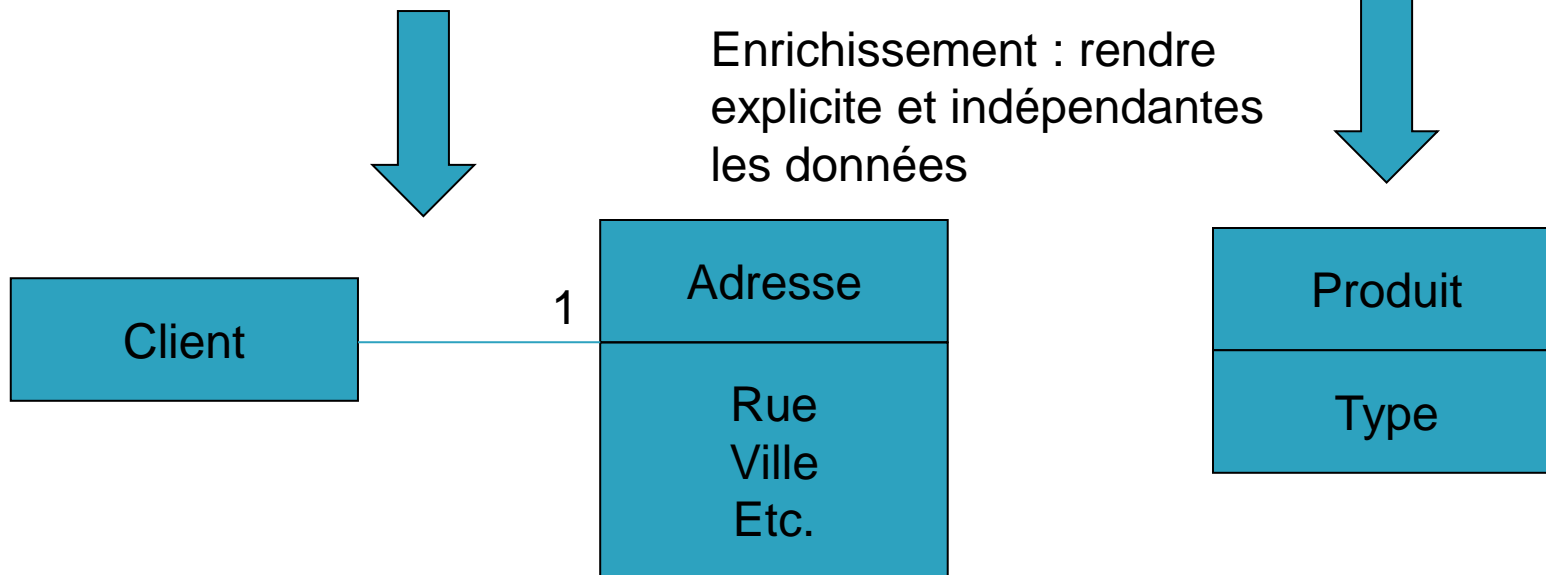
Ventes (produit not null, quantité, prix) avec clé étrangère sur Produit



Transformation « logical to conceptual »


Client(cid, nom, prenom, adresse)
/* si l'on connaît le format de
«adresse» et on sait le traiter

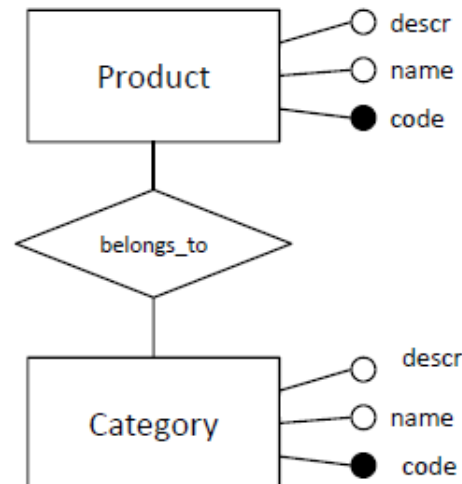
Produit(pid, descriptif, prix) /* si
l'on connaît le format de «pid» et
on sait le traiter, par exemple on
sait pouvoir extraire un «type» ou
un «type» est connu en fonction
de la source



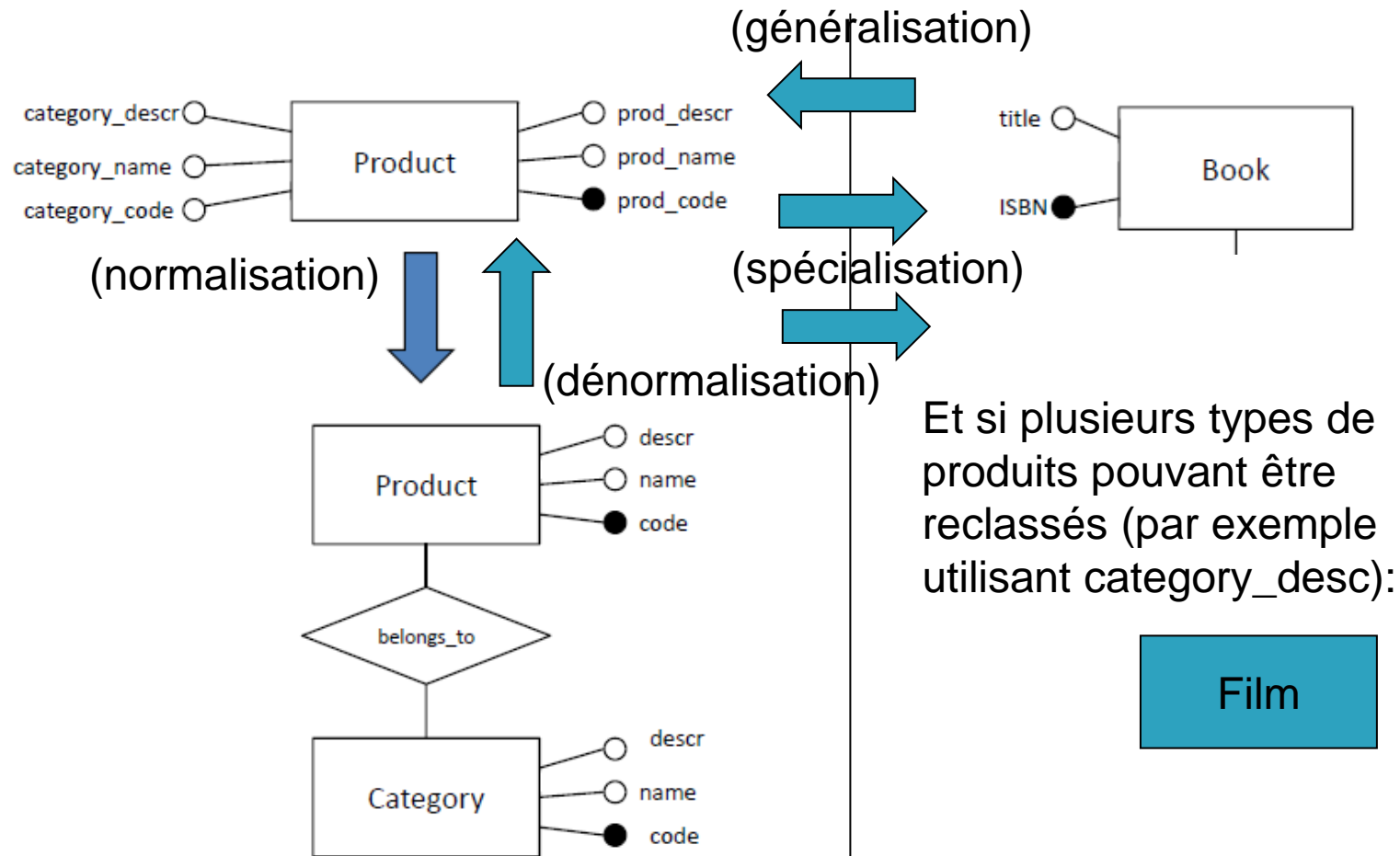
Transformation « conceptual to conceptual »



Transformation (C to C)  (normalisation : rendre indépendantes les données)



Transformation « conceptual to conceptual »

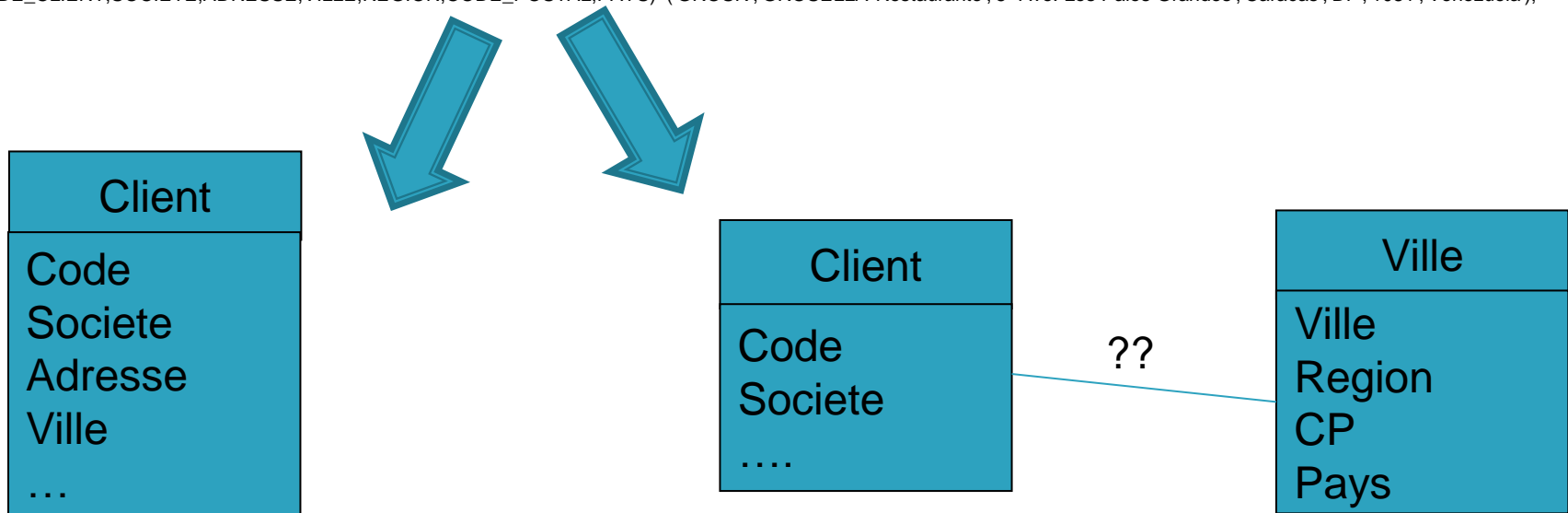


Transformation « logical to conceptual »

- ▶ Toute typologie de source (sgbd relationnels ou non relationnels – **noSQL** –, autres structures, fichiers textuels (**XML, CSV,...**), doit être modélisée d'une manière conceptuelle faisant ressortir
 - La signification de données plutôt que la forme
 - La notion d'identifiant au sens conceptuel (clé naturelle), plutôt que d'identifiant au sens de l'implémentation informatique
- ▶ En général, cette tâche n'est pas trop complexe, d'autant plus que ces formats ont été extensivement explorés (voir documents sur la plateforme : ERXML, ERnoSQL, ERnoSQL2)

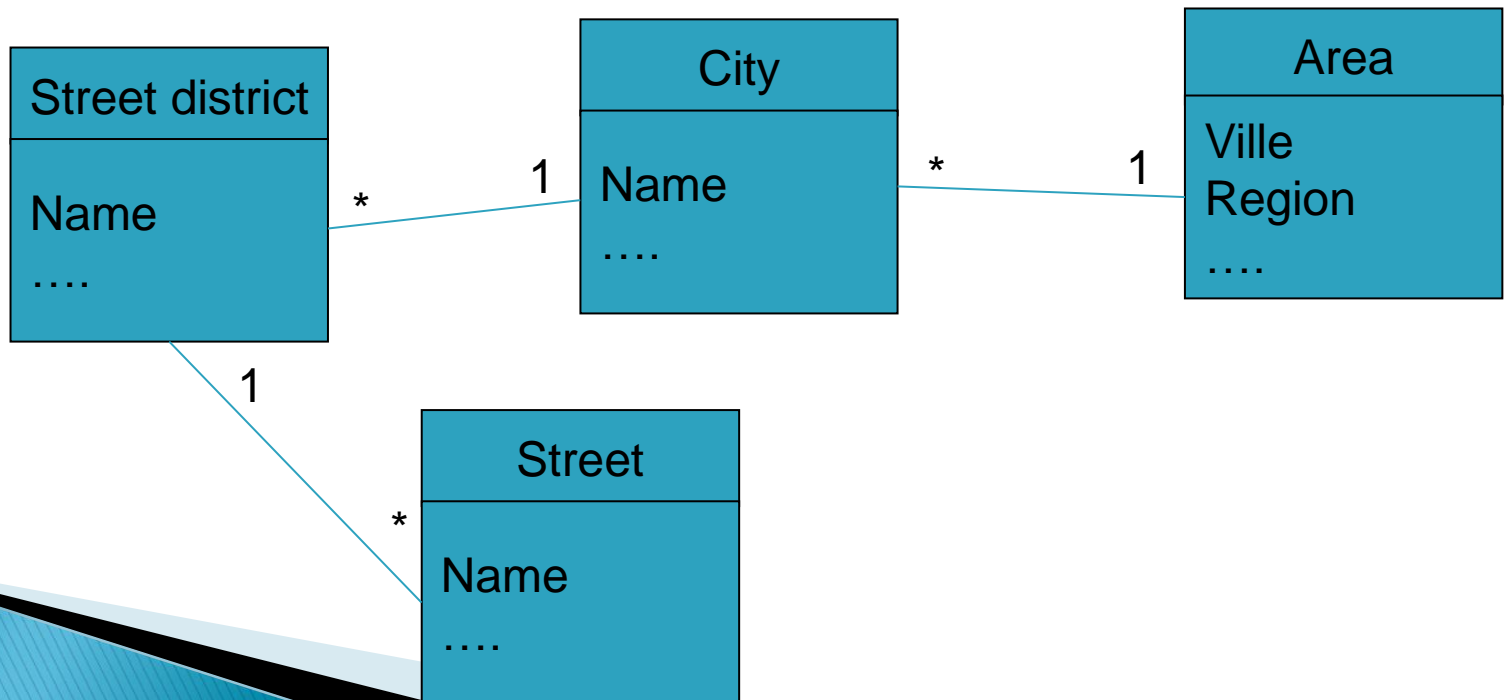
Exemple (txt)

```
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('VINET','Vins et alcools Chevalier','59 rue de l'Abbaye','Reims',null,'51100','France');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('HANAR','Hanari Carnes','Rua do Paço, 67','Rio de Janeiro','RJ','05454-876','Brésil');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('VICTE','Victuailles en stock','2, rue du Commerce','Lyon',null,'69004','France');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('SUPRD','Suprêmes délices','Boulevard Tirou, 255','Charleroi',null,'B-6000','Belgique');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('CHOPS','Chop-suey Chinese','Hauptstr. 29','Bern',null,'3012','Suisse');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('HILAA','HILARIÓN-Abastos','Carrera 22 con Ave. Carlos Soublette #8-35','San
Cristóbal','Táchira','5022','Venezuela');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('ERNSH','Ernst Handel','Kirchgasse 6','Graz',null,'8010','Autriche');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('OTTIK','Ottilies Käseladen','Mehrheimerstr. 369','Köln',null,'50739','Allemagne');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('QUEDE','Que Delícia','Rua da Panificadora, 12','Rio de Janeiro','RJ','02389-673','Brésil');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('RATTC','Rattlesnake Canyon Grocery','2817 Milton Dr.','Albuquerque','NM','87110','États-Unis');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('FOLKO','Folk och få HB','Åkergatan 24','Bräcke',null,'S-844 67','Suède');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('FRANK','Frankenversand','Berliner Platz 43','München',null,'80805','Allemagne');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('GROSR','GROSELLA-Restaurante','5ª Ave. Los Palos Grandes','Caracas','DF','1081','Venezuela');
```



Exemple (xml)

```
<areas>
  <area city="Paris">
    <street district="2eme arrondissement">Rue de la Paix</street>
  </area>
  <area city="Paris">
    <street district="8eme arrondissement">Champs Elysees</street>
  </area>
  <area city="New York City">
    <street district="Manhattan">Madison avenue</street>
  </area>
  <area city="New York City">
    <street district="Brooklyn">Washington heights</street>
  </area>
</areas>
```



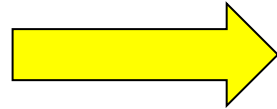
Transformation « logical to logical » par composition

Professor (Id Name Sal)

Student (Name Yr)

PayRate (Rank HrRate)

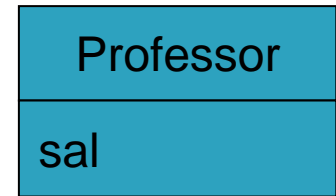
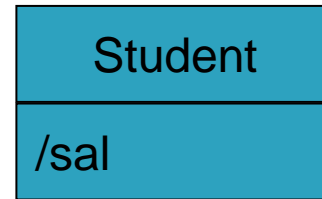
WorksOn (Name Proj Hrs ProjRank)



Personnel (Id Name Sal Addr)



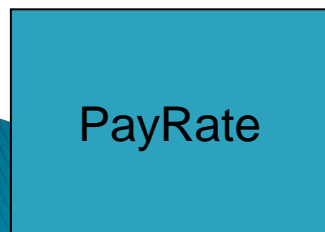
Transformation (C to L)



Transformation L to C
vers un schéma
UML_like (affiché sans
attributs)

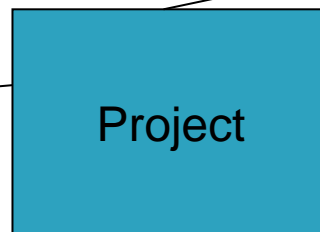


Transformation (C to C)



1

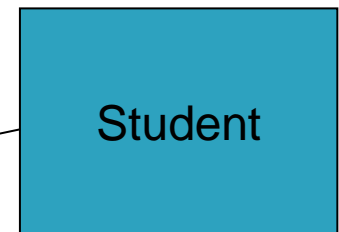
*



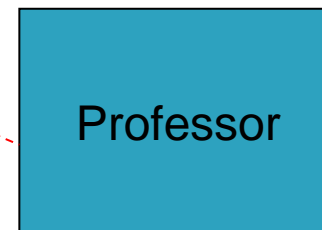
1



*



??



Transformations de schémas dans l'approche « supply driven »

Identification sources	Conception SI	Conception SD	Conception Flux	Conception
Analyse (statique) de la qualité de données et choix de sources	Transf Schéma (C→C) : résoudre les conflits entre schémas distincts portants sur de données se référant à une même réalité	Transf Schéma (C→C) : obtenir un schéma multidimensionnel se basant sur les données et schéma intégrés	Conception de flux d'extraction : Échange de données visant la limitation de données erronées, la standardisation (mappings conceptuels)	
Transf Schéma (L→C) : expliciter les données, améliorer la qualité (au sens large) de données à transférer vers l'entrepôt	Intégration schéma (conceptuel) : produire de schéma à partir de plusieurs schémas, limitant (excluant) la redondance de données et augmentant leurs complétude		Conception de flux d'intégration : Intégration de données visant non redondance, complétude et fraîcheur (mappings conceptuels)	
Réimplantation sources	Implantation SI	Implantation SD	Programmation flux	Mise en œuvre
Transf Schéma (C→L;L→P sources) + déploiement	Transf Schéma : C→L;L→ P schéma intégré) + déploiement	Transf Schéma : L→ L, L→ P schéma dimensionnel)+ déploiement	Flux de chargement : Échange de données (chargements successifs)+ déploiement (parallélisation)	
			Flux d'intégration (intégrations successives) + déploiement (parallélisation)	
			Flux d'extraction : Échange de données (mappings exécutables et extractions successives)+ déploiement (parallélisation)	

Points clé :

- Les transformations en rouge sont complexes ; plus ces transformations sont soignées, plus l'intégration de schémas sera simple ;
- Les transformations en vert sont généralement plus simples car bien cadrées, quoique parfois il faut faire des choix et la transformation vers un schéma physique peut être assez articulée, en particulier dans les cas des entrepôts

Identification des sources :

Transformations de schémas ($L \rightarrow C$)

- ▶ Une transformation de schémas correspond à des hypothèses de qualité sur les données disponibles
- ▶ Exemple (introduction d'une contrainte)
 - Prix > 0 (Hypothèse)
 - Lancement d'une requête (ou similaire) pour compter les données ayant un Prix ≤ 0
 - S'il y en a pas (ou peu), la transformation peut être validée
 - S'il y en a, le choix est entre « données erronées » et « hypothèse non validée »
 - Si « données erronées » alors, validation de la transformation
 - Si « hypothèse non validée » alors la situation doit être approfondie

Identification des sources :

Transformations de schémas

► Exemple (normalisation)

- Dans une table, l'on suppose l'existence d'une dépendance fonctionnelle $A \rightarrow B$, A n'étant pas clé de la table (ou UID d'entité ou contrainte classe PK-like)
- Lancement d'une requête (ou similaire) pour compter (ou visualiser) les données ne respectant pas la dépendance fonctionnelle
 - S'il y en a pas (peu), la transformation peut être validée (rajout contrainte « unique » ou décomposition de table/entité/classe)
 - S'il y en a suffisamment, le choix est entre « données erronées » et « hypothèse non validée »
 - Si « données erronées » alors validation de la transformation
 - Si « hypothèse non validée » alors la situation doit être approfondie

Identification des sources : Transformations de schémas

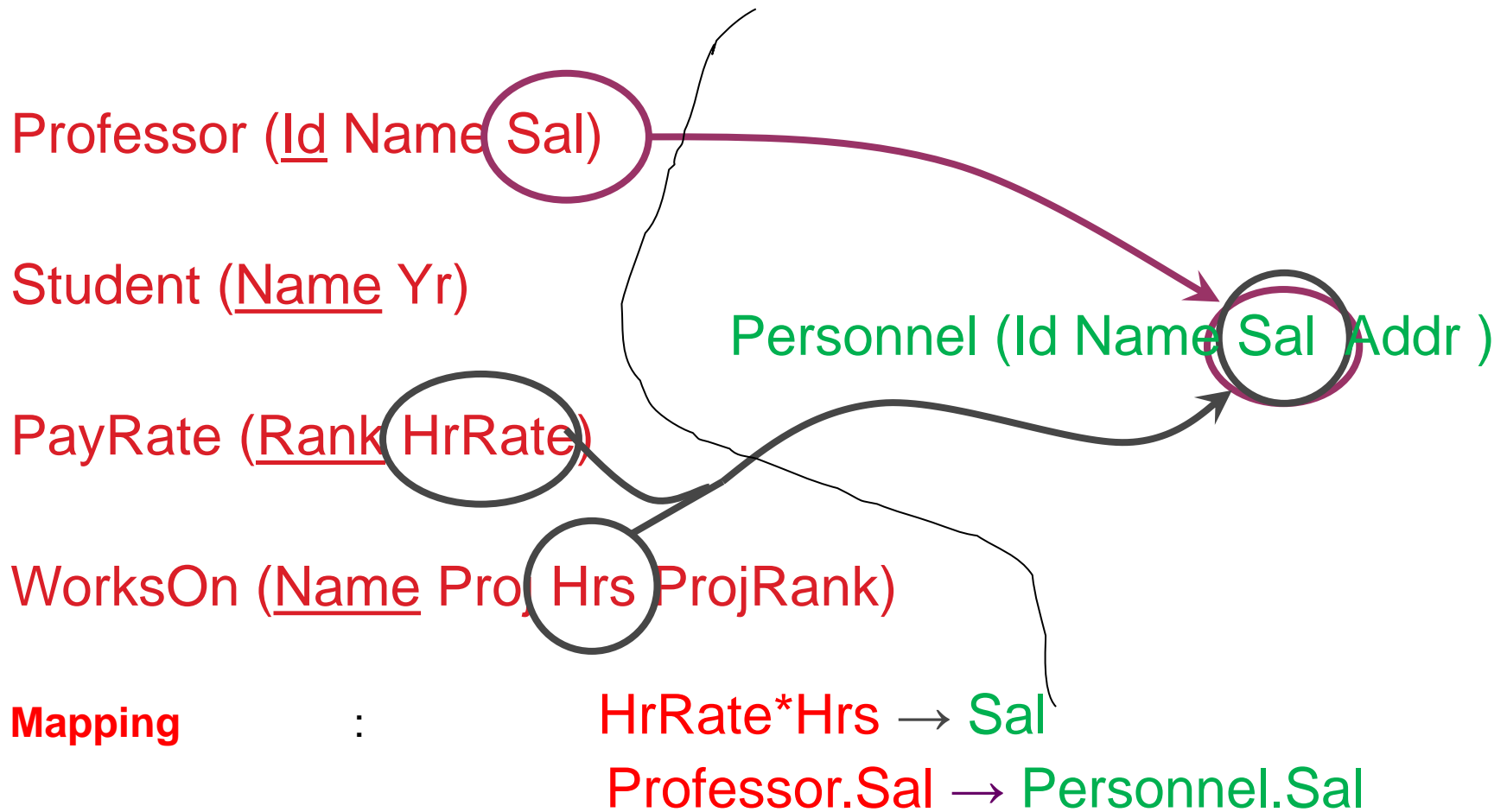
► Exemple (transaction time)

- Les données dans les sources peuvent ne pas être positionnées dans le temps (par exemple, on pourrait savoir qu'il y a eu une vente mais sans connaître la date de cette vente)
- Il est peut être possible de connaître le temps auquel la donnée était insérée dans la source (Tempsinsert)
 - Si la source est une base de données, cela s'appelle le « transaction time » (et peut être connu à travers plusieurs techniques)
- Si ce temps est connu, le temps dans lequel la donnée se positionne pourrait être estimé par
 - $\text{Tempsinsert} - \text{délaimoyeninsert}$
 - Où le délaimoyeninsert est l'intervalle de temps nécessaire à transférer vers la source une donnée connue
- D'une manière générale il est intéressant de disposer des ces 2 temps

Echange de données (ED)

- ▶ Dans ce cas, il est supposé de disposer d'un **schéma source** et d'un **schéma cible**
- ▶ Objectif : produire des **données cible** respectant un **schéma cible** à partir de **données source** (exemples : flux d'extraction, flux de chargement, flux SI→SD)
- ▶ Conceptuellement similaire à la **définition de vue dans les bases de données**
- ▶ Nécessite de préciser comment à partir de données sources les données cible sont générées ; cela passe par la spécification de **mappings (exécutables)** ; il est donc usuel d'utiliser des **schémas logiques** pour spécifier l'échange de données

Exemple I



Comment générer la base de données cible à partir d'une base source ? Générer une (ou plusieurs) requête(s) correspondante(s)

Exemple II

On considère le schema source ci-dessous :

Cours (cn, intitulé, credits)

Promo (cn, sem)

Enseignant (id, nom)

Enseigner (id, cn, sem)

Titularité (cn, id)

Clés primaires
surlignées

et le schéma cible ci-dessous :

NSchema (cours, intitulé, nom_ens)

Exemple II

Schéma source :
Cours (cn, intitulé, credits)
Promo (cn, sem)
Enseignant (id, nom)
Enseigner (id, cn, sem)
Titularité (cn, id)

Cours

cn
intitulé
credits

Enseignant

id
nom

Mapping conceptuel (exprimée
comme relations entre colonnes) :

Schéma cible

cours
intitulé
nom-ens

Tables sources

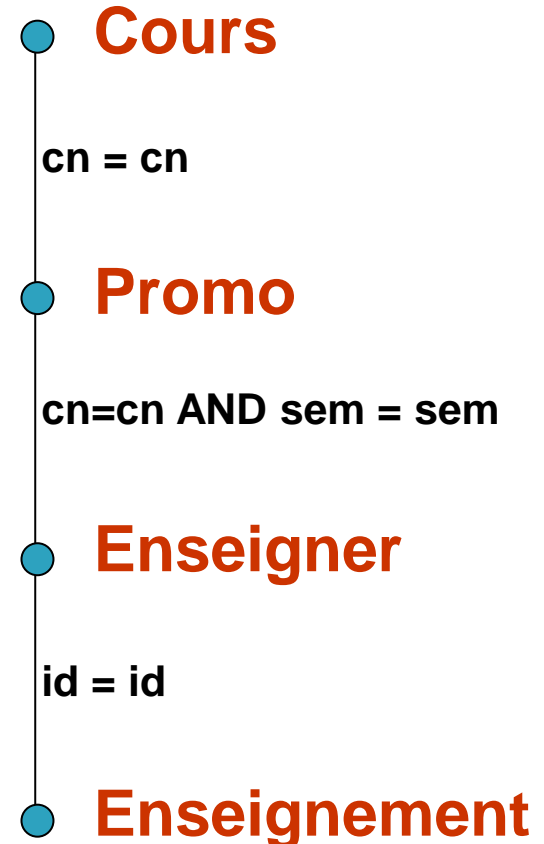
Tables cibles

Génération de la base de données cible I

Mapping exécutable → génération d'une requête

Cours (cn, intitulé, credits)
Promo (cn, sem)
Enseignant (id, nom)
Enseigner (id, cn, sem)
Titularité (cn, id)

Une requête possible (cours enseignés par un enseignant)



Génération de la base de données cible II

Mapping exécutable → génération d'une requête

Cours (cn, intitulé, credits)

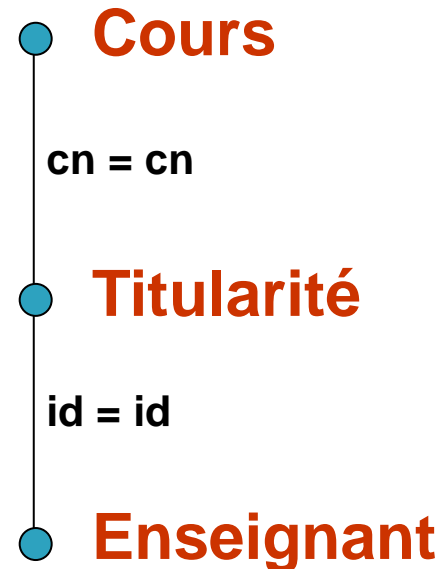
Promo (cn, sem)

Enseignant (id, nom)

Enseigner (id, cn, sem)

Titularité (cn, id)

Une autre requête possible
(cours et les titulaires)



Quelle requête est correcte ? Celle-ci ou la précédente ?

Élimination des erreurs, standardisation de données générées

- ▶ L'échange de données dans le contexte des entrepôts peut être réalisé dans le but **d'améliorer la qualité de données d'origine** ou d'en **standardiser la représentation, suivant les schémas reconstitués des sources**
- ▶ Cela permet de simplifier la successive intégration de données
- ▶ Exemples :
 - Supprimer des espaces, mettre en majuscule, éliminer les accents (standardisation), rajouter ou supprimer des caractères (standardisation/élimination d'erreur)
 - Indiquer une valeur manquante (standardisation), substituer une valeur par une autre (élimination d'erreur)
 - Éliminer les doublons (standardisation)
- ▶ Les mappings permettent d'éviter l'écriture de programmes/requêtes assez longs et articulés

Échanges SQL-Like (Extraction de données pertinentes – Filtrage)

- ▶ Recherche de similarité :
 - Select a,b from T a, S b where **sim**(a,b) > seuil
- ▶ Recherche de données de mauvaise qualité :
 - Select a [ou count(a)] from T where b is null [ou a is null]
 - Select a from T where a not in (...) [ou a > seuil ou a < seuil]
 - Select a, b from T where a > b [ou a + b > seuil, a + n < seuil]
 - Select a from (select a,b from T) group by a having count(a) > 1
 - Select a from T,P p where not **match**(a, p.a)

Échanges SQL-Like (Extraction de données pertinentes – Données extraites)

- ▶ Recodage :
 - `select case when a=v then w when a=v' then w' end from T`
 - `Select F(a) from T`
- ▶ Changement d'unité de mesure-valeur :
 - `select F(a) from T / select F(T.a,S.b) from T,S (join)`
- ▶ Changement de clés-valeur :
 - `Select ID, seq.next from T`
- ▶ Fusion :
 - `select merge(a,b) from T / select merge(a,b) from T, S (join)`
- ▶ Split :
 - `select F1(a), F2(a),... from T`
- ▶ Agrégation :
 - `select OP(a), b from T group by b`
- ▶ Modification :
 - `update T set a=... where`

Échanges non SQL (Extraction de données pertinentes – Données extraites)

```
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('VINET','Vins et alcools Chevalier','59 rue de l'Abbaye','Reims',null,'51100','France');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('HANAR','Hanari Carnes','Rua do Paço, 67','Rio de Janeiro','RJ','05454-876','Brésil');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('VICTE','Victuailles en stock','2, rue du Commerce','Lyon',null,'69004','France');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('SUPRD','Suprêmes délices','Boulevard Tirou, 255','Charleroi',null,'B-6000','Belgique');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('CHOPS','Chop-suey Chinese','Hauptstr. 29','Bern',null,'3012','Suisse');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('HILAA','HILARIÓN-Abastos','Carrera 22 con Ave. Carlos Soublette #8-35','San
Cristóbal','Táchira','5022','Venezuela');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('ERNSH','Ernst Handel','Kirchgasse 6','Graz',null,'8010','Autriche');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('OTTIK','Ottilies Käseladen','Mehrheimerstr. 369','Köln',null,'50739','Allemagne');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('QUEDE','Que Delícia','Rua da Panificadora, 12','Rio de Janeiro','RJ','02389-673','Brésil');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('RATTC','Rattlesnake Canyon Grocery','2817 Milton Dr.','Albuquerque','NM','87110','États-Unis');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('FOLKO','Folk och få HB','Åkergatan 24','Bräcke',null,'S-844 67','Suède');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('FRANK','Frankenversand','Berliner Platz 43','München',null,'80805','Allemagne');
(CODE_CLIENT,SOCIETE,ADRESSE,VILLE,REGION,CODE_POSTAL,PAYS) ('GROSR','GROSELLA-Restaurante','5ª Ave. Los Palos Grandes','Caracas','DF','1081','Venezuela');
```

<<table>>
Ville
ID <<PK>>
Ville
Region
CP
Pays

Utilisation d'une expression régulière en Java pour le CP

"[^\d]*[\d]+([^\d]+)" 2^{ème} numéro ---

"[^\d]*[\d]+([^\d]+[-][\d]+)" numéro suivi de
« - » suivi d'un numéro

Et pour extraire le pays "[^\d]*[\d]+([^\d]+[\d]+(.*))"

Échanges SQL-Like (Extraction de données pertinentes – Données extraites)

- ▶ Les mappings SQL-like fournis comme exemples, peuvent être facilement réalisés par un ETL quelconque
- ▶ Cependant, ces mappings ne prennent pas en considération **2 problématiques majeures** à résoudre lors de la programmation des flux d'extraction
 - Comment réaliser les **extractions successives**
 - Comment prendre en compte des éventuels **changements des clés dans les sources**

Prise en compte des extractions successives

- ▶ Si la source est une base de données et le SGBD est suffisamment complet (peu importe le modèle de données), plusieurs mécanismes peuvent être mis en œuvre
- ▶ Le plus simple (mais difficile à gérer à cause de l'entrelacement transactionnel) est celui utilisant un trigger vers une table contenant les données modifiées ; par exemple

Create trigger ChangementsT after update or insert or delete on T

For each row

Begin declare r T%ROWTYPE; s varchar(20);

If updating then Insert into ChOfT values (:old.key, :new.a, :new.b,..., 'updated'); s='updated';
r.key=:old.key..... end if; /* il est supposé qu'en cas de changement de clés, la nouvelle donnée correspond à l'ancienne donnée

If inserting then Insert into ChOfT values (:new.key, :new.a, :new.b,..., 'inserted'); s='inserted'; r:=new;
end if;

If deleting then Insert into ChOfT values (:old.key, :old.a, :old.b, ..., 'deleted'); s='deleted'; r:=old; end if;

Exception when others update ChOfT set a=r.a; b=r.b, ..., status=s where ChOfT.key=r.key end;

- ▶ La table contenant les changements sera donc vidée par un flux ETL une fois pris en compte ces changements

Prise en compte des extractions successives

- ▶ Une solution plus viable pourrait être celle des « log », si disponibles ; par exemple (ORACLE)

```
create table T  
(a number primary key, b varchar2(19))
```

```
create materialized view log on T with rowid, primary  
key, sequence (b) including new values;
```

```
-- select * from MLOG$_T
```

Prise en compte des extractions successives

- ▶ Si **la source n'est pas une base de données**, la solution ne peut être que spécifique
- ▶ Par exemple, si un fichier texte est utilisé,
 - il faudra probablement comparer l'ancien fichier au nouveau fichier pour repérer les changements ou
 - Limiter la comparaison rajoutant dans la source une information sur la prise en compte par le flux ETL des données contenues
- ▶ Évidemment, il est toujours possible de vider entièrement la zone de staging et refaire une extraction complète mais cette solution pourrait ne pas être possible et pourrait ne pas être scalable

Échange de données dans l'approche « supply driven »

Identification sources	Conception SI	Conception SD	Conception Flux
Analyse (statique) de qualité : choix de sources	Transf Schéma (C→C)	Transf Schéma (C→L;C→C;C→L)	Conception de flux d'extraction : Échange de données visant l'amélioration de la qualité (mappings conceptuels)
Transf Schéma (L→C;C→L)	Intégration schéma (conceptuel)+Transf Schéma (C→L)		Intégration de données (conceptuel)

Conception

Réimplantation sources	Implantation SI	Implantation SD	Programmation flux
Transf Schéma (L→P sources) + déploiement	Transf Schéma : L→ P schéma intégré) + déploiement	Transf Schéma : L-> P, L→ L schéma dimensionnel)+ déploiement	Flux de chargement : Échange de données (chargements successifs)+ déploiement (parallélisation)
			Flux d'intégration + déploiement (parallélisation)
			Flux d'extraction : Échange de données (mappings exécutables et extractions successives)+ déploiement (parallélisation)

Mise en œuvre

Points clé :

- Les échanges en rouge sont complexes ; plus ces échanges sont soignés, plus l'intégration de données sera simple ;
- Les échanges en vert sont généralement plus simples car bien cadrés et sous l'hypothèse de disposer d'outils adaptés