

SGBDR et conception d'un système d'information avec MERISE

Séminaires Codes & Travaux @ IRISA
26 Avril 2007

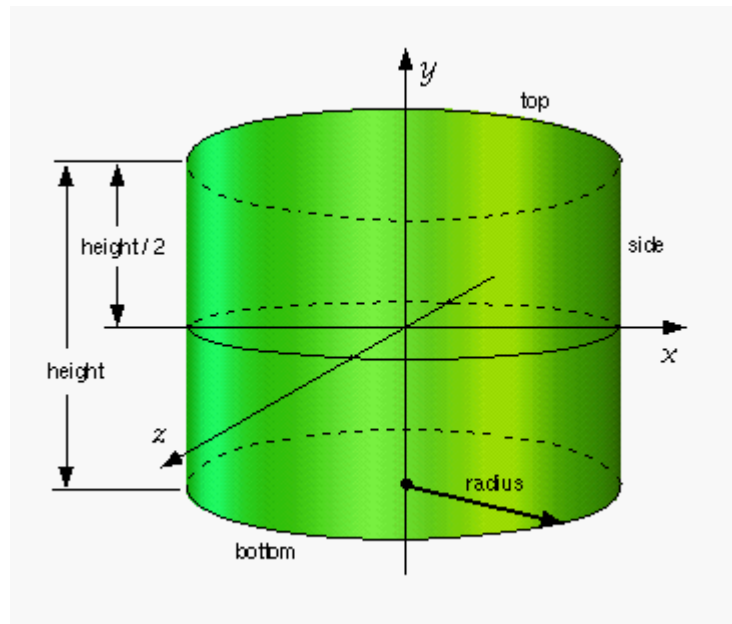
Anthony ASSI
Ingénieur Expert R&D
Plateforme Bio-Informatique / Equipe Symbiose

INSTITUT NATIONAL
DE RECHERCHE
EN INFORMATIQUE
ET EN AUTOMATIQUE



SGBDR : Système de Gestion de Bases de Données Relationnelles

Une Base de Données est un ensemble de données mémorisées sur des supports physiques d'une façon *bien structurée*.



Les couches d'un SGBD

Un SGBD est composé de 3 couches:

- 1) **Systeme de gestion de fichiers** : gestion du stockage physique de l'information (couche dépendante du matériel)
- 2) **SGBD interne** : gestion logique des données (assemblage, placement, liens et accès)
- 3) **SGBD externe** : gestion de l'interface avec l'utilisateur et le concepteur (langage de requêtes, outils de présentation)

Avantages d'une BD

- **Indépendance physique** : un SGBD permet de manipuler des données sans se soucier de la façon dont sont stockées physiquement ces données
- **Indépendance logique** : un SGBD permet de présenter sous différentes vues les mêmes données
- **Efficacité des accès aux données** : un SGBD repose sur des algorithmes (transparents pour l'utilisateur) performants qui optimisent l'accès aux données
- **Sécurité des données** : un SGBD permet de gérer différents niveaux d'accessibilité aux données en fonction de profils d'utilisateurs

Differents types de BD:

- **BD Hiérarchiques** : les plus anciennes fondées sur une modélisation arborescente des données
- **BD Relationnelles** : organisation des données sous forme de table et exploitation à l'aide d'un langage déclaratif (SQL, PL/SQL)
Ex: Oracle, IBM DB2, MySQL, PostgreSQL, etc.
- **BD Déductives** : organisation des données sous forme de table et exploitation à l'aide d'un langage logique (SQL 3)
- **BD Objets** : organisation des données sous forme d'instances de classes hiérarchisées (qui possèdent leur propres méthodes d'exploitation)
Ex: O2 (INRIA), Object Store

Le Modele Relationnel

Concept fondamentaux :

- Tables
- Domaines
- Relation
- Clés

Organisation des données en Table :

- des lignes (enregistrements ou n-uplets)
- des colonnes (champs ou attributs)

Concepts fondamentaux des Domaines :

- Domaine : ensemble de valeurs distinctes que peut prendre une donnée élémentaire ($D1=\{\text{rouge,vert}\}$, $D2=\{3,2\}$)
- Produit Cartésien d'un ensemble de domaine $D1$, $D2$, Dn est l'ensemble des n-uplets $\{<v1,v2,vn>\}$ tels que $vi \in Di$
 $D1 * D2 = \{<\text{rouge},3>, <\text{rouge},2>, <\text{vert},3>, <\text{vert},2>\}$

Le Modele Relationnel : Relation

3/4

Une relation est un sous-ensemble du produit cartésien d'une liste de domaines ; elle est caractérisée par:

- un nom
- des champs (attributs) qui correspondent aux noms de chacun des domaines mis en jeu (les colonnes)
- des n-uplets (tuples) qui correspondent aux différentes combinaisons valides du produit cartésien (les lignes)

Personne

<i>Nom</i>	<i>Prenom</i>
Gosling	James
Assi	Anthony

Le Modele Relationnel : Clé

4/4

Clé d'une relation : un ou plusieurs attributs permettant d'identifier, de façon unique, chaque n-uplets de la relation

Remarque : il peut éventuellement exister plusieurs clés pour une même relation ; dans ce cas, il faut :

- (1) identifier les clés candidates
- (2) choisir une clé primaire parmi les clés candidates

Exemple : ETUDIANT(matricule, nom, prénom, adresse, num-insee)

Clé primaire



Clé candidate



SQL: Structured Query Language

SQL se décompose en 5 types d'ordres :

- **DDL (Data Definition Language)** : permet de modifier la structure de la base de données
- **DML (Data Manipulation Language)** : permet de consulter / modifier le contenu de la base de données
- **DCL (Data Control Language)** : permet de gérer les privilèges, c'est-à-dire les utilisateurs et les actions qu'ils peuvent entreprendre
- **TCL (Transaction Control Language)** : permet de gérer les transactions, c'est-à-dire rendre automatique divers ordres enchaînés en séquence
- SQL procedural : **PSM (Persistent Stored Module)**, **CLI (Call Level Interface)**, Embedded SQL, ... qui est un ensemble d'outils pour que SQL s'interface avec des langages hôtes.

Optimisation de Requetes SQL

1/2

```
Select * form Table1 T1,Table2 T2
```

```
where T1.countrycode='fr' and T2.x=T1.x and T2.sex='F'
```

Optimisation de Requetes SQL 2/2

Select * form Table1 T1,Table2 T2

where T1.countrycode='fr' and T2.x=T1.x and T2.sex='F'

Select * form Table1 T1,Table2 **T2**

where **T2**.sex='F' and T1.countrycode='fr' and **T1**.x=T2.x

Q&A

<http://www.irisa.fr/manifestations/seminaires/codes-travaux>

anthony.assi@irisa.fr



MERISE



Introduction

- MERISE est une méthode de conception, de développement et de réalisation de projets informatiques.
- La méthode MERISE (100% Française) est basée sur la séparation des données et des traitements à effectuer en plusieurs modèles conceptuels et physiques.
- Dialogue entre l'utilisateur et l'informaticien
- Complémentaire à UML.

Plan de l'exposé Merise

1)La Modelisation

2)La démarche Merise

3)La formalisation Conceptuelle

4)La formalisation Organisationnelle

5)La formalisation Operationnelle

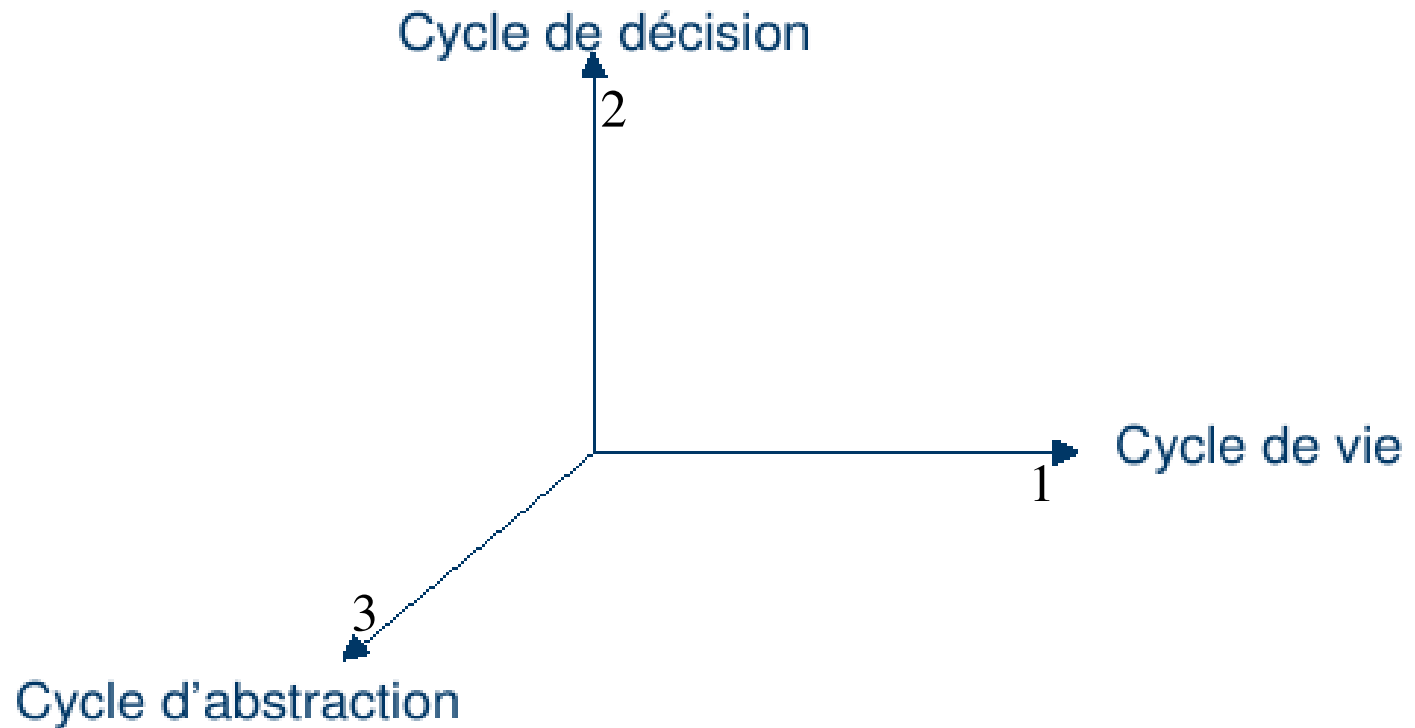
6)Conclusion

La Modelisation

- Qu'est-ce que la Modelisation ?
 - Une technique d'ingénierie permettant de représenter un système; elle s'appuie sur l'établissement de modèles
- Qu'est-ce qu'un Modèle ?
 - Une abstraction simpliste de la réalité, centrée sur la représentation conceptuelle et physique d'un système.
- Pourquoi Modéliser?
 - Nous modélisons des systèmes complexes parce que nous sommes incapables de les comprendre dans leur totalité.
- Qu'apportent les Modèles?
 - Un support de raisonnements et de simulations, ainsi qu'un vecteur de communication entre personnes différentes.

La démarche Merise

Appui sur 3 axes:



Le cycle de vie

- Le schéma directeur
- L'étude préalable (MCT)
- L'étude détaillée (MCD + MOT + MLD)
- L'étude technique (MPD)
- La réalisation
- La mise en oeuvre
- La maintenance

Le cycle de décision

- Identification
- Gestion
- Organisation
- Technique
- Economique

Le cycle d'abstraction

- Le niveau *conceptuel*
- Le niveau *organisationnel*
- Le niveau *opérationnel*

La formatlisation conceptuelle

- Fixer les choix des informations et traitements à manipuler dans le système d'information

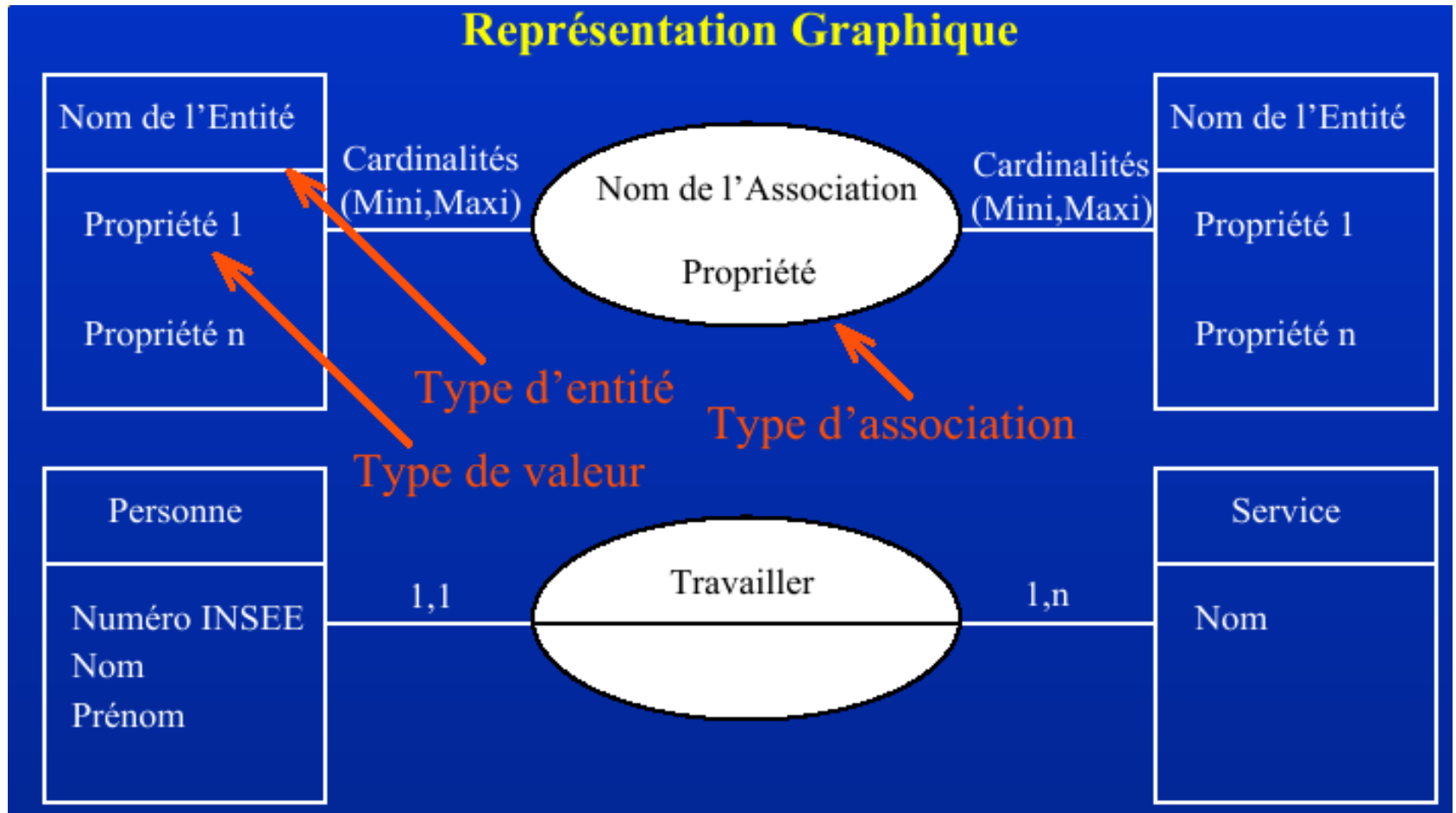
- Deux méthodes de formalisation sont utilisées:
 - Le Modèle Conceptuel des Données (MCD)
 - Le Modèle Conceptuel des Traitements (MCT)



MCD: Modèle Conceptuel des Données

- Description statique du système d'information visé à l'aide d'**entités**, d'**associations** et de **propriétés**
 - Entité : un objet matériel ou immatériel ayant une existence propre et conforme aux choix de gestion du système étudié : personne, voiture, client
 - Association : un lien entre des entités (binaire, ternaire, n-aire)
 - Propriété : une donnée élémentaire et indécomposable du système d'information : couleur d'une voiture, âge d'une personne

Le MCD

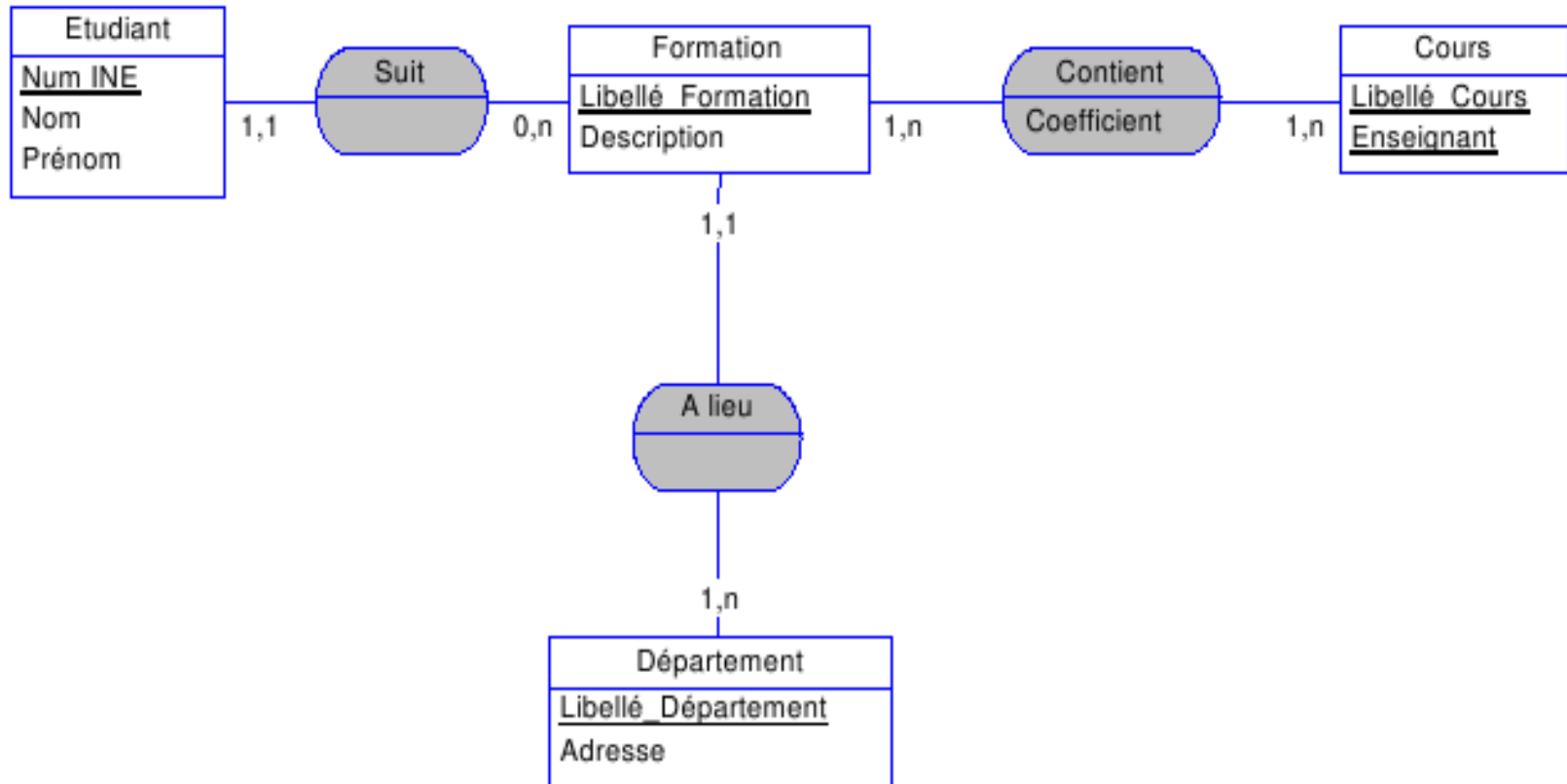


Cardinalités

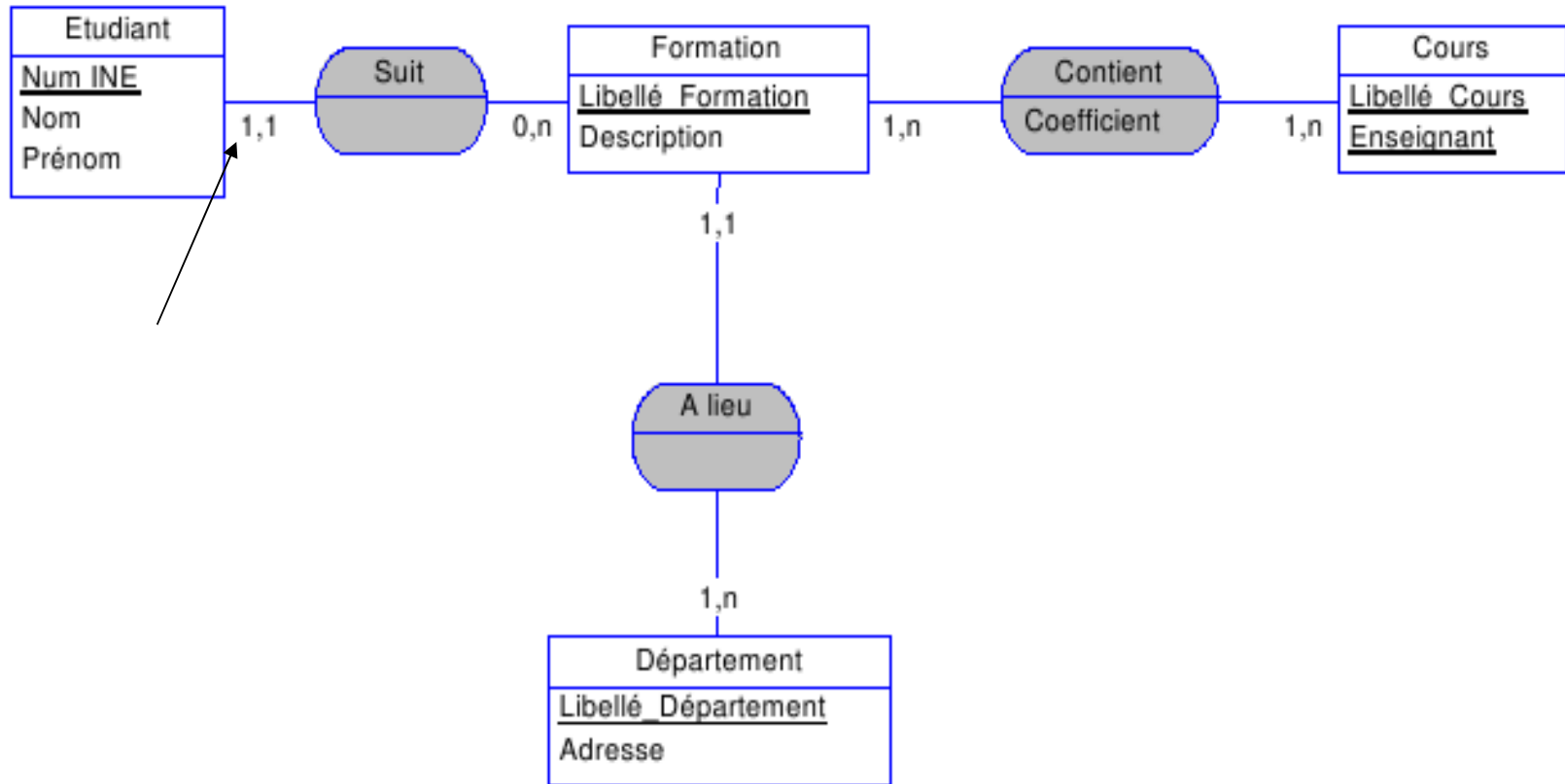
Les cardinalités définissent des contraintes d'intégrités sur l'association

- borne minimale : nombre minimum de fois qu'une occurrence d'une entité participe aux occurrences de l'association (0, 1, n)
- borne maximale : nombre maximum de fois qu'une occurrence d'une entité participe aux occurrences de l'association (1, n)

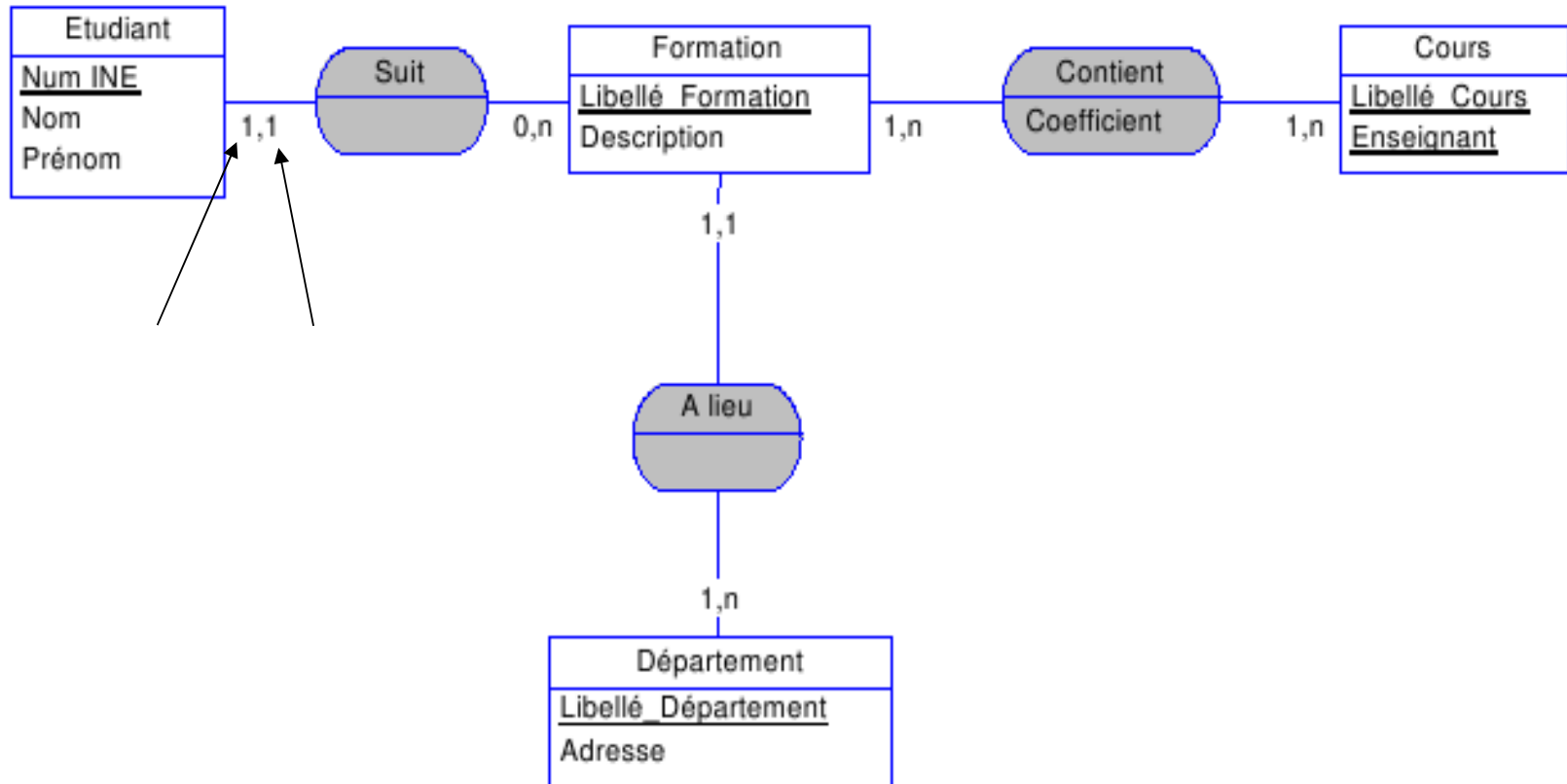
Exemple de Cardinalités du MCD



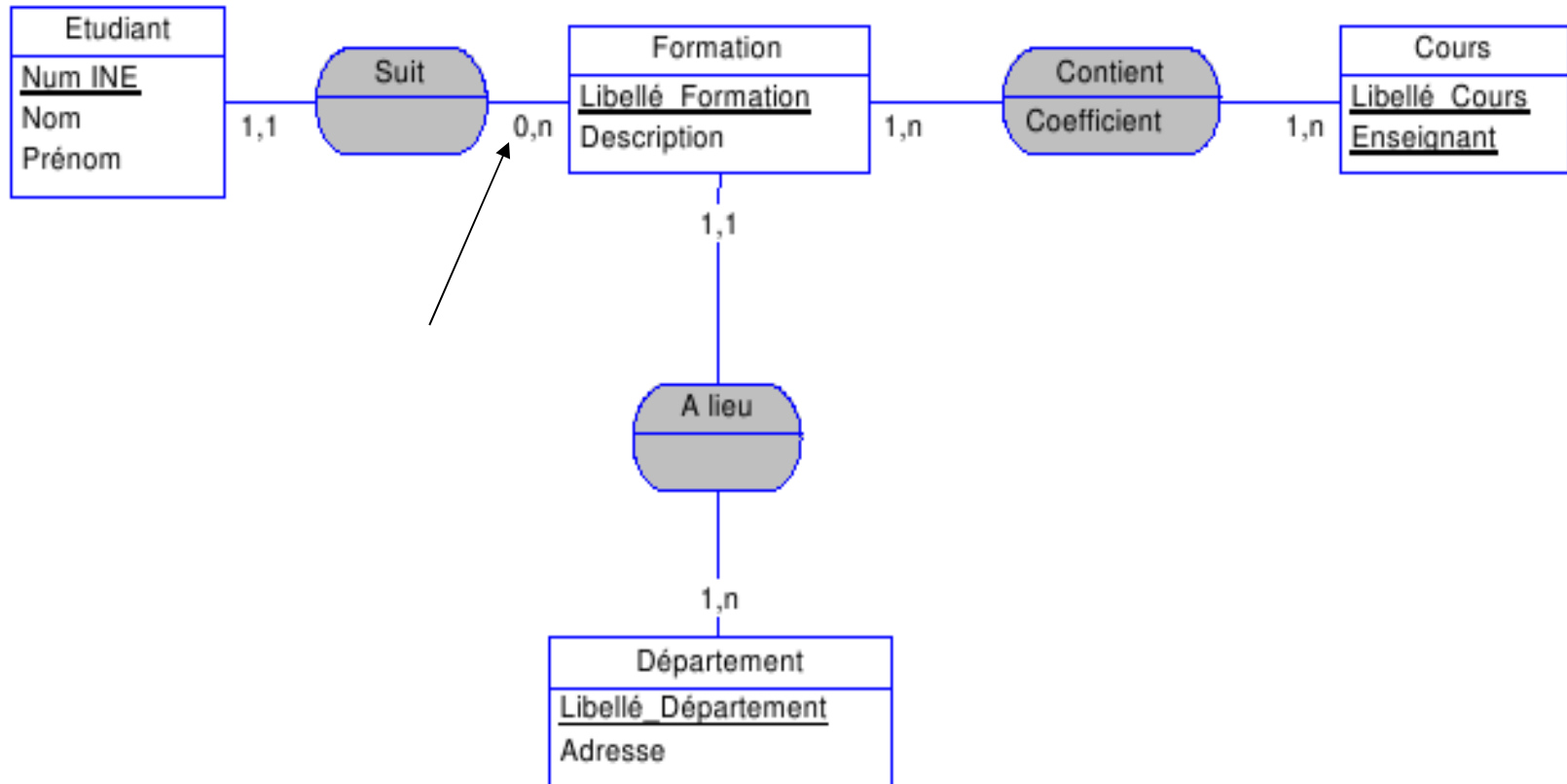
Exemple de Cardinalités du MCD



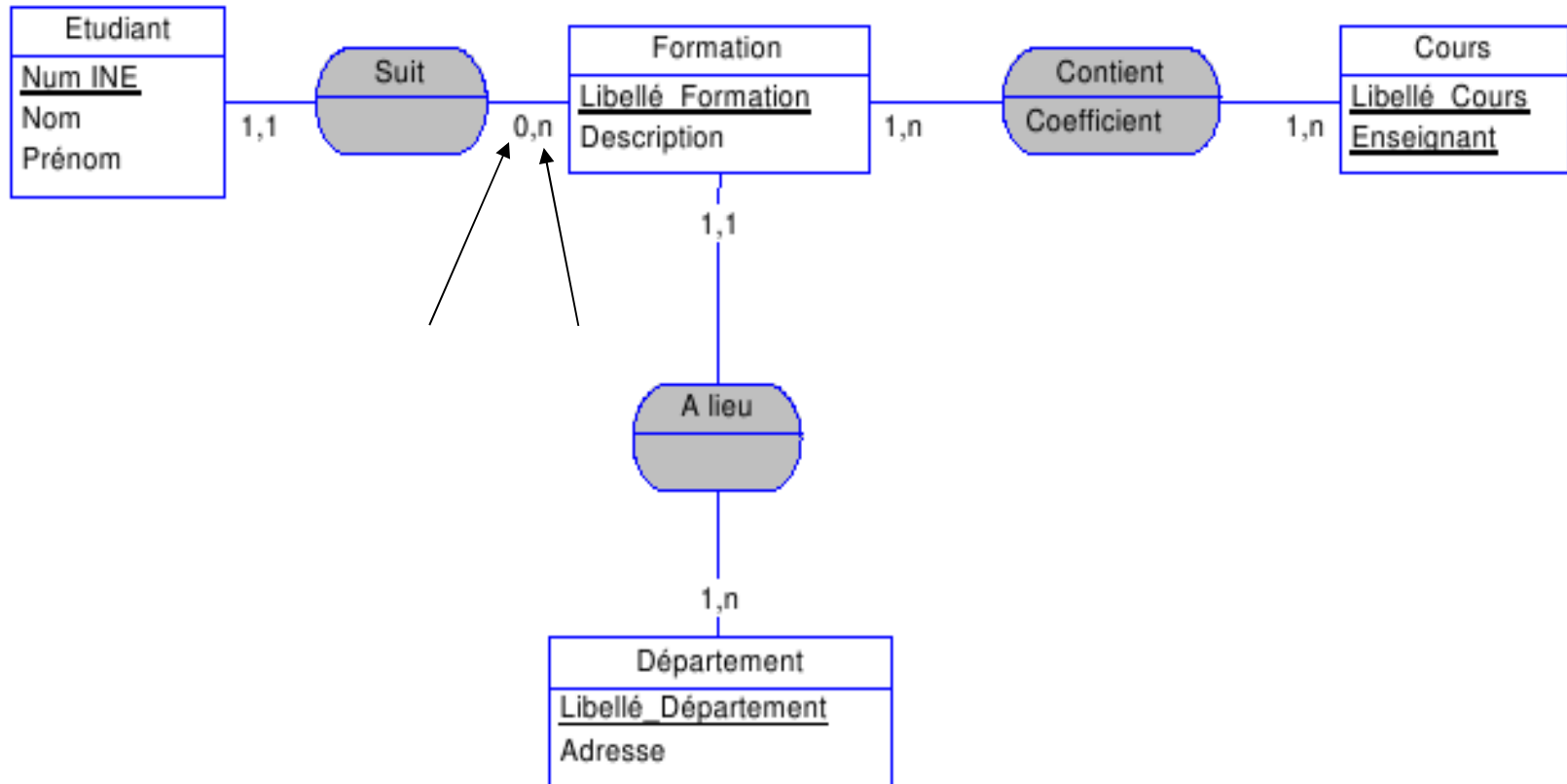
Exemple de Cardinalités du MCD

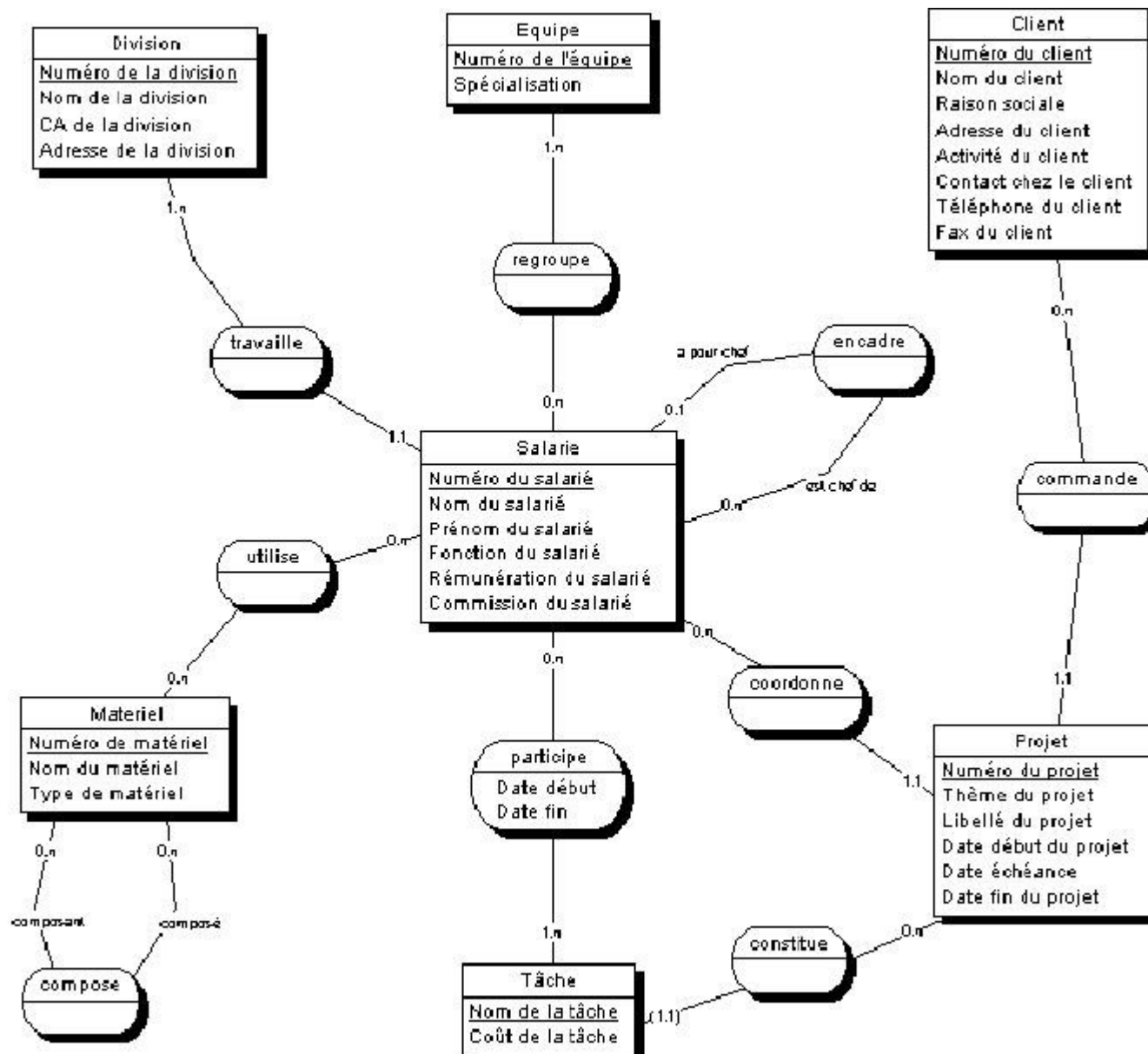


Exemple de Cardinalités du MCD



Exemple de Cardinalités du MCD





Du MCD aux tables Relationnelles

- Traitement des Entités
 - chaque entité devient une table
 - chaque propriété d'une entité devient une colonne
 - L'identifiant de l'entité devient la clé primaire
- Traitement des Associations diffère selon la cardinalité
 - Association avec au moins une cardinalité $(-,1)$
 - Association avec deux cardinalités $(-,n)$ & $(-,n)$

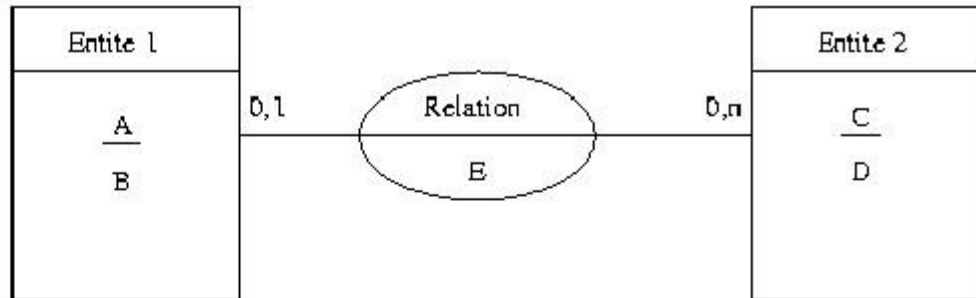
- Cas 1 : association avec au moins une cardinalité maximale à 1
 - Création de 2 tables (une par entité) avec migration de la clé primaire de l'entité de cardinalité (-,n) dans la table de l'entité de cardinalité(-,1)
 - S'il existe des propriétés dans l'association, elles migrent comme précédemment

Example: Cas 1

**Schéma
Entités/Associations**



**Tables relationnelles
correspondantes**



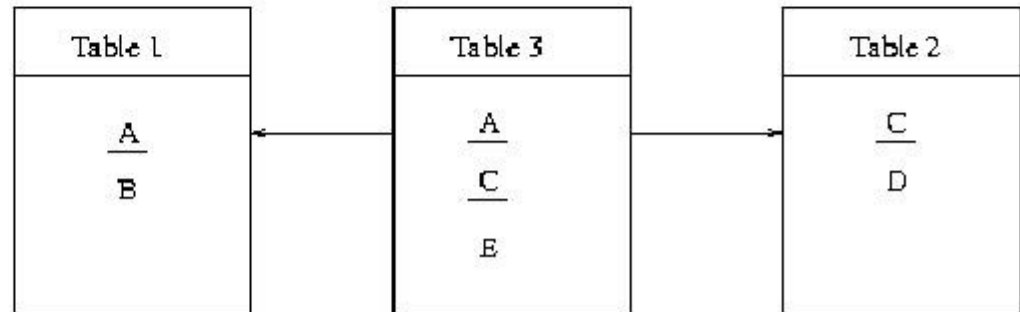
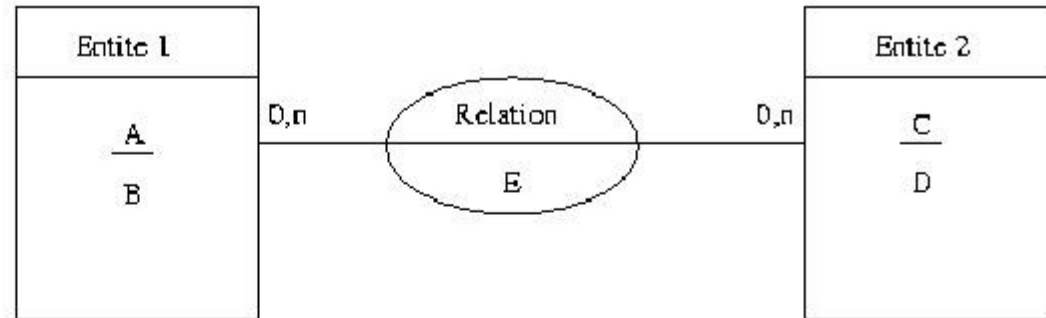
- Cas 2 : association avec les deux cardinalités maximales à n
 - création de 3 tables (une par entité + une pour l'association) avec migration des clés primaires des 2 entités dans la table de l'association
 - S'il existe des propriétés dans l'association, elles migrent dans la table de l'association

Example: Cas 2

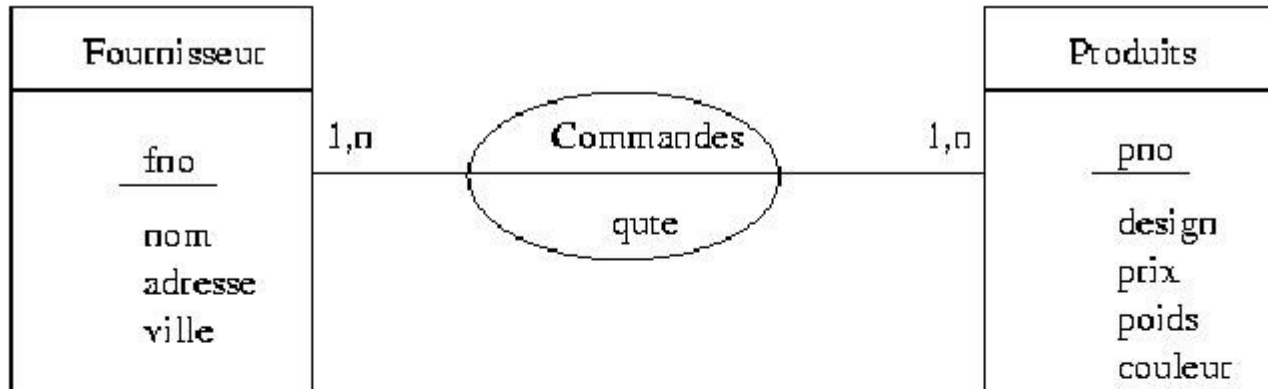
Schéma
Entités/Associations



Tables relationnelles
correspondantes



Exemple Produits-Fournisseurs-Commandes



MCD Fournisseur-Produits-Commandes

Tables relationnelles + intégrité référentielle

Produits (pno, design, prix, poids, couleur)

Commandes (fno, pno, qute)

Fournisseurs (fno, nom, adresse, ville)



MCT: Modèle Conceptuel de Traitements

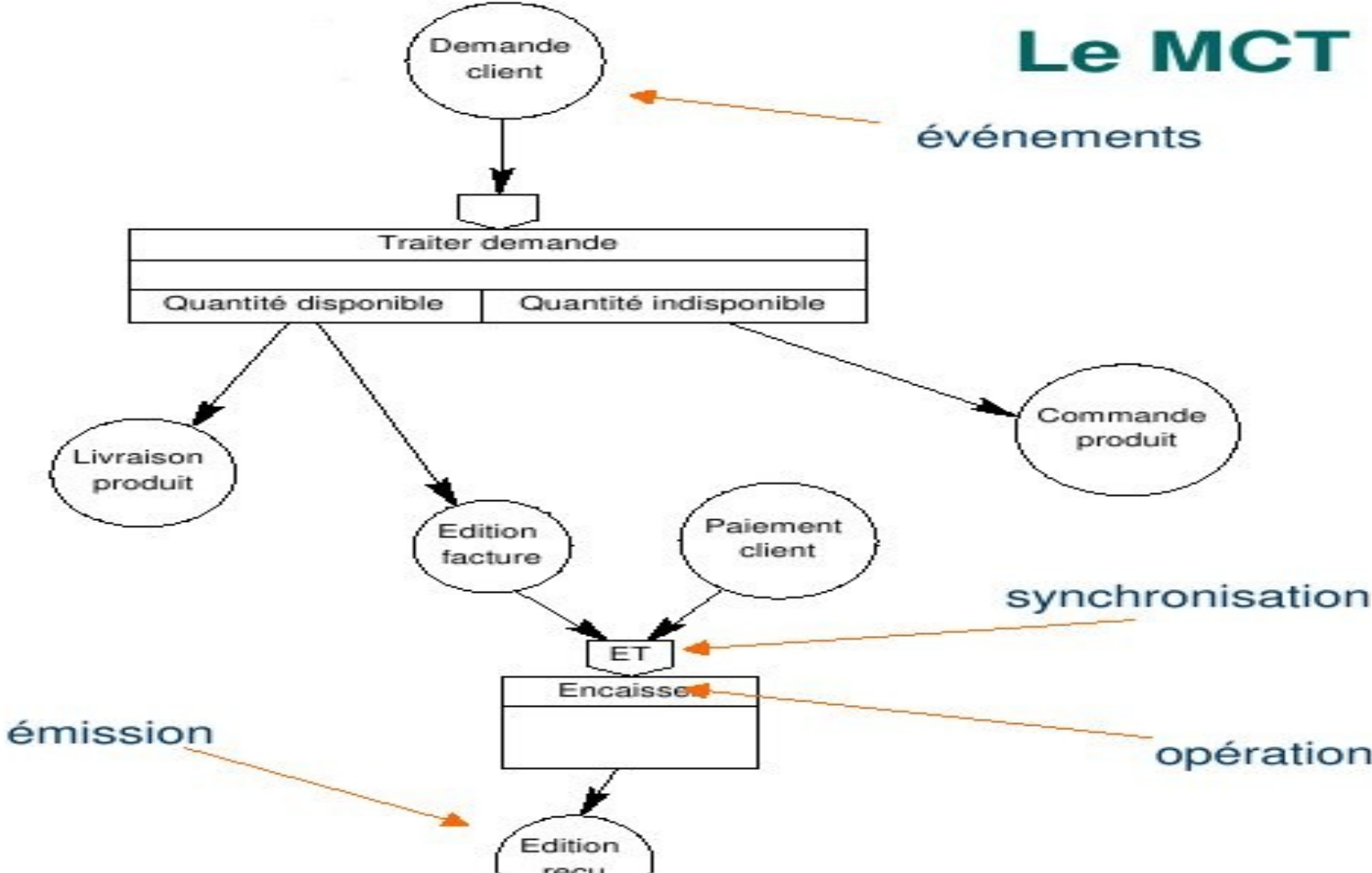
Le MCT est un schéma représentant les traitements, en réponse aux événements à traiter.

Le MCT repose sur les notions d'*événement* et d'*opération*, celle de *processus* en découle.

- **L'événement:** Un événement est assimilable à un message porteur d'informations donc potentiellement de données mémorisables
- **L'opération:** Une opération se déclenche uniquement par le stimulus d'un ou de plusieurs événements synchronisés
- **Le processus:** Un processus est une vue du MCT correspondant à un enchaînement pertinent d'opérations du point de vue de l'analyse

Ex MCT : la prise en compte de la commande d'un client

Le MCT



La formalisation organisationnelle

- Spécifier l'organisation qui régira les données et traitements

- Deux méthodes de formalisation sont utilisées:
 - Modèle Organisationnel des Traitements (MOT)
 - Modèle Logique des Données (MLD)

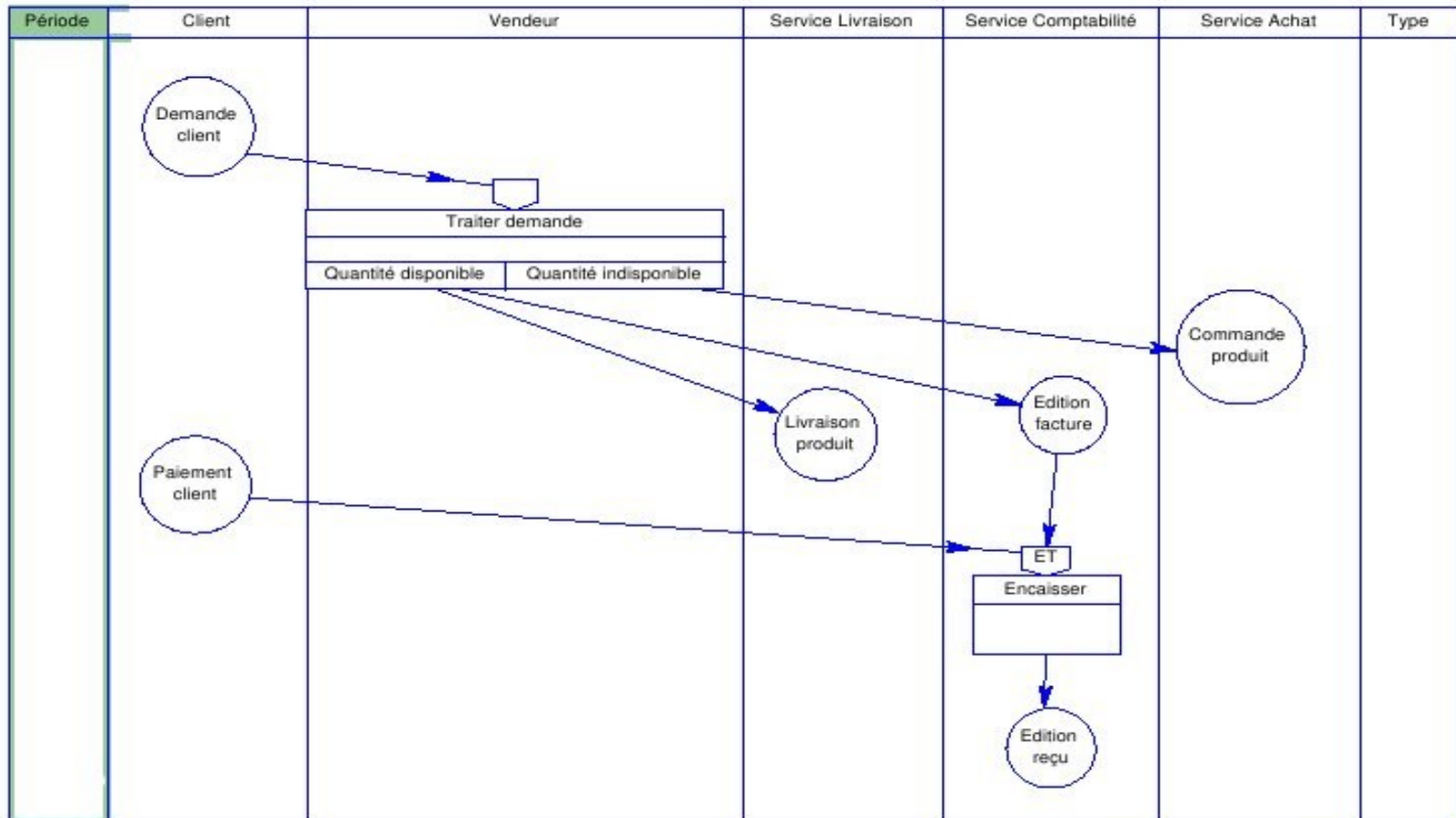
MOT : Modèle Organisationnel des Traitements

Complète le MCT avec les notions suivantes :

- Le Temps
- Les Acteurs
- Le types d'opération

Le MOT

MOT

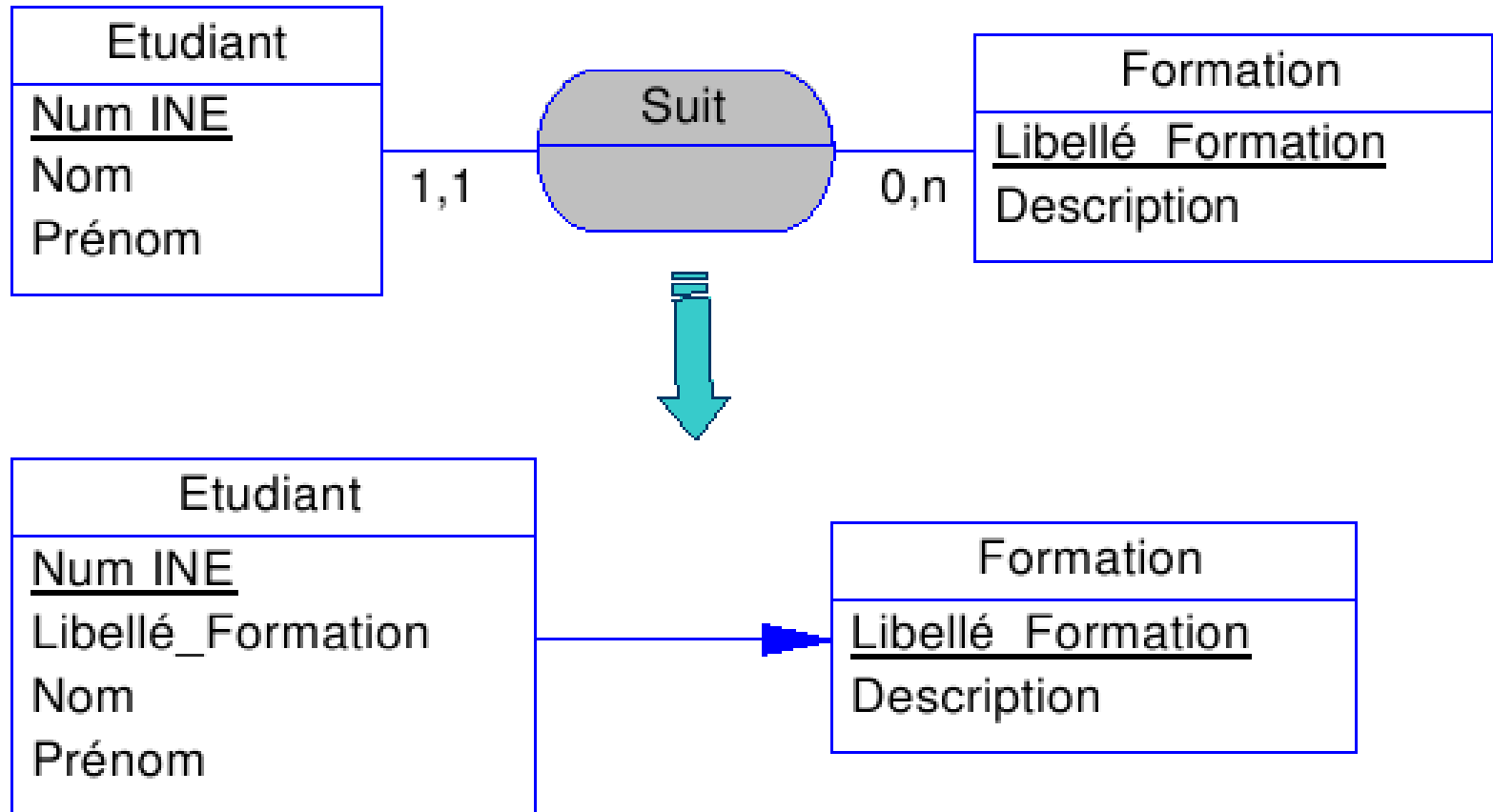


MLD : Modèle Logique des Données

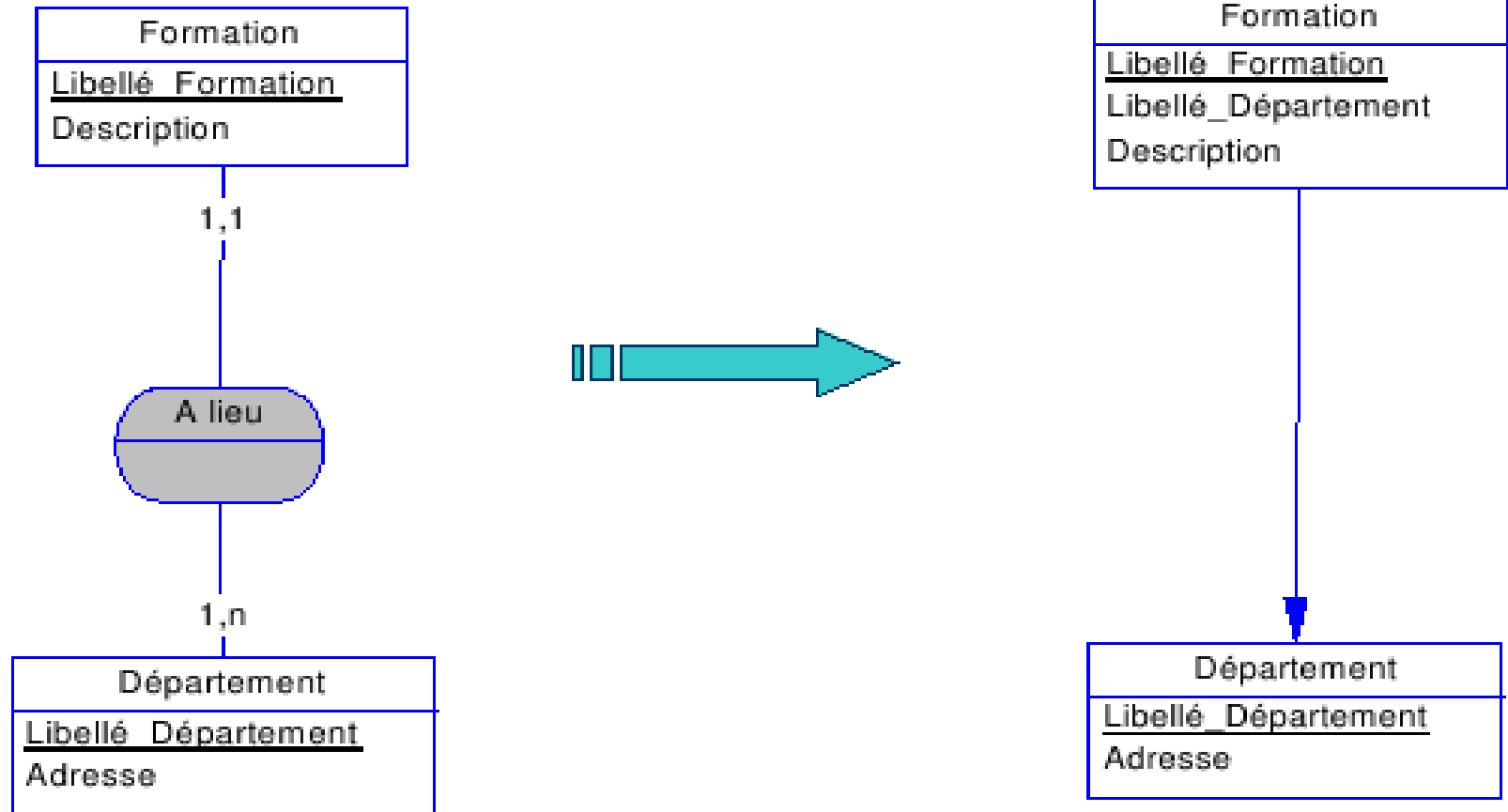
Issu du MCD, le MLD reprend le contenu du MCD, afin de mieux organiser les données, en précisant:

- La volumétrie
- La structure
- L'organisation des données

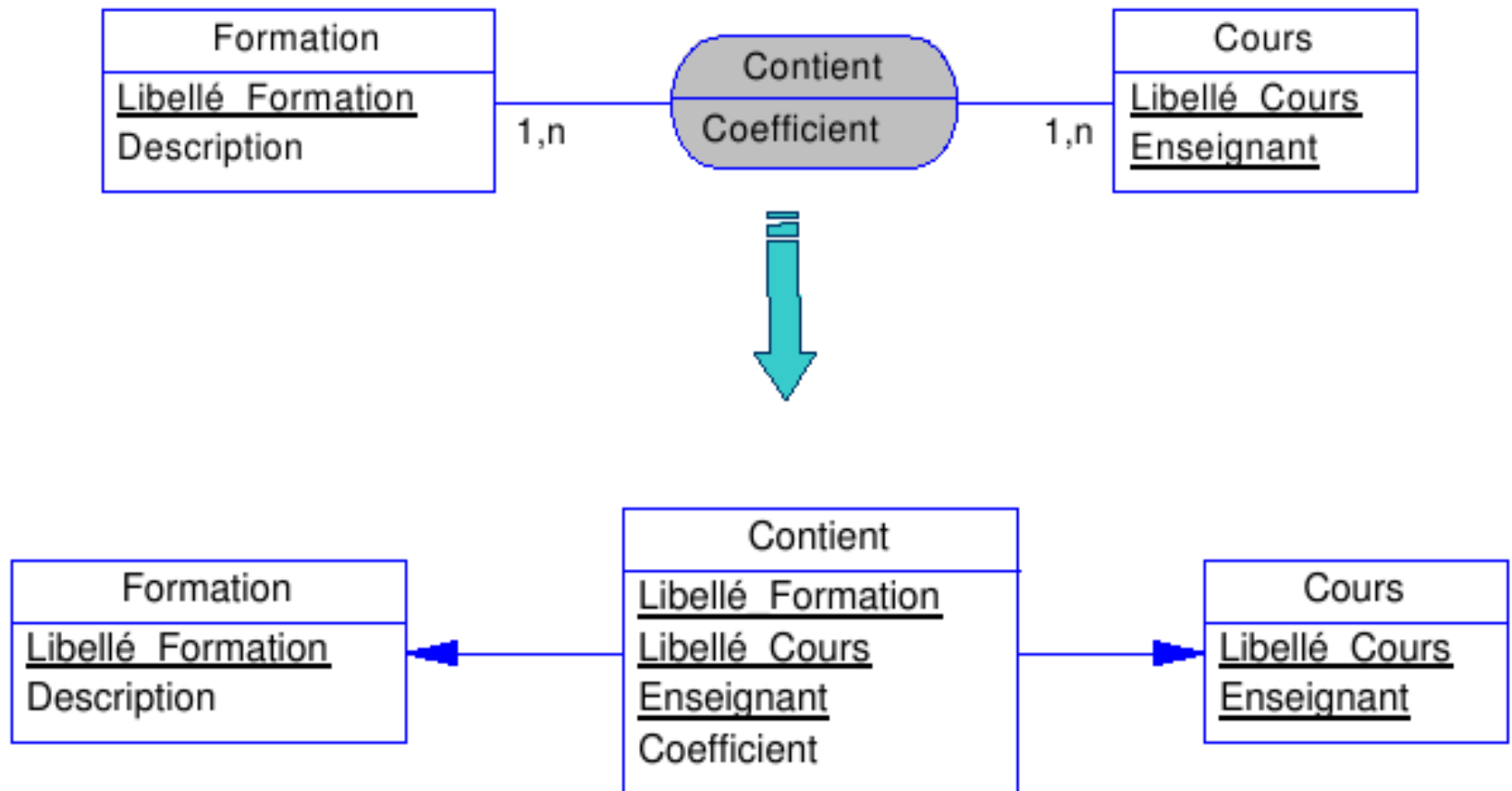
MLD : Exemple pour une base de données relationnelle 1/4



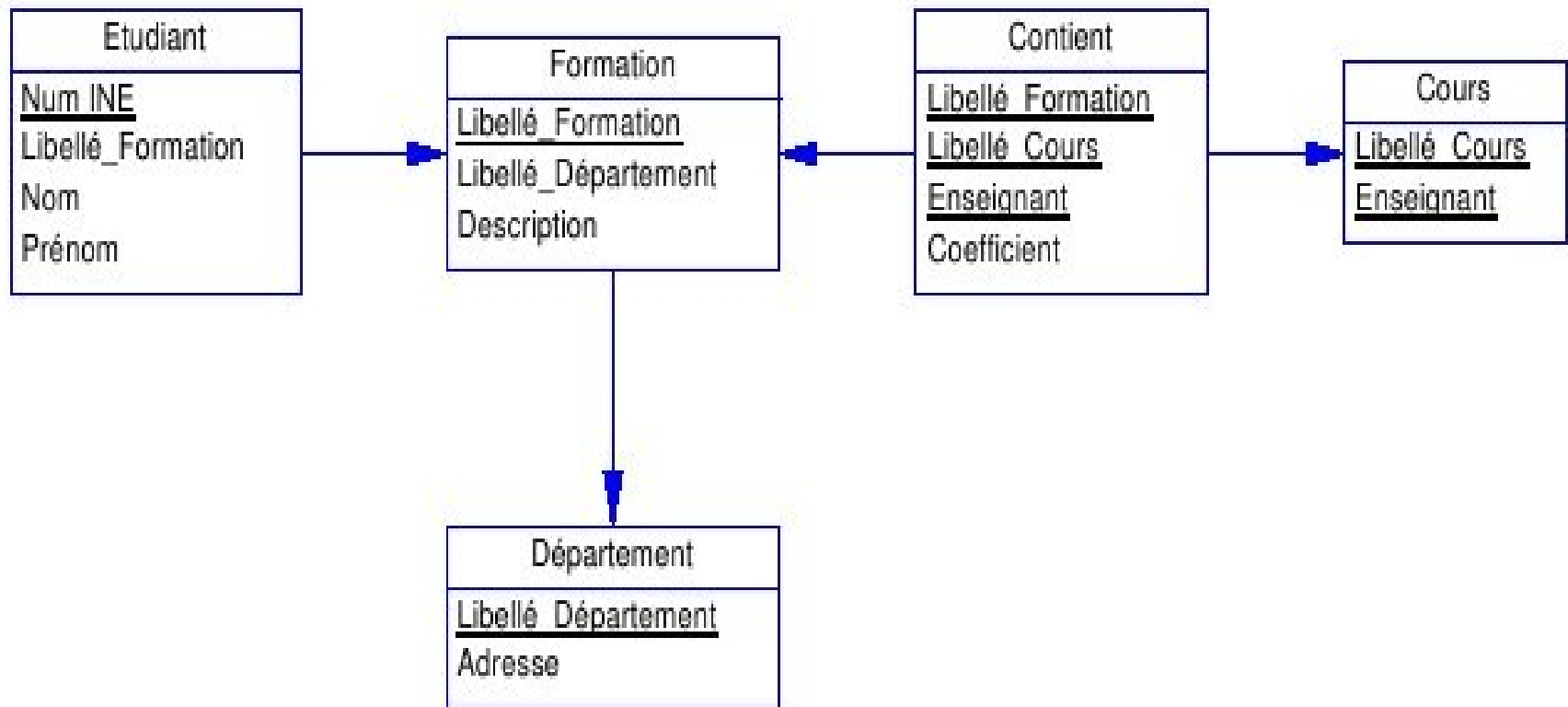
MLD : Exemple pour une base de données relationnelle 2/4



MLD : Exemple pour une base de données relationnelle 3/4



MLD : Exemple pour une base de données relationnelle 4/4



La formalisation opérationnelle

- Spécifier comment seront réalisés les éléments du projet

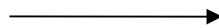
- Deux méthodes de formalisation sont utilisées:
 - Modèle Physique des Données (MPD)
 - Modèle Opérationnel des Traitements (MOPT)

MPD : Modèle Physique des Données

Issu du MLD, le **Modèle Physique des Données**, reprend le contenu du MLD, en précisant le type de données ainsi que les systèmes de stockage employés.

MPD : Modèle Physique des Données

Etudiant		
Num_INE	Long	*
Libellé_Formation	Varchar(100)	
Nom	Varchar(100)	
Prenom	Varchar(100)	



Formation		
Libellé_Formation	Varchar(100)	*
Description	Varchar(400)	
Libellé_Département	Varchar(100)	

MOPT: Modèle OPérationnel des Traitements

Le **M**odèle **O**pérationnel des **T**raitements permet de spécifier les fonctions telles qu'elles seront ensuite réalisées par le programmeur.

MOPT: Modèle OPérationnel des Traitements

boolean *setEtudiant*(String Nom,String Prenom,int idFormation)

Etudiant *getEtudiant*(int idEtudiant)

MERISE V/s UML

Les Atouts :

- Merise est plus adaptée à la conception d'une BD
- UML est plus adapté à la conception d'une Application

<i>Modèle MERISE</i>	<i>Diagramme UML</i>
MCD	Diag de Classes
MCT	Diag d'activités
MOT	Diag de Sequences
MLD	Diag d'objets
MOPT	
MPD	
	Diag de cas d'utilisation
	Diag d'etats transitions
	Diag de collaboration
	Diag de composants
	Diag de deploiement

Conclusion

- Merise est mieux adapté a la Modelisation d'une BD que UML.
- Une bonne conception avec Merise peut vous faire gagner énormément de temps lors d'une future evolution du Modèle.

Q&A

<http://www.irisa.fr/manifestations/seminaires/codes-travaux>

anthony.assi@irisa.fr



A vous d'essayer:

Modeliser en MERISE la Base de Données du système de réservation des Billets TGV-SNCF. (pensez à implementer les programmes de fidélités et l'accumulation des Smiles.)

