Merise

Le cours

- **Le cours comporte plus de 50% d'exercices à faire**
  - En groupe pour les MCD
    - Pour apprendre à travailler en équipe
  - Seul pour les MLD
- **Pendant la semaine il y aura**
  - 1 ou 2 exercices notés → MCD
  - 1 exercice noté → MLD

La Méthode Merise

- **Merise c'est :**
  - Un système franco français
  - Un ensemble de méthodes greffées les unes aux autres
  - Vient du MERISIER
    - Un tuteur sur lequel on greffe différents arbres fruitiers
  - C'est la démarche de construction d'un système d'information
    - Analyse
    - Conception
    - Réalisation, gestion

Méthode Merise : historique

- **Initiée en 1974 à la demande du ministère de l'industrie**
- **Cette méthode a été appliquée en 1979 – 1980**
  - Informatisation massive des organisations gouvernementales
    - Ministères
- **C'est un travail d'anticipation**
  - Prépare les développements informatiques
  - Chiffre (en cout , en temps , en énergie) les projets quelque soit leur taille
    - Chiffré en jour/homme

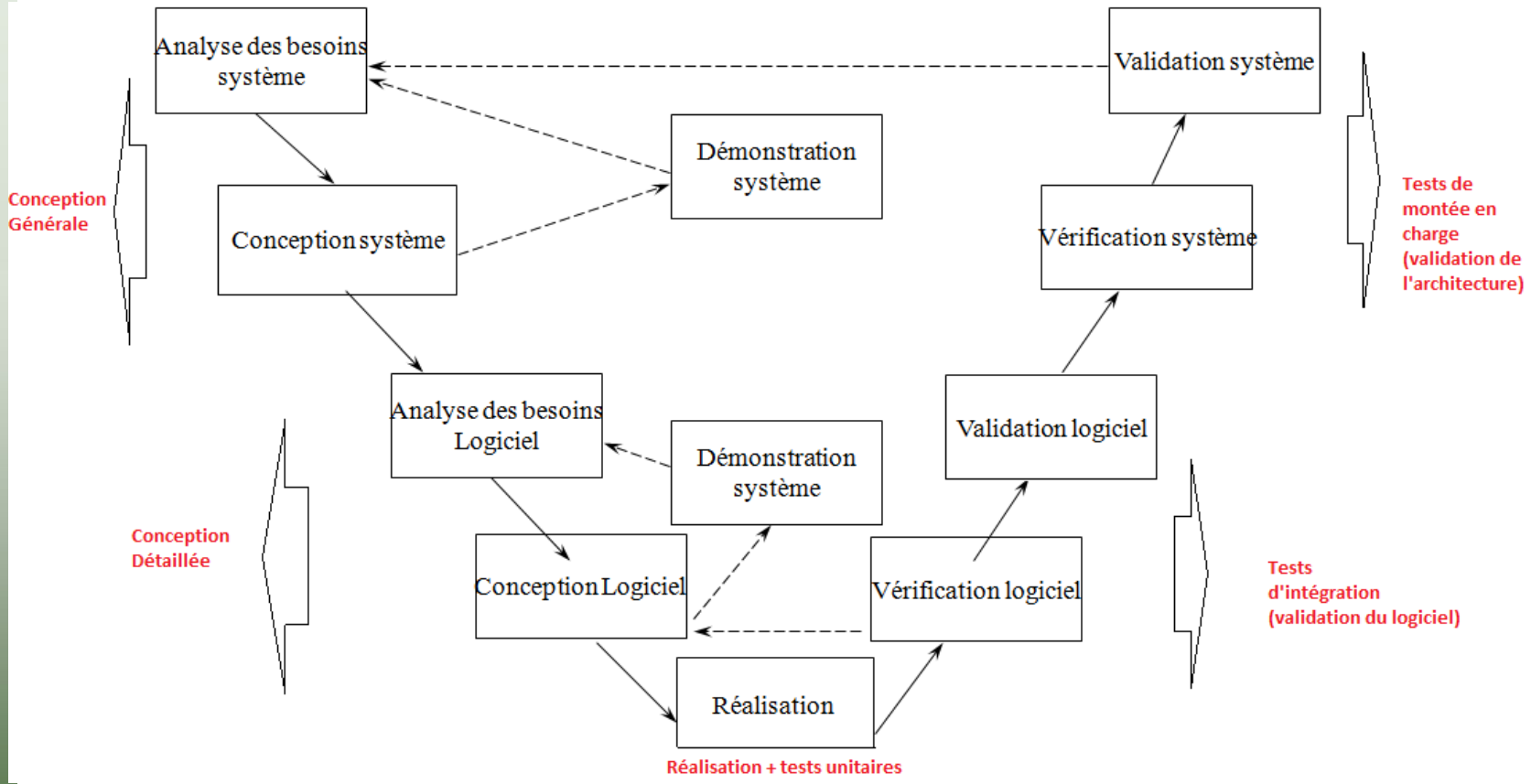
Méthode Merise

- **Les différentes phases de conception**
  - Étude préalable
    - C'est l'étude de l'Existant ,
      - on pose les premières pierres du projet futur
    - Pour le projet futur,
      - le graphe de circulation de l'information (MCC)
      - le quoi (premiers MCD, MCT)
      - Scenario (cout, organisation)
  - Etude générale
    - Le quoi (MCD général , MCT général)
      - Plus détaillés qu'en étude préalable
    - Maquette de l'application
      - Permet de dialoguer avec l'utilisateur

Méthode Merise

- **Différentes phases de conception (suite)**
  - Etude détaillée
    - Le qui, le ou, le quand (MOD , MOT)
    - Le MLD en fin d'étude détaillée
      - C'est la fin de la conception
  - Réalisation
    - Niveau physique (MPD)
    - Niveau opérationnel (logiciel)

Cycle de vie de MERISE



Les niveaux d'abstraction

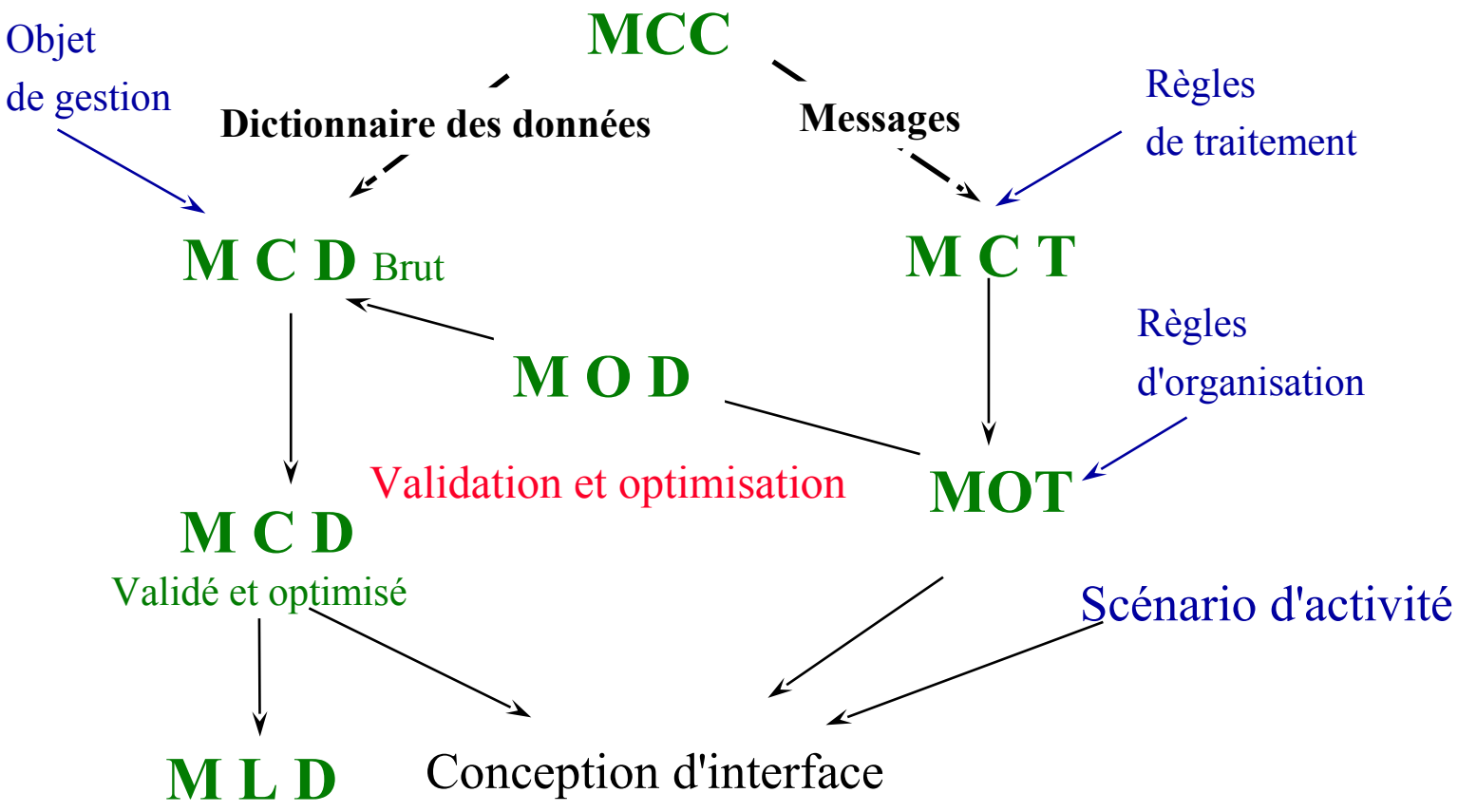
- **Il en découle les niveaux d'abstraction**
  - Niveau conceptuel → étude générale
  - Niveau organisationnel → étude détaillée
  - Niveau logique → fin étude détaillée
  - Niveau physique → réalisation

Les modèles de Merise

INTERFACES	COMMU-NICATION	DONNEES	TRAITEMENT
CONCEPTUEL	MCC	MCD	MCT
ORGANISA-TIONNEL	MOC	MOD	MOT
LOGIQUE		MLD	

Le MCC est le premier modèle permettant la réalisation des MCD et MCT

Le cycle d'abstraction



- **Le MCC est le premier modèle permettant la réalisation des MCD et MCT**

Les niveaux de préoccupation

- **Niveau Système**
  - Solution dans le domaine de l'information
    - Les données
- **Niveau logiciel**
  - Solution dans le domaine de l'informatique
    - Les traitements

Qui intervient

- **Pour réaliser la conception en suivant la méthode Merise**
  - On interview notre utilisateur
    - C'est la maitrise d'ouvrage
    - C'est l'utilisateur qui exprime son besoin
  - C'est l'informatique (les concepteurs) qui réalisent le projet
    - C'est la maitrise d'œuvre
    - La maitrise d'œuvre doit répondre à la demande utilisateurs

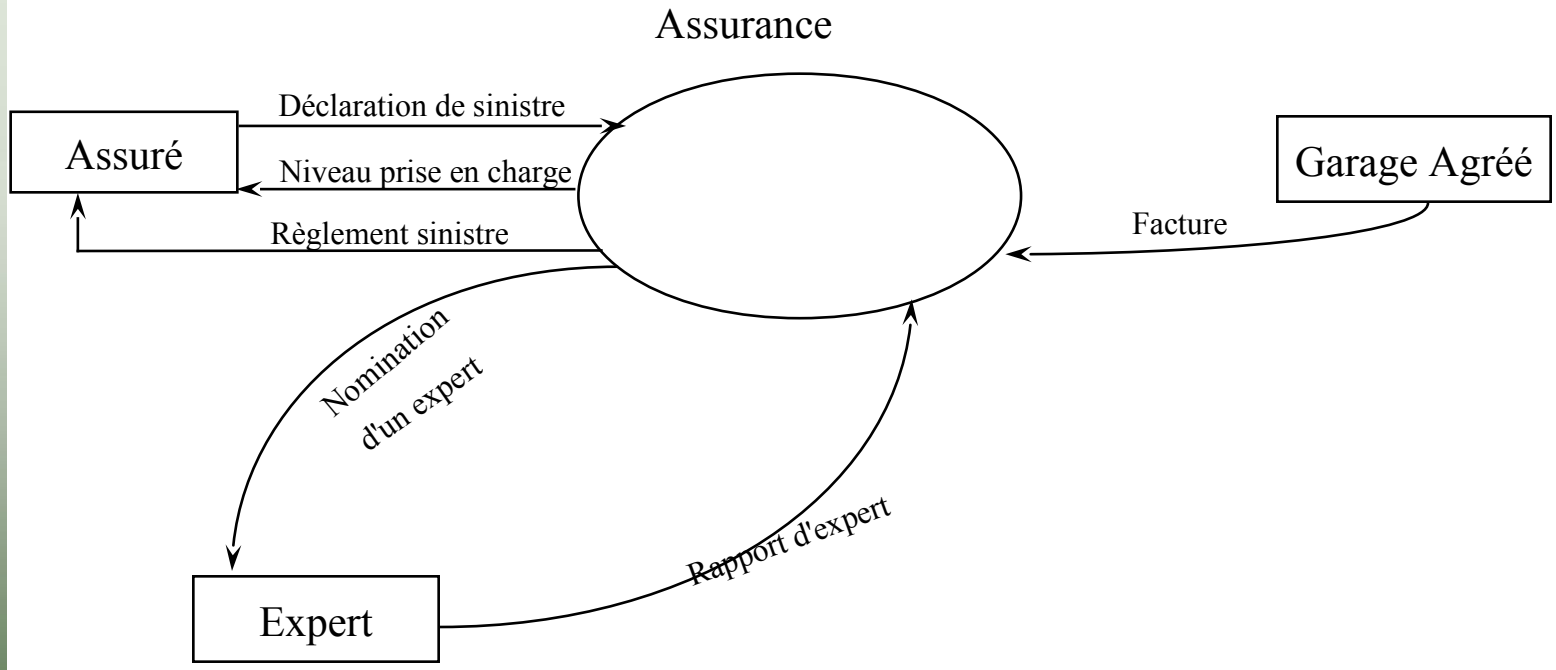
Modèle Conceptuel de communication

- **Le MCC est le premier modèle**
  - C'est lui qui permettra la réalisation des MCD et MCT
- **Définition**
  - Un MCC détermine, par affinage successifs des activités, la composition du domaine d'étude sans en décrire le comportement
- **Le MCC se construit par raffinement successif**

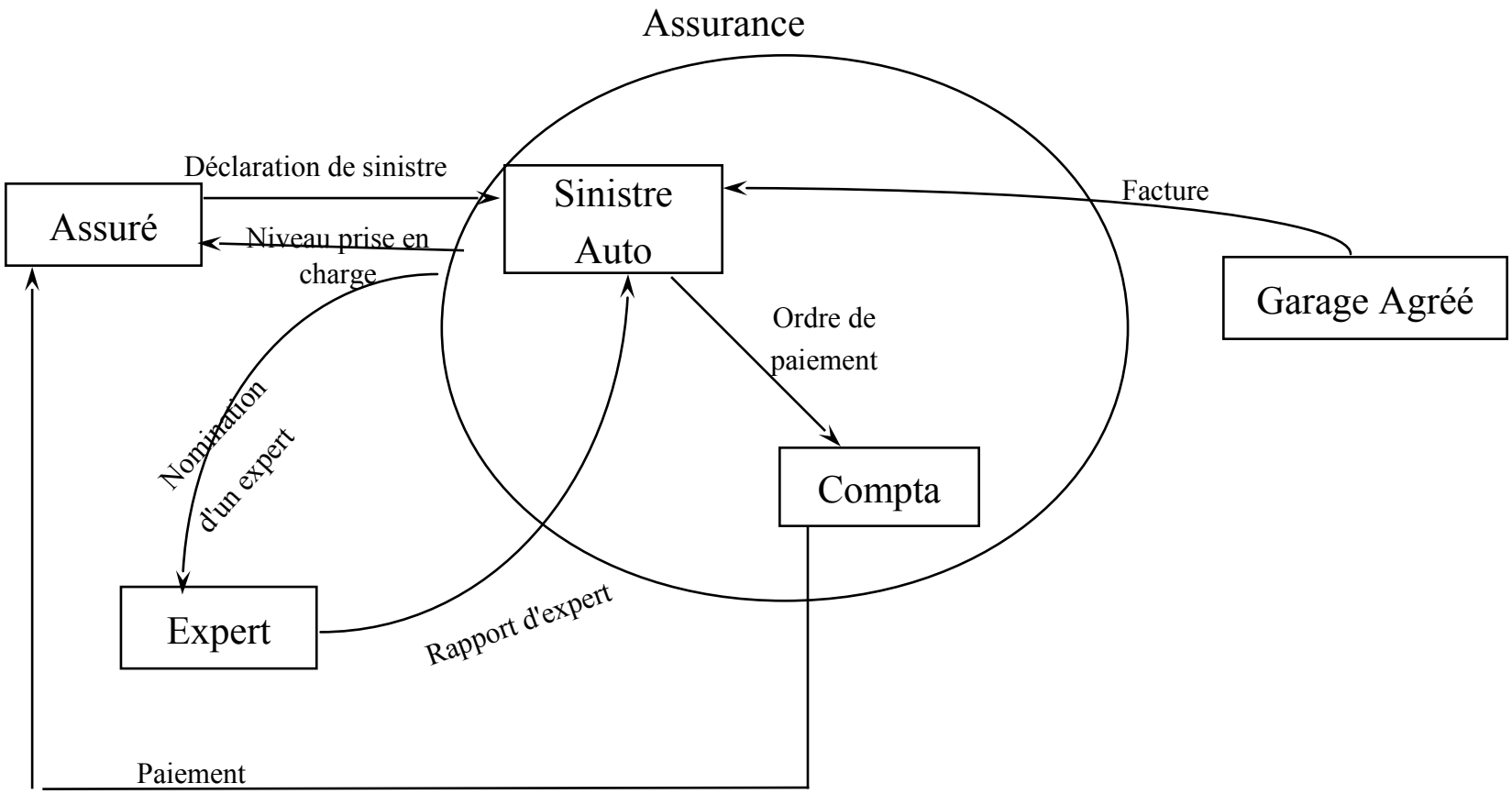
Modèle conceptuel de communication (MCC)

- **Définition :**
  - Le MCC détermine le domaine d'étude et ses échanges avec l'environnement
- **Concepts associés**
  - Domaine d'étude
    - représenté sous forme d'ellipse
    - C'est le domaine à informatiser (le projet informatique)
  - Acteurs externes
    - représentés sous forme de rectangles
    - Un acteur est une personne, un service ou une application (un système)

Exemple de MCC
Niveau 0



Exemple de MCC
Niveau 1



Les concepts associés

Domaine d'étude

- Sous ensemble cohérent de l'entreprise ou de l'organisme, bien délimité et formant le contenu du sujet à étudier

Activité

- Ensemble de traitements homogènes qui transforment ou manipulent des données

Message

- Représentation d'un échange d'informations entre deux composants du système ou entre un composant du système et un système extérieur

Acteur externe

- Source ou destination de données située en dehors du système étudié

Gammes opératoires

Objectifs

- Partitionner le domaine étudié en activités
- Point de passage obligé pour modéliser les traitements
- Maitriser la progression vers le détail du système

Niveau de détail

- On s'arrête quand l'activité correspond à une opération.

Démarche

- Identifier les flux de données entrant et sortant du domaine
- Identifier les activités
- Raffiner par conservation ou décomposition

Merise coté données

- **Dans notre cours nous n'étudierons que la partie**
  - données (MCD)
  - Nous n'étudions pas la partie traitements (MCT)

Le Modèle conceptuel de données (MCD)

- **A partir du MCC nous allons construire le MCD**
- **Pour cela nous allons récupérer la totalité des documents échangés entre les acteurs et le domaine à informatiser**
  - Valider les données à conservées dans notre base de données
  - C'est le dictionnaire de données
    - Liste des informations avec leur définitions et quelques contraintes
  - Ces données vont apparaitre en tant qu'informations dans le MCD
  - Le MCD est la représentation de notre future base de données

Exercice

CAS 8 : ACCIDENT

- OBJECTIF
- Construire le dictionnaire de données à partir des documents ci-dessous :
  - Constat amiable d'accident
  - Déclaration d'accident

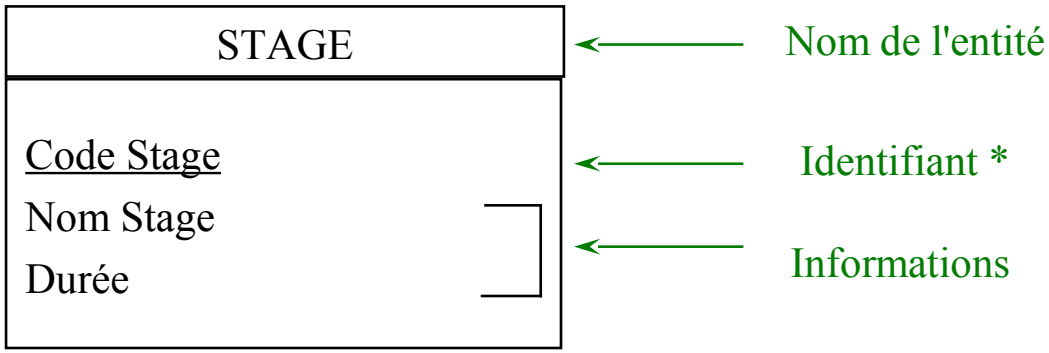
Ne conserver que les informations nécessaires au projet

Modèle Conceptuel de Données (MCD)

- **Vision Statique du Système d'Information**
- **Représentation sémantique des données**
- **Modèle Entité / Relation**

Entités

- Une entité est représentée par un rectangle
- Elle porte un nom représentatif des informations qu'elle contient



* Représente l'unicité de chaque occurrence d'entité
* L'identifiant est toujours souligné

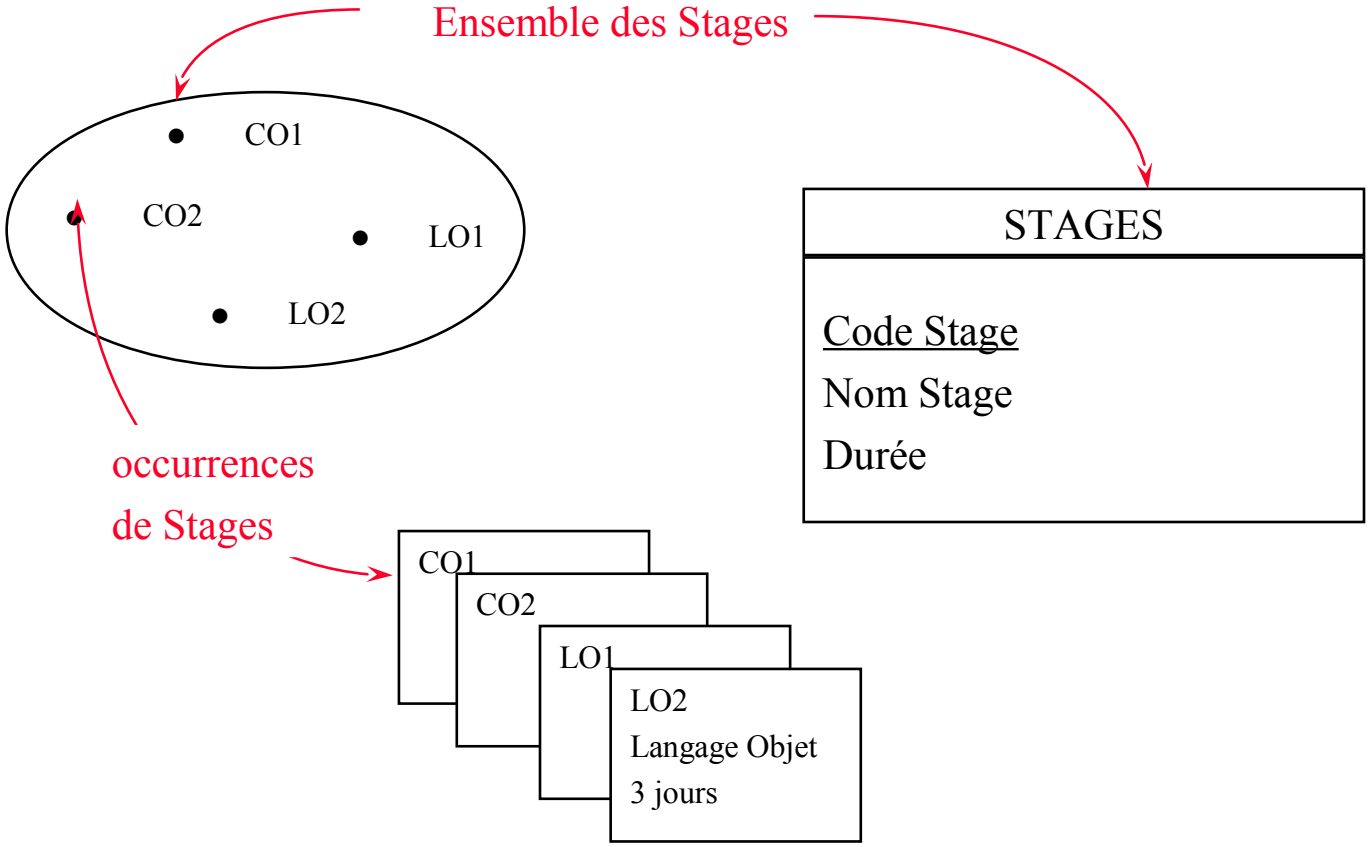
- Une entité correspond
  - à un nom ou à un complément dans les phrases de l'utilisateur

Entités

- Dans les entités les informations apparaissent sous forme atomique
- Dans le MCD
  - On ne stocke pas d'information calculées
    - On stocke les informations qui servent au calcul
    - Le calcul est un traitement
  - Une information n'apparait qu'à un seul endroit
    - Pas de redondance
  - On ne stocke pas d'information inutile
    - Toute information est utilisée dans un traitement

Entités et occurrences d'entités

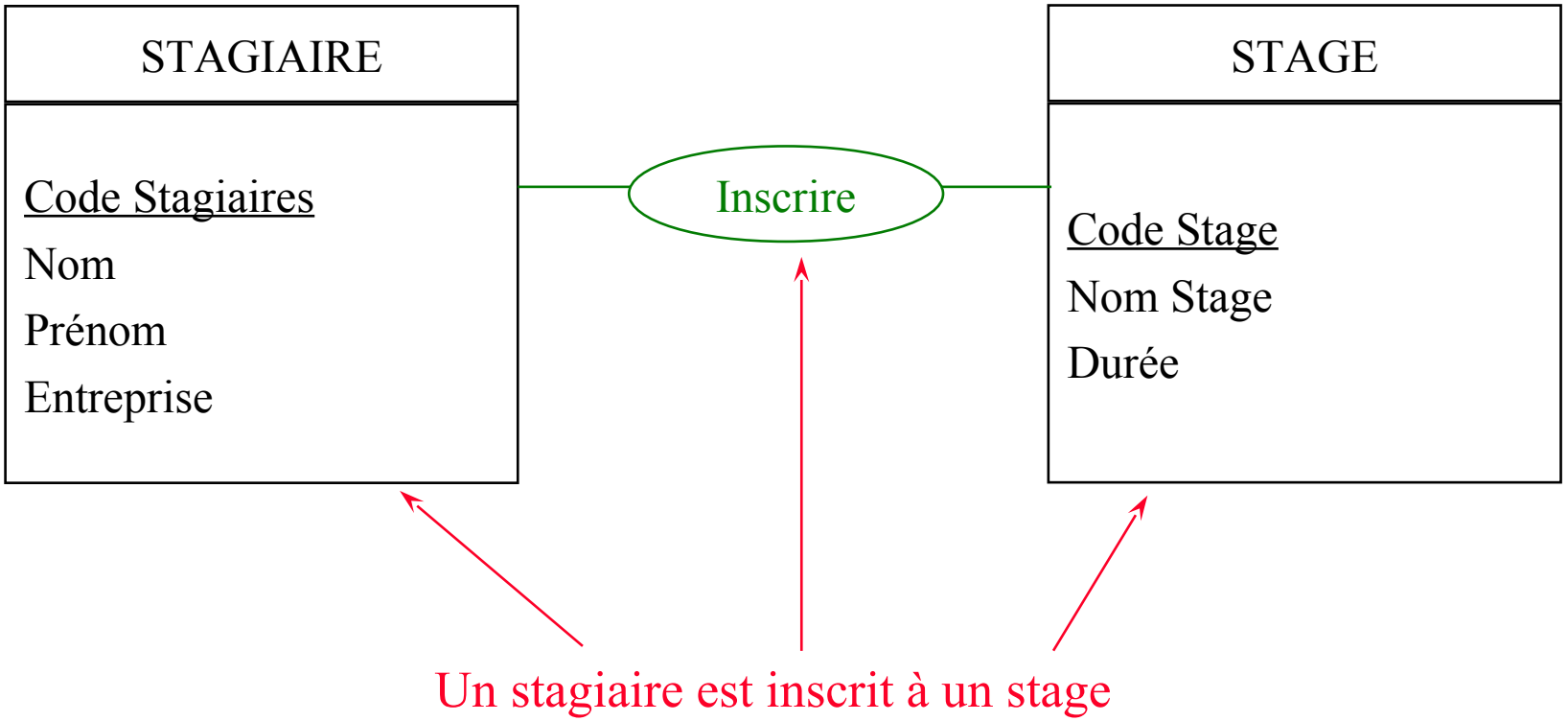
- Une occurrence d'entité correspond à un exemple de l'entité



Entités

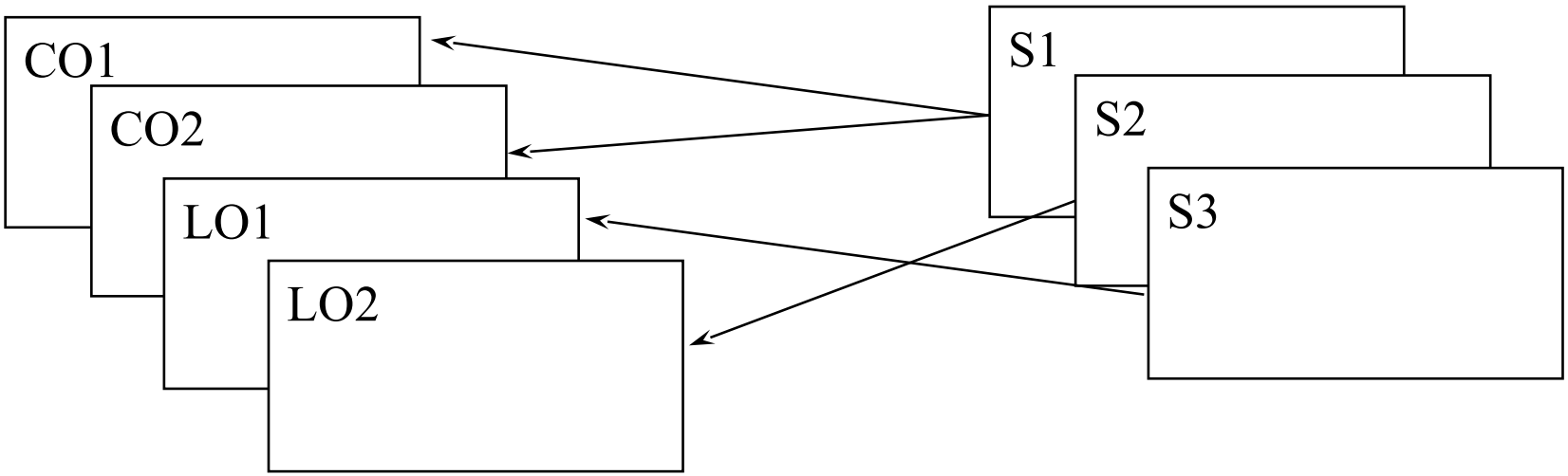
- Pour trouver les entités on a le choix entre 2 façons de faire :
  - 1 - A partir du dictionnaire de données : regrouper les informations par « famille » ➔ les ENTITEs
  - 2 - Écouter l'utilisateur lors des interviews et traduire les noms et les compléments des phrases en ENTITE
    - Les verbes seront transformées en relation
  - *Nous allons apprendre Merise en appliquant la 2eme méthode*
    - *Plus facile*

Relation "Liens de sens entre entités"



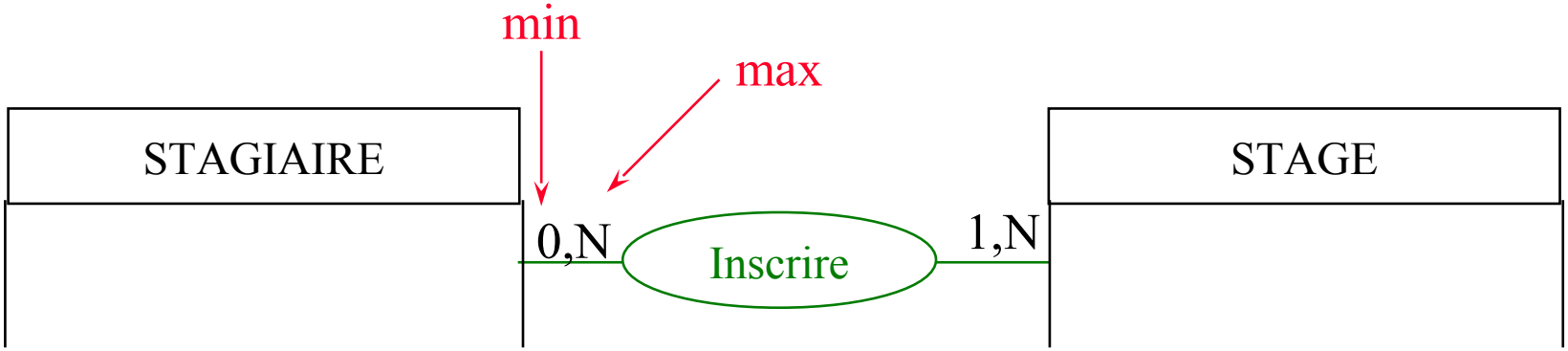
Occurrences de relations

- Une occurrence de relation correspond à la participation d'une entité à la relation
  - par exemple « S3 s'est inscrit à LO1 »



Par construction, une seule association est possible entre une occurrence de stage et une occurrence de stagiaire.

Cardinalités



La cardinalité exprime le nombre minimal et maximal de participations d'une occurrence d'entité à la relation.

- A une occurrence de Stagiaire peut correspondre de 0 à N (plusieurs) occurrence de relations "Inscrire".
- A une occurrence de Stage peut correspondre de 1 à N (plusieurs) occurrences de relations "Inscrire".

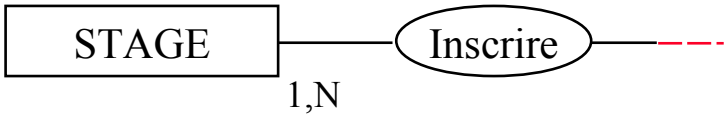
Cardinalités

- Les cardinalités Merise prennent la valeur
  - 0 , 1
  - 1 , 1 minimum 1 et maximum 1 c'est obligatoirement 1
  - 0 , N
  - 1 , N

- Il n'y a pas d'autres valeurs possibles

CARDINALITES

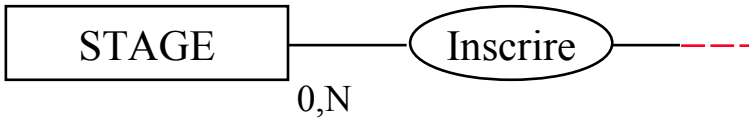
"Elles expriment les règles de gestion"



A une occurrence de STAGE correspond de 1 à N occurrence de STAGIAIRE

- L'occurrence stage ne peut être créée que s'il y a au moins un inscrit.

⇨ L'inscription du premier stagiaire entraîne la création de l'occurrence de Stage

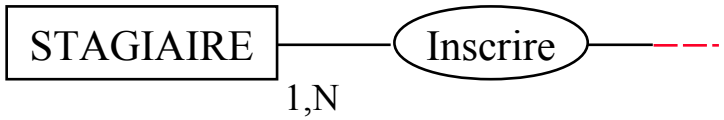


A une occurrence de STAGE correspond de 0 à N occurrence de STAGIAIRE

- L'occurrence stage peut être créée avant toute inscription.

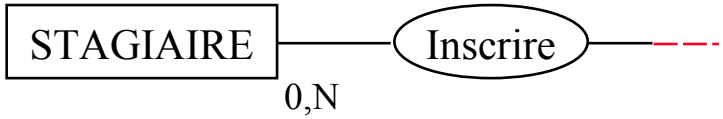
Cardinalités

"Elles expriment les règles de gestion"



A une occurrence de STAGIAIRE correspond de 1 à N occurrence de STAGE.

- L'occurrence stagiaire ne peut être créé indépendamment de son inscription.

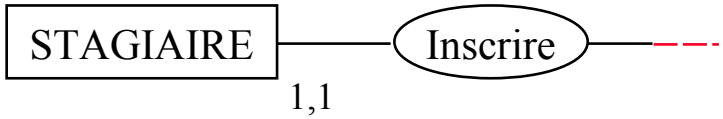


A une occurrence de STAGIAIRE correspond de 0 à N occurrence de STAGE.

- L'occurrence stagiaire peut être créée indépendamment de son inscription

Cardinalités

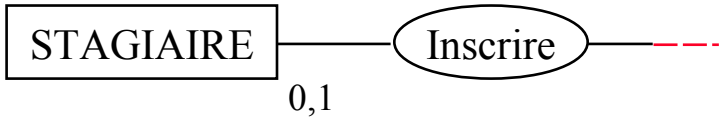
"Elles expriment les règles de gestion"



A une occurrence de STAGIAIRE correspond 1 et 1 seule occurrence de STAGE.

- On ne peut créer l'occurrence Stagiaire indépendamment de son inscription.

⇨Le stagiaire ne peut s'inscrire qu'une seule fois.



A une occurrence de STAGIAIRE correspond de 0 à 1 occurrence de STAGE

- On peut créer l'occurrence Stagiaire indépendamment de son inscription.

⇨Il ne peut s'inscrire qu'une seule fois.

Exercice

CAS 1 : HELITOUR

- OBJECTIF
- Placer les cardinalités en fonction des règles de gestion

Cardinalités

"Les relations ternaires"

Une relation ternaire est une relation avec

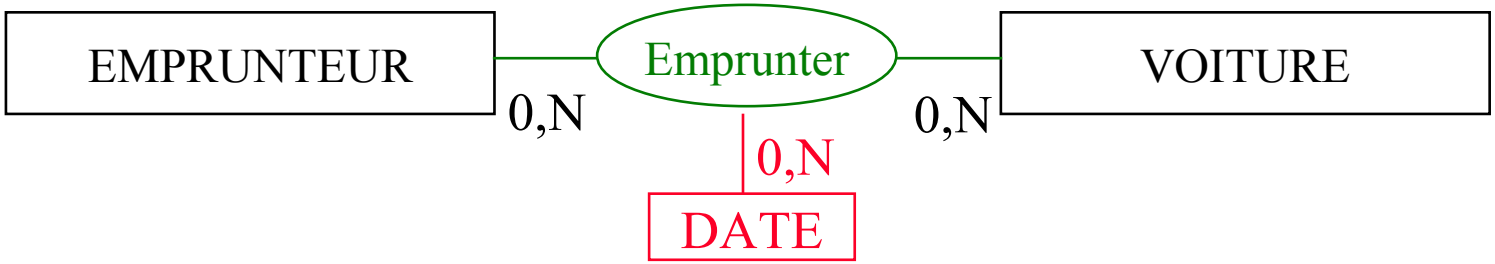
0,N ou 1,N de chaque coté de la relation

Par construction, la même occurrence d'Emprunteur ne peut emprunter plusieurs fois la même occurrence de Voiture.

↳ Le couple Id.Emprunteur - Id.Voiture est unique

Cardinalités

"Les relations ternaires"



La même occurrence d'Emprunteur ne peut emprunter à plusieurs reprises la même occurrence de Voiture.

La date permet de distinguer chaque occurrence de relation.

- Le triplet Date - Id.Emprunteur - Id.Voiture est unique

Une relation qui a plus de 2 pattes est obligatoirement une relation ternaire

Cardinalités

"Les relations ternaires"

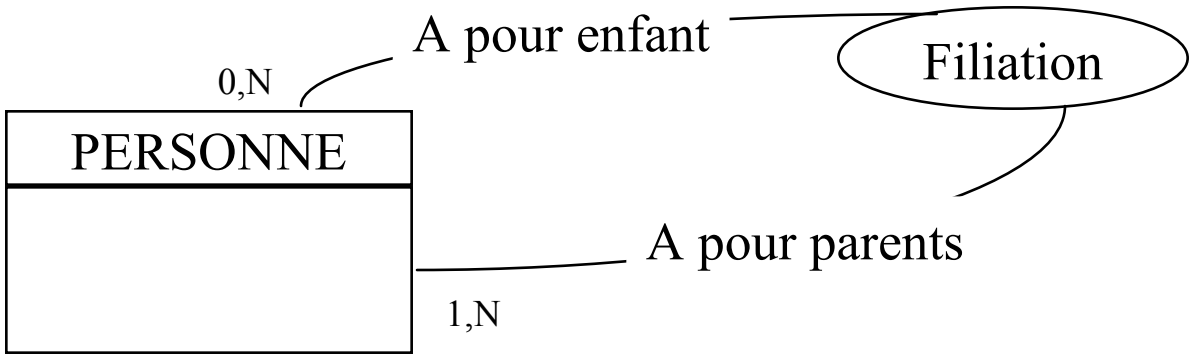
Une relation ternaire est une relation avec

0,N ou 1,N de chaque coté de la relation

Une relation qui a plus de 2 pattes est obligatoirement une relation ternaire

- Pour valider une relation ternaire on vérifie que :
  - la concaténation des identifiants des entités qui participent a la relation est unique

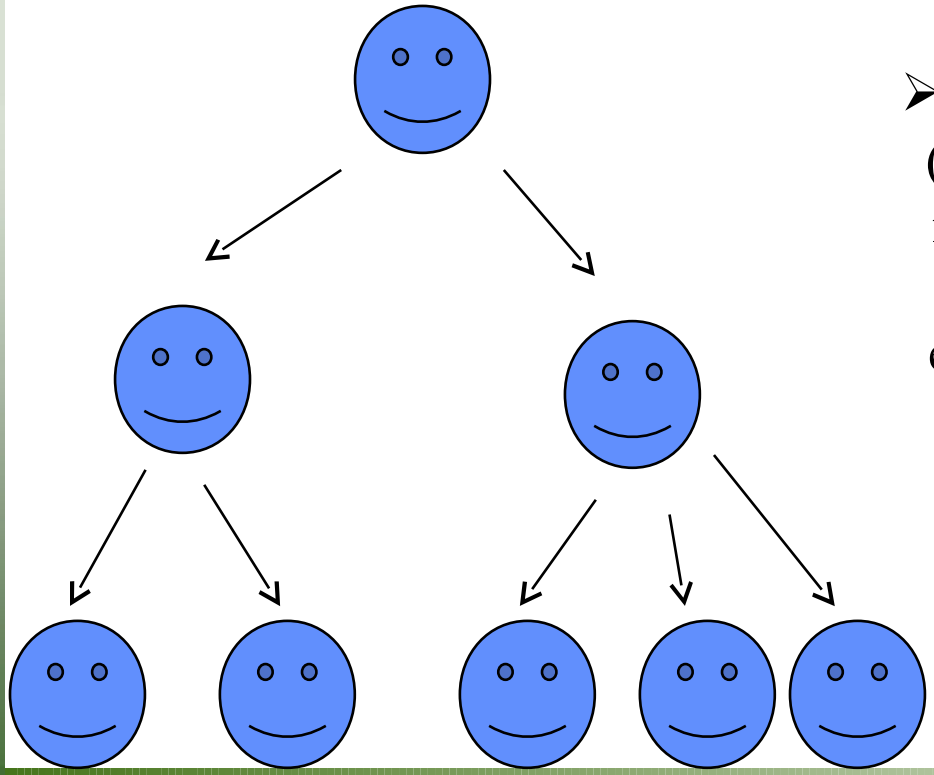
Relation reflexive



Une relation réflexive représente une hiérarchie !

Relation reflexive

C’est une représentation hiérarchique des données



- Comme nous sommes en SGBDR (système gestion base de données relationnelles) le relationnel gère très mal le hiérarchique ,ce qui entraine un temps de réponse trés long

Valider un MCD

Les forme normales permettent de valider le MCD

⇨ 3 formes normales ...

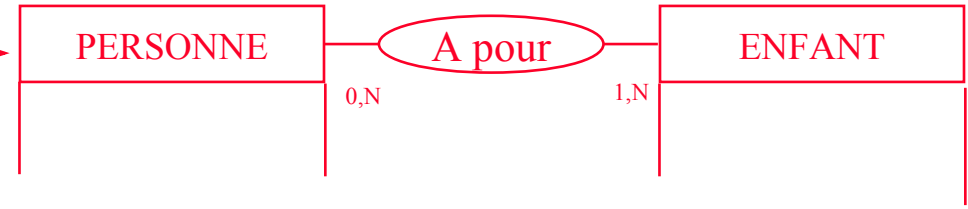
Règles de construction
"1ère FN"

Les propriétés d'une entité ou d'une relation doivent être sous forme atomique.

⇨ Pas de listes, tableaux ...

PERSONNE
N° Immat.
Nom patronymique
Prénom usuel
Date de Naissance
Prénoms Enfants

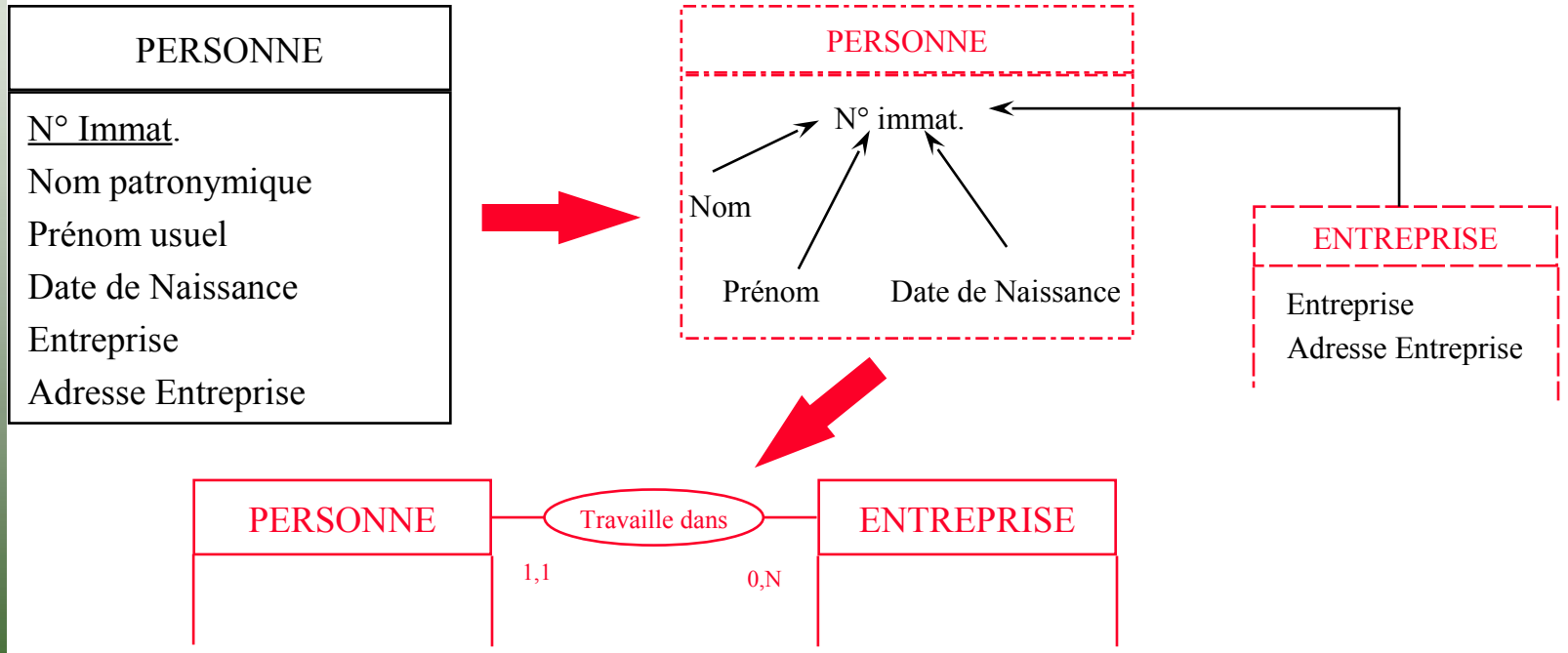
Liste



Règles de construction
"3ème FN"

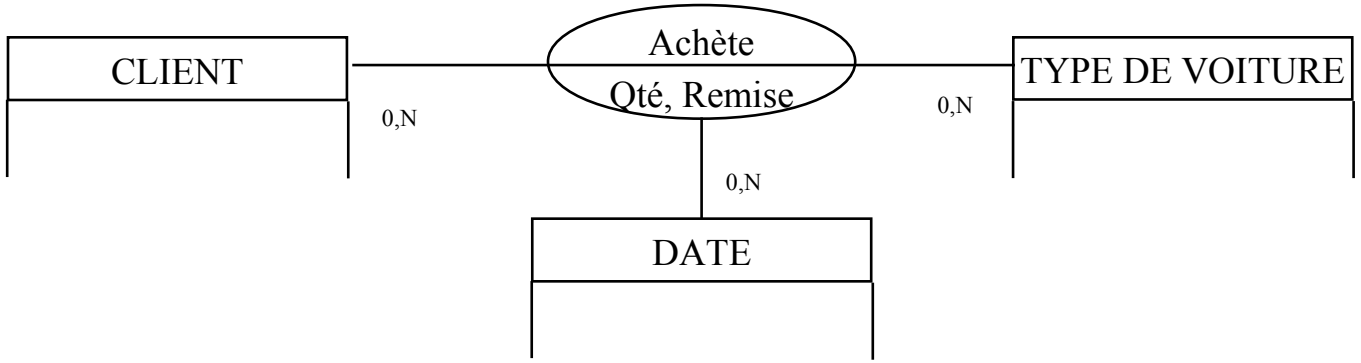
Les propriétés d'une entité ou d'une relation doivent être en dépendance directe avec l'indentifiant.

⇨ Pas de transitivité.



Règles de construction
"2ème FN"

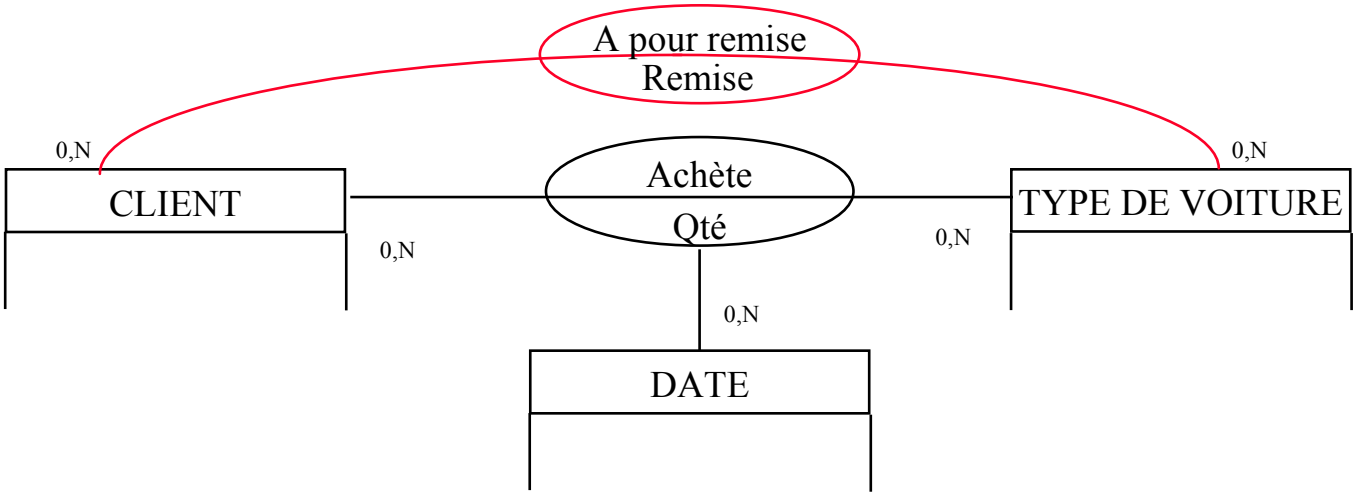
Les propriétés d'une relation doivent être en dépendance complète avec les identifiants des entités reliées.



Règles de construction
"2ème FN"

Si la remise pour un client et un type de voiture est toujours la même,

⇨ La remise ne dépend pas de la date



Démarche de construction

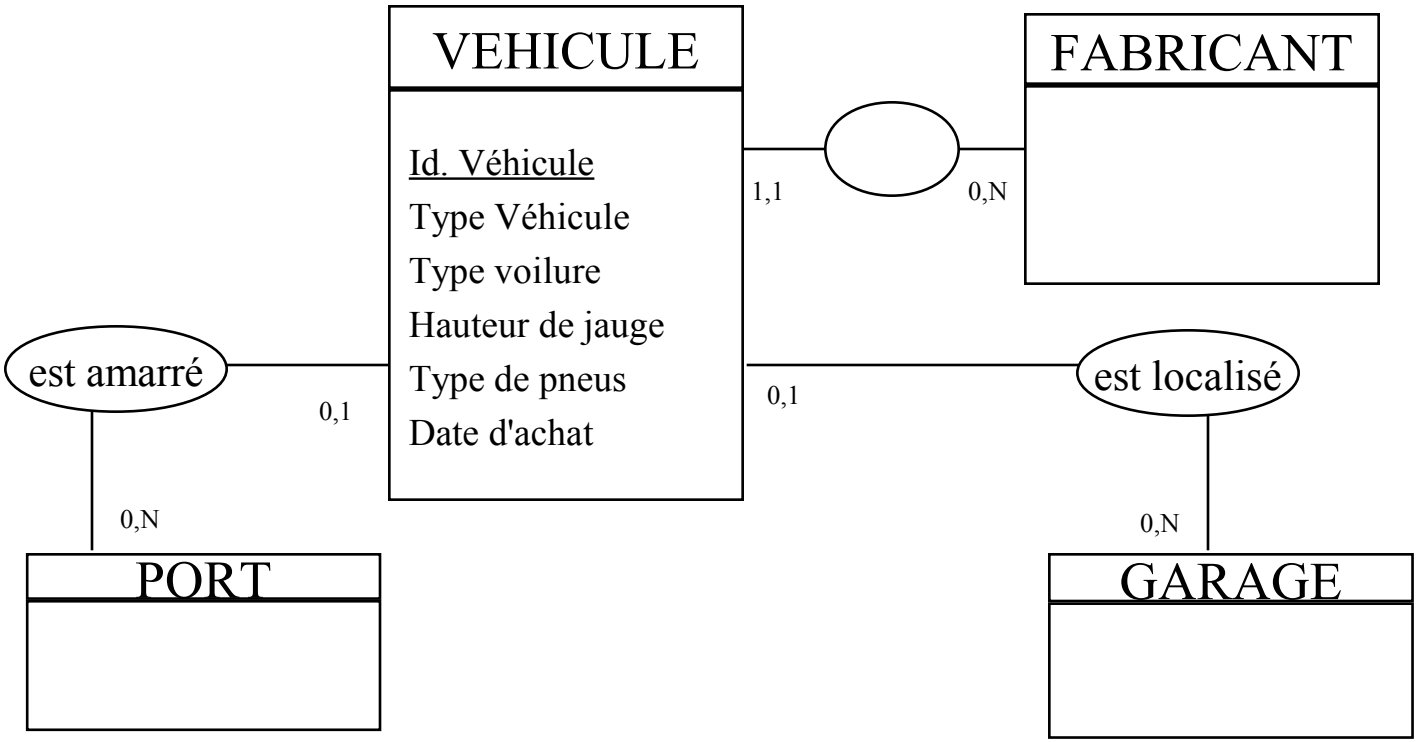
- **Etablir le dictionnaire de données.**
- **Repérer les entités.**
- **Attribuer à chaque entité un identifiant (s'il n'existe pas le créer).**
- **Placer les propriétés dans les entités.**
- **Placer les relations (éventuellement les propriétés des relations).**

Exercice

CAS 1 : LYCEE

- OBJECTIF
- Réaliser le MCD

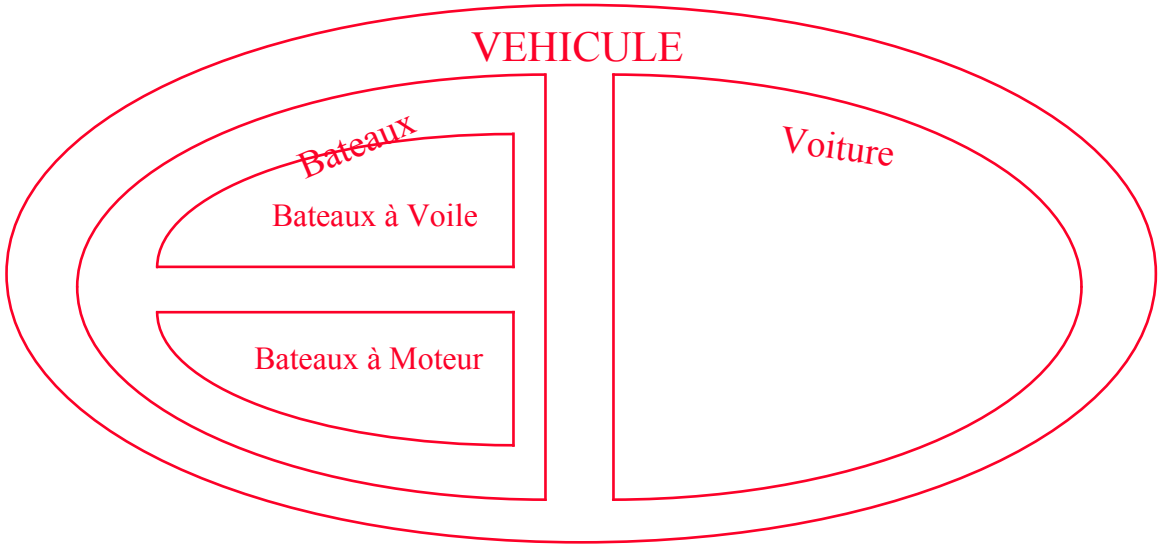
Généralisation-Spécialisation



Une occurrence de Véhicule de type "Bateau" est toujours reliée à une occurrence de type "Port".

Une occurrence de Véhicule de type "Voiture" est toujours reliée à une occurrence de type "Garage".

Généralisation-Spécialisation

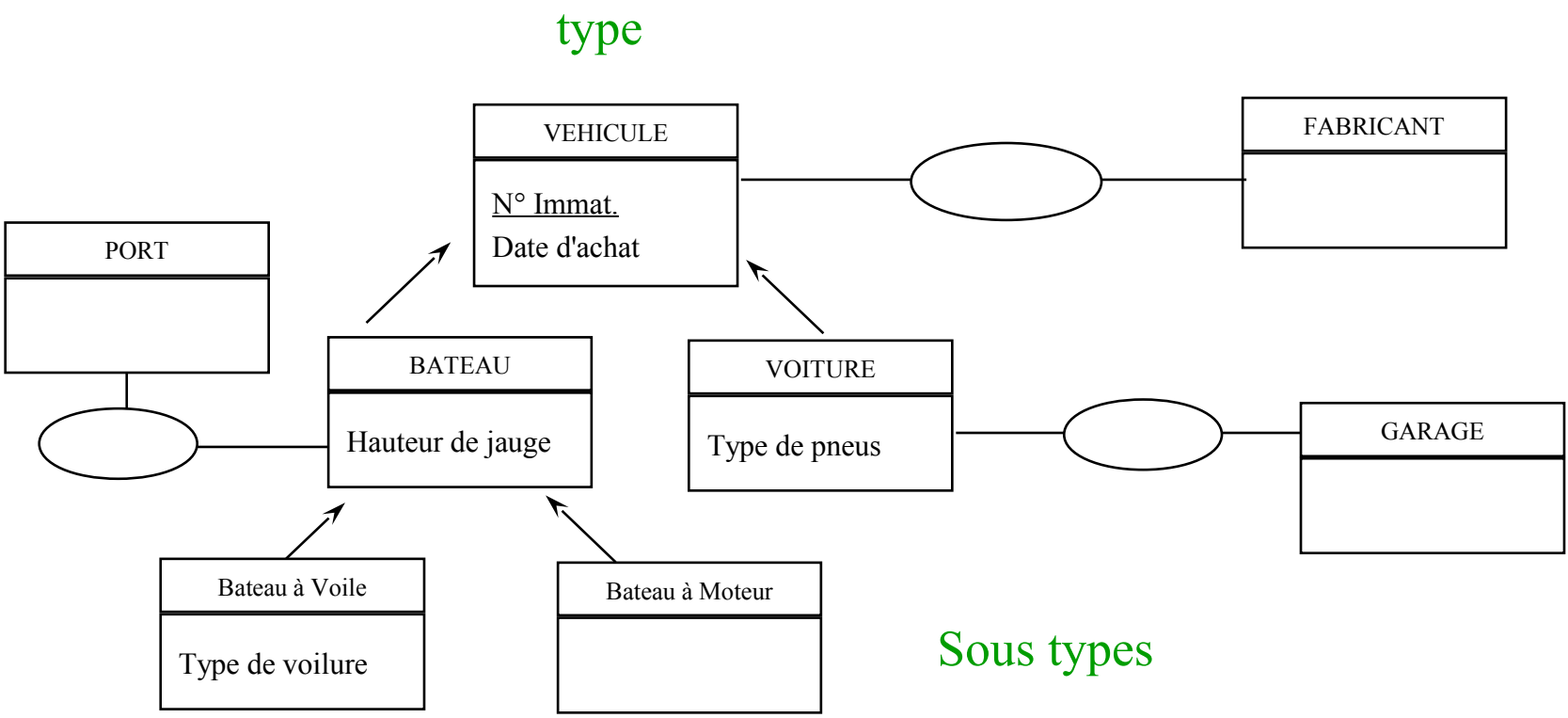


On a un ensemble de véhicules décomposé en sous ensembles

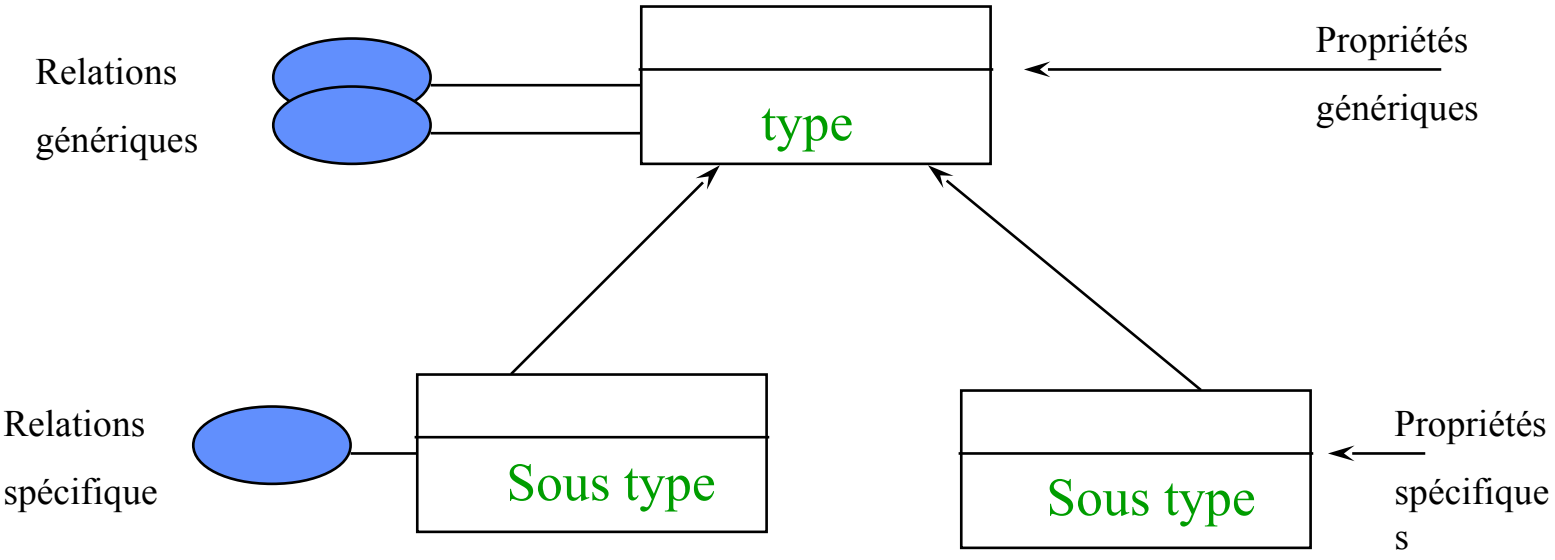
bateaux à moteur, bateaux à voiles, voitures

➔ en Merise on parle de types et de sous types

Généralisation-Spécialisation



Généralisation/Spécialisation



Contrôle du modèle

- **Vérifier que chaque propriété se trouve à un seul endroit du modèle.**
- **Contrôler chaque entité en vérifiant :**
  - Que chaque entité possède un identifiant.
  - Que chaque propriété est significative.
  - La 1ère FN.
  - La 3ème FN.

Contrôle du modèle

- **Contrôler chaque relation en vérifiant :**
  - Qu'une occurrence de relation ne lie qu'une et une seule occurrence de chacune des entités reliées.
  - Que les relations de cardinalités 1,1 ne porte pas de propriété.
  - La 1ère FN.
  - La 2ème FN.
  - La 3ème FN.
- **Contrôler que le modèle produit les résultats attendus.**

Exercice

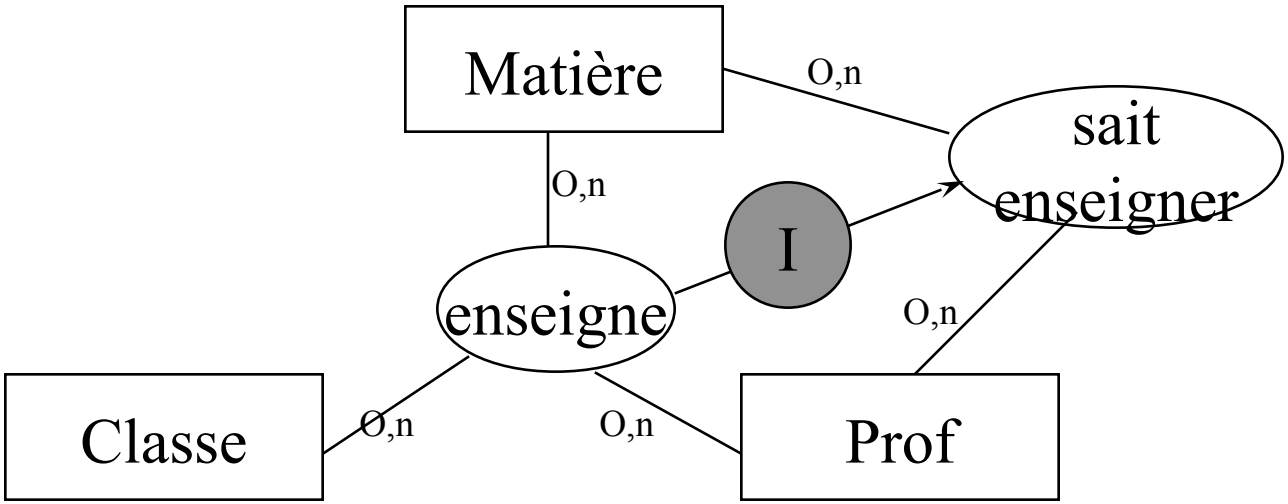
CAS 2 : Parc de véhicules

- OBJECTIF
- Réaliser le MCD

Contraintes sur les relations ou les pattes

- **Contrainte d'exclusion**
- **Contrainte de totalité**
- **Contrainte d'inclusion**
- **Contrainte de stabilité**
- **Contrainte d'unicité**

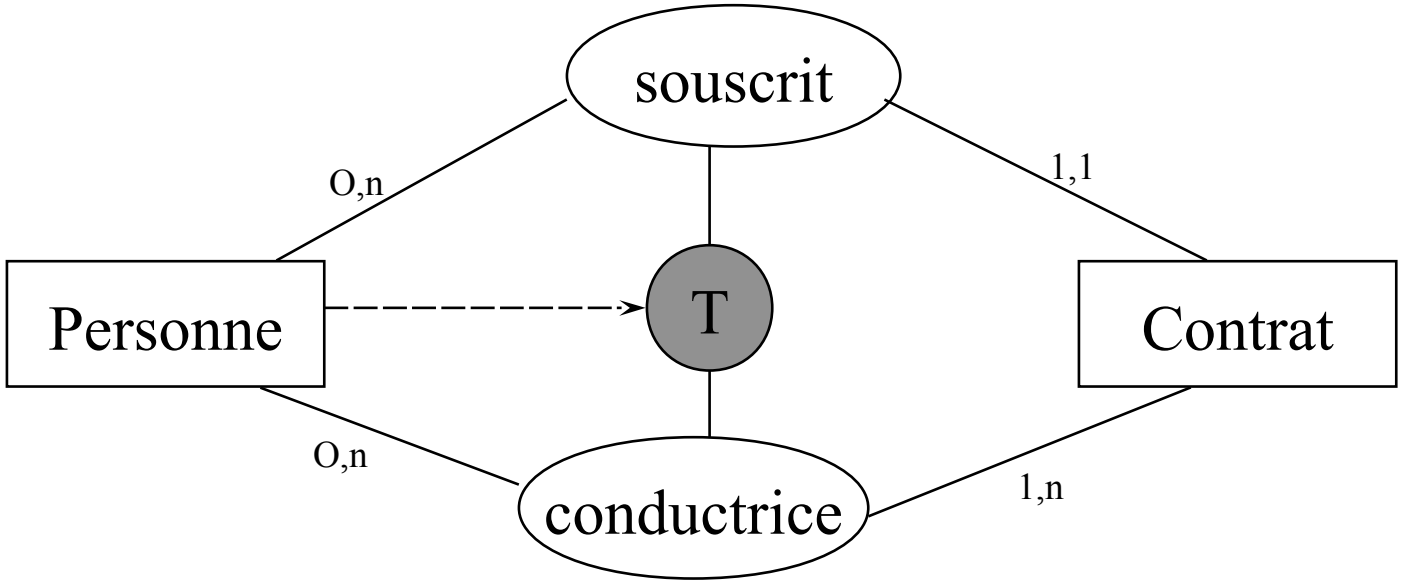
Contrainte d'inclusion



Si enseigne (x:matière, y : professeur)

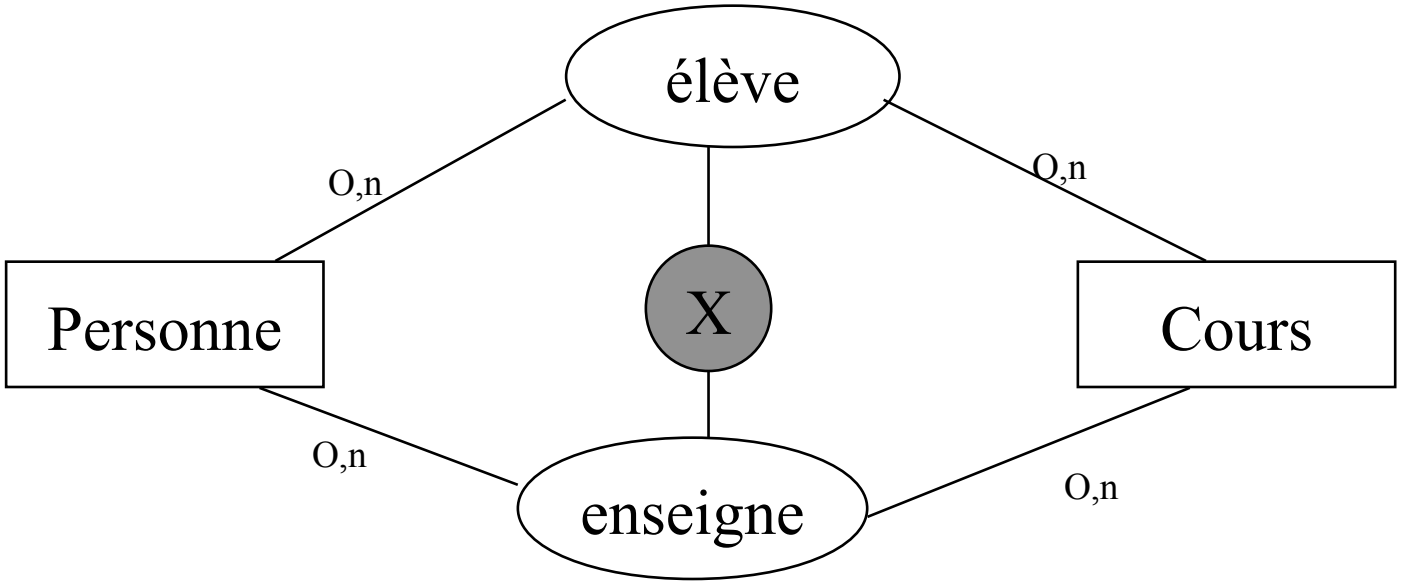
alors sait enseigner (x:matière, y : prof)

Contrainte de totalité



Toute personne doit être reliée à un contrat d'une façon ou d'une autre

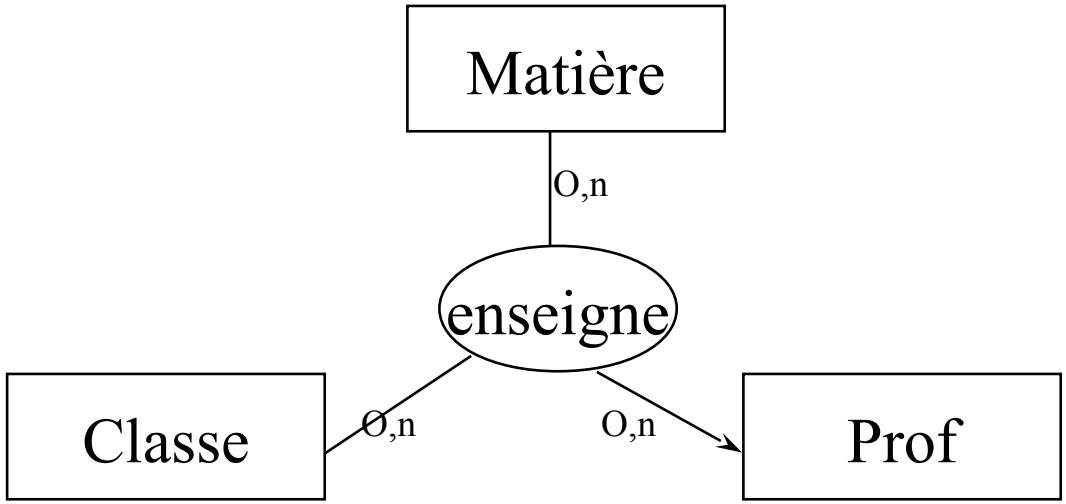
Contrainte d'exclusion



Si enseigne (x:personne, y :cours)

alors non élève (x:personne, y : cours)

Contrainte d'unicité



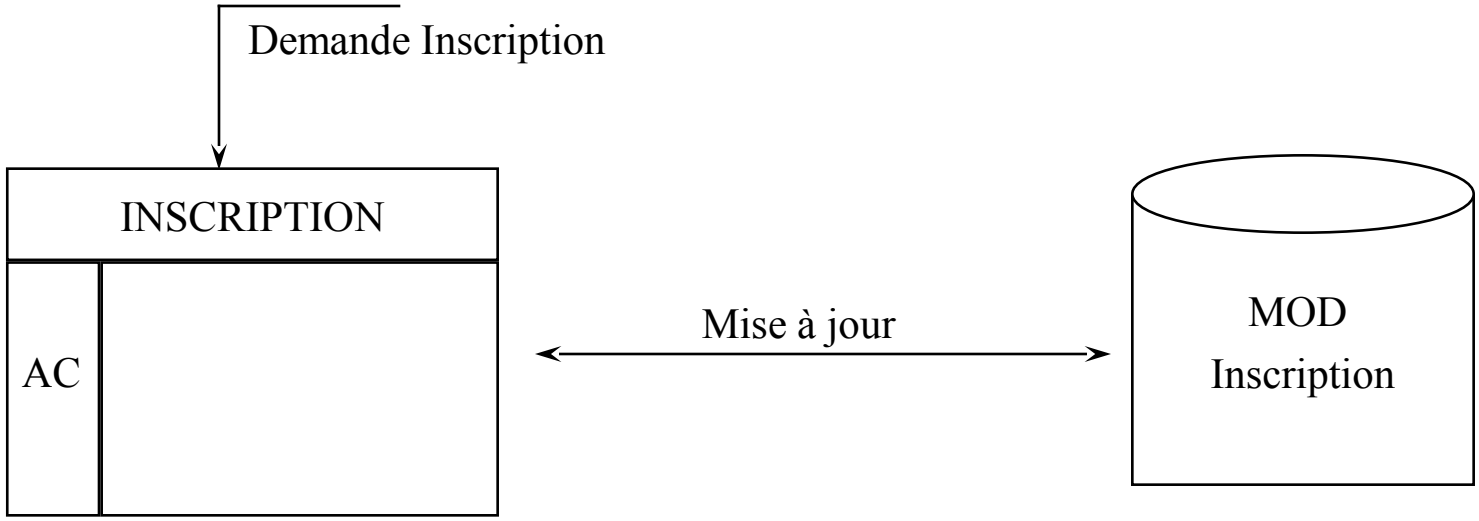
Dans une classe, une matière n'est enseignée que par un seul professeur

Contrainte de stabilité

Les relations sont définitives.

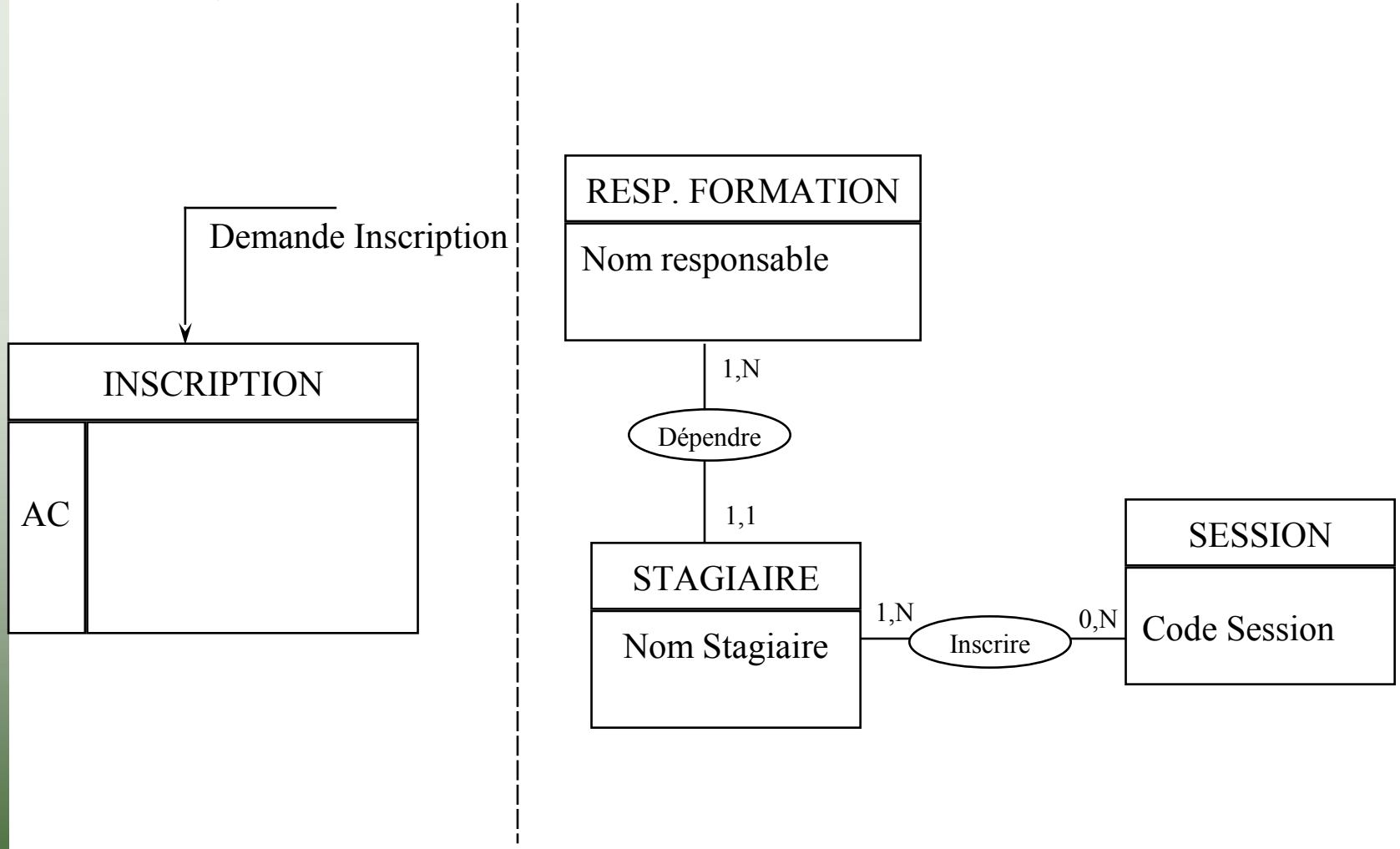
Une occurrence ne peut être détruite que si l'entité qu'elle met en jeu est elle même détruite.

Validation données/traitements



On valide les données du MOD avec le MOT

Exemple de MOD



Validation données/traitements

Vérifier que chaque information du MCD ou MOD est utilisée dans un traitement :

- un écran
- une impression
- une opération dans un traitement
- …

Vérifier que les traitements n'utilisent pas des informations oubliées …

On ne stocke pas d'information inutiles

Optimisation fonctionnelle

- Veiller à ce que les opérations ne soient pas coûteuses en temps.
  - Réduire le nombre de tables,
  - Limiter le nombre de jointures,
  - Introduire des redondances, des compteurs, des états.

- L'optimisation fonctionnelle en contre partie rend :
  - Les mises à jour plus complexes,
  - Les évolutions plus difficiles.

Exercice

CAS 8 : Accident

- OBJECTIF
- Réaliser le MCD

Exercice noté

- **Au bon vouloir du formateur**
  - Bonne chance ☺

MLD

Définition

- Un Modèle Logique des Données (MLD) est une représentation des données d'un système devant être mémorisées sur des supports informatiques permanents (fichier, base de données) et des liens existants entre ces données.
- Il traduit le MOD dans un formalisme compatible avec l'état de l'art, mais encore portable par rapport à des choix techniques précis liés à des famille de SGBD

Modèle logique de données (MLD)

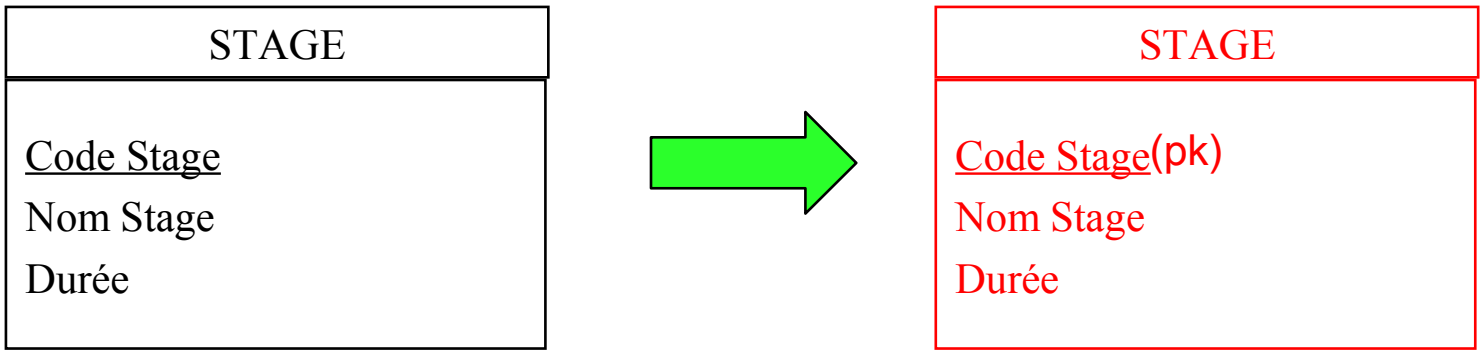
Règles de passage du MCD au MLD
(x vaut 0 ou 1)

- Les entités sont transformées en tables.
  - Les identifiants des entités deviennent clé primaires
- Les relations x,N / x,N sont transformées en tables.
  - La primary key est constituée de la concaténation des identifiants des entités qui concourent à la relation
- Les relations x,1 / x,N deviennent clé étrangères.
  - L'identifiant coté x,N est migré dans l'entité coté x,1.
- Les relations x,1 / x,1 sont transformées en tables ou deviennent clé étrangères.
  - L'identifiant en est déduit en fonction de la solution choisie.

Modèle logique de données (MLD)

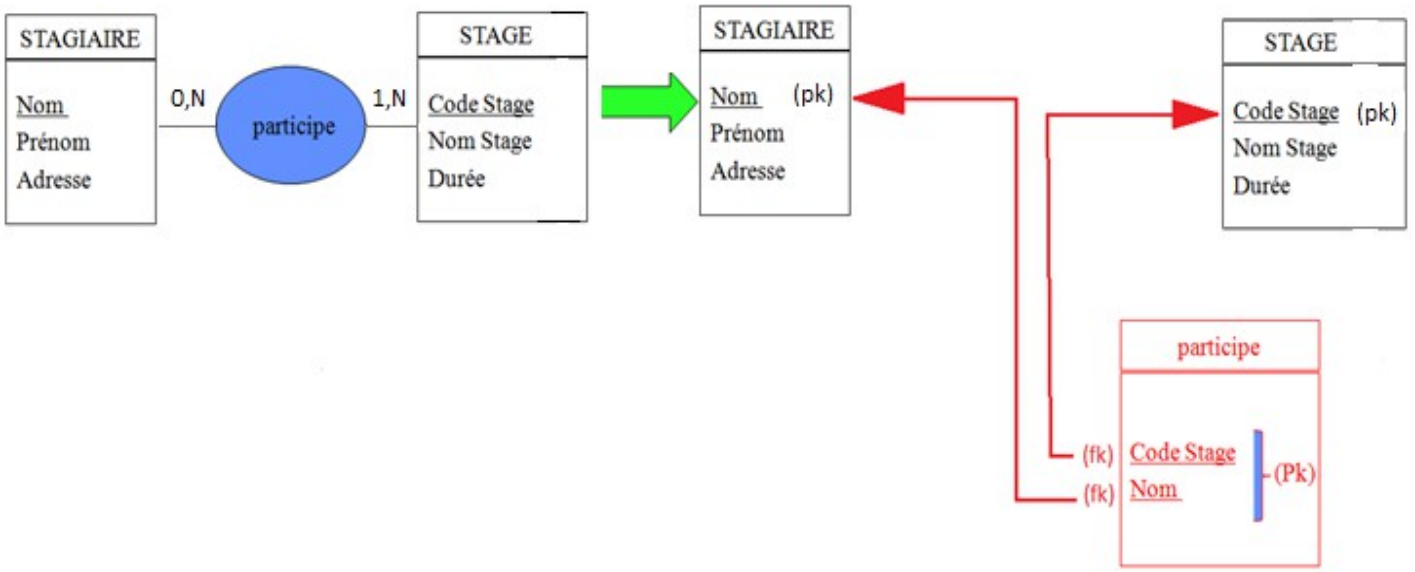
Règles de passage du MCD au MLD

- Les entités sont transformées en tables.
  - Les identifiants des entités deviennent clé primaires



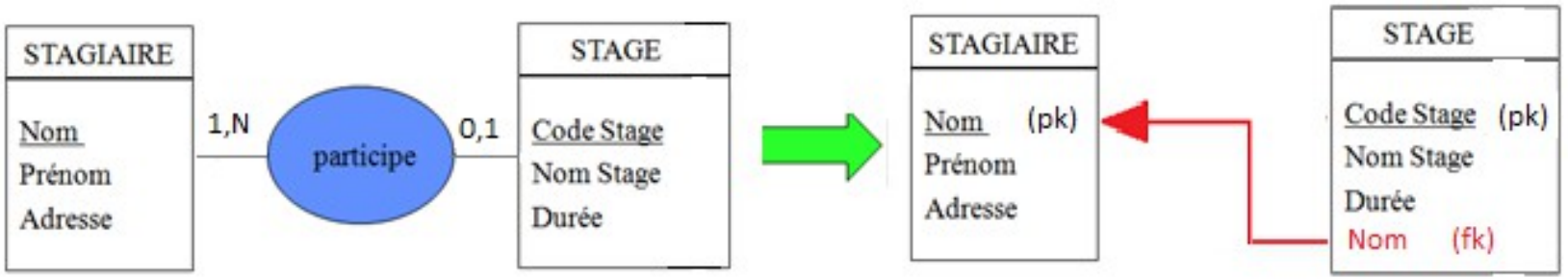
Modèle logique de données (MLD)

- **Règles de passage du MCD au MLD (x vaut 0 ou 1)**
  - Les relations x,N / x,N sont transformées en tables.
    - La primary key est constituée de la concaténation des identifiants des entités qui concourent à la relation



Modèle logique de données (MLD)

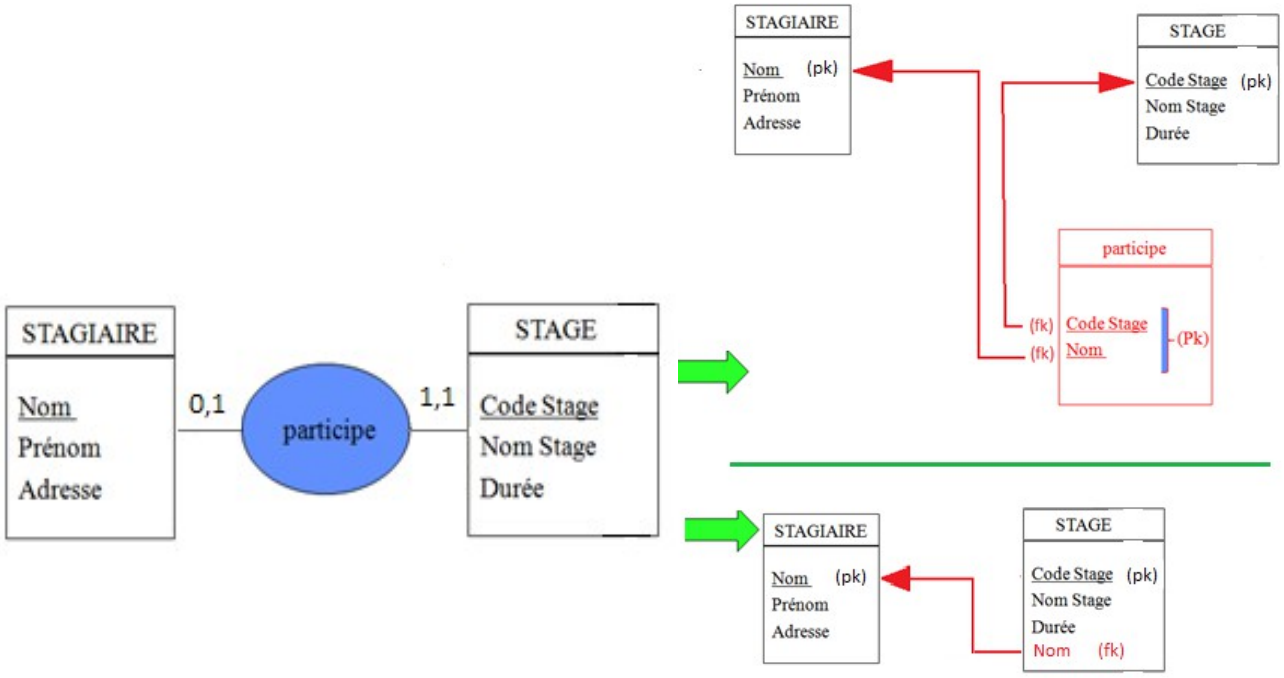
- **Règles de passage du MCD au MLD (x vaut 0 ou 1)**
  - Les relations x,1 / x,N deviennent clé étrangères.
    - L'identifiant coté x,N est migré dans l'entité coté x,1.



Modèle logique de données (MLD)

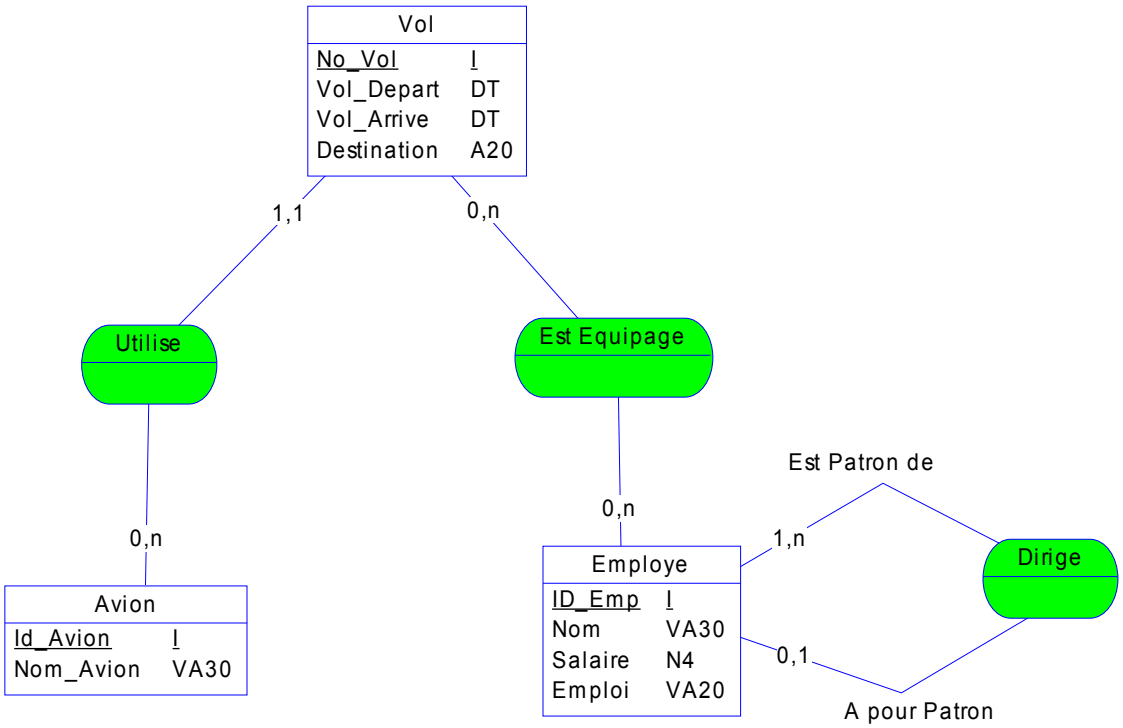
Règles de passage du MCD au MLD (x vaut 0 ou 1)

- Les relations x,1 / x,1 sont transformées en tables ou deviennent clé étrangères.
  - L'identifiant en est déduit en fonction de la solution choisie.

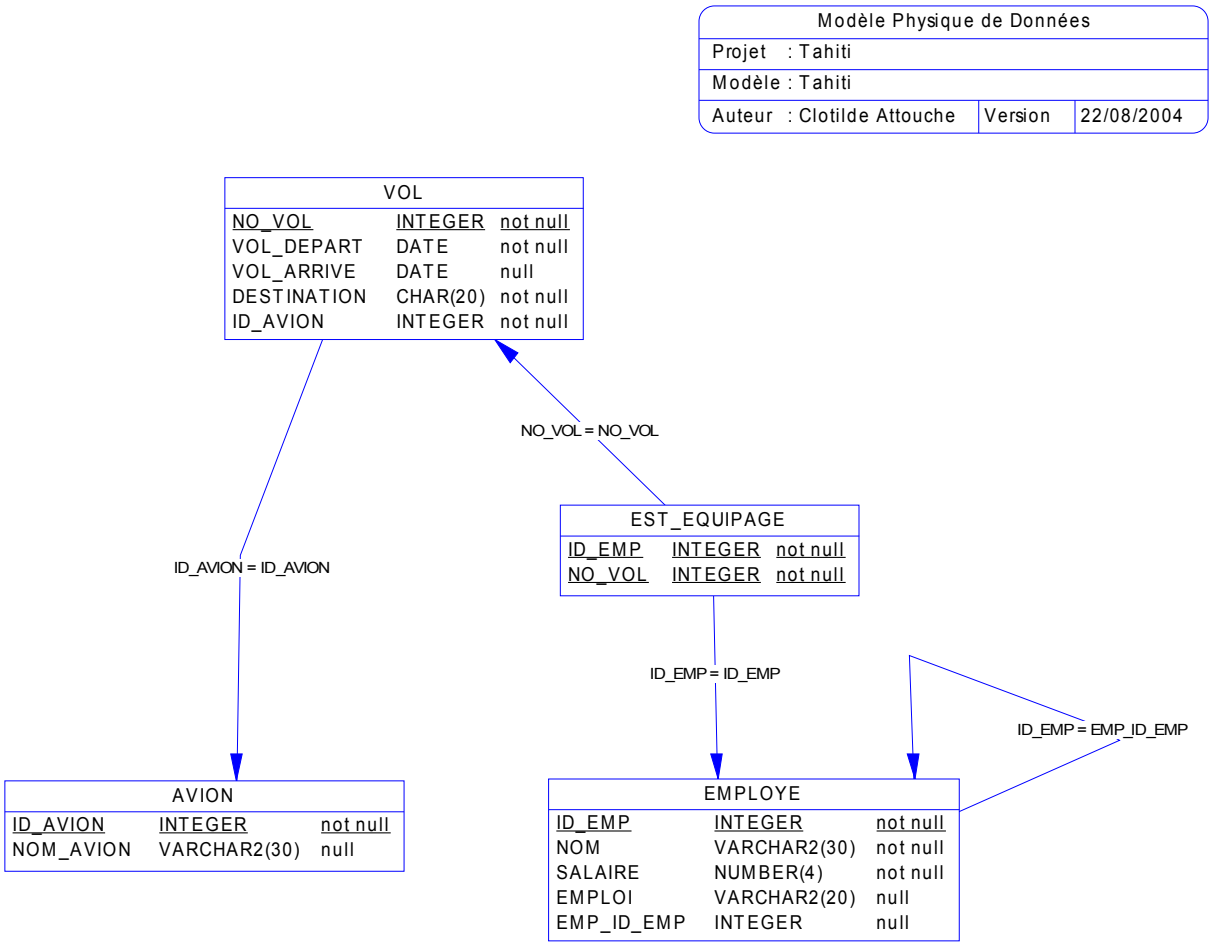


Exemple de MCD

Modèle Conceptuel de Données		
Projet : Tahiti		
Modèle : Tahiti		
Auteur : Clotilde Attouche	Version	22/08/2004



Génération du MLD



Exercices

- **Construire les MLDs**

Modèle Logique de données Réparti

Définition

- Un Modèle Logique des Données réparti précise l'implantation logique des données permanentes sur chacune des machines logiques type d'un système (Schéma d'Architecture Logique des Moyens Informatiques)
- Le MLD Réparti est composé de 2 à N **Modèles logiques de données** locaux
- Chaque MLD local est propre à une machine logique type.

Algèbre Relationnel

Projection

- **La projection permet de ne conserver que certains attributs(colonnes) d'une table en éliminant les doublons.**
- **Expression :**
  - PROJECT (Nom de tables, attribut1, attribut2, etc)

Restriction

- **La restriction d'un fichier permet de ne conserver que les lignes qui satisfont à une condition.**
- **Expression :**
  - RESTRICT (Nom de table, condition)
  - RESTRICT (VENTES, Nom = « Martin »)

Join

- **L'opération de jointure fonctionne sur deux tables possédant au moins un attribut commun.**
  - Il consiste à créer une troisième table dont la structure est composé de l'ensemble des attributs des deux tables et dont les enregistrements sont ceux qui satisfont à la condition.
- **Expression :**
  - JOIN (R1, R2, Condition)
  - JOIN (R1,R2, R1.Code client = R2.Code client)

Difference

- **La différence fonctionne entre deux tables de même structure.**
- **Elle consiste à créer une troisième table de même structure dont les lignes sont celles qui appartiennent à la premiere table diminué de celles appartenant à la seconde.**
- **Expression :**
  - DIFFERENCE (R1, R2)

Union

- **L'opération d'union fonctionne sur deux tables de même structure.**
- **Elle consiste à créer une troisième table composée de l'ensemble des lignes des deux tables de départ à l'exclusion des doublons (enregistrements identiques)**
- **Expression :**
  - UNION (R1, R2)

Intersection

- **L'opération d'intersection fonctionne sur deux tables de même structure.**
- **Elle consiste à créer une troisième table composé de l'ensemble des lignes appartenant simultanément aux deux tables de départ**
- **Expression *:***
  - *INTERSECTION (R1, R2)*

Exercice

- **A partir du MLD Lycée répondre aux questions :**
  - Nom et adresse des élèves du lycée
  - Nom et prénom des professeurs principaux
  - Nom et prénom des élèves délégués
  - Nom, prénom et nom des classes dans lesquelles les professeurs enseignent une matière
  - Liste des professeurs qui enseignent le français dans la classe « ROUGE »
  - Liste des élèves qui ont eu 16 en Mathématique
  - Liste des professeurs diplômés en Géographie
  - Liste des élèves qui n'ont pas de note

Nom et adresse des élèves du lycée

- **On regarde les tables qui contiennent les informations à afficher et utiles pour faire les tests**
- **Ici la table eleve**
- **R1=Project (eleve, nom_elv, adresse_elv)**

Nom et prénom des professeurs principaux

- On regarde les tables qui contiennent les informations à afficher et utiles pour faire les tests
- Ici les tables classe et professeur
- « Est professeur principal » est représenté par la clé étrangère « id_prof » dans la table classe du MLD (voir MCD eleve)

- R1=join (classe, professeur, classe.id_prof=professeur.id_prof)
- R2=project (R1, nom_prof, prenom_prof)

Nom et prénom des élèves délégués

- **R1=restrict (eleve, est_delegue='O')**
- **R2=project(R1, nom_elv, prenom1_elv)**

Nom, prénom des professeurs et nom des classes dans lesquelles les professeurs enseignent une matière

- **R1=join(enseigne, professeur, enseigne.id_prof=professeur.id_prof)**
- **R2=join(R1,classe,R1.id_classe=classe.id_classe)**
- **R3=project(R2,nom_prof, prenom_prof,nom_classe)**

Liste des professeurs qui enseignent le français dans la classe « ROUGE »

- **R1=join(enseigne, professeur, enseigne.id_prof=professeur.id_prof)**
- **R2=join(R1,classe,R1.id_classe=classe.id_classe)**
- **R3=restrict(R2,nom_classe=« Rouge »**
- **R4=join(R3,matiere,R3.id_matiere=matiere.id_matiere)**
- **R5=restrict(R4,nom_mat=« Français »)**
- **R6=project(R5, nom_prof)**

Liste des élèves qui ont eu 16 en Mathématique

- **R1=join(note,eleve,note.id_elev=eleve.id_eleve)**
- **R2=join(R1,matiere,R1.id_matiere=matiere.id_materiere)**
- **R3=restrict(R2,nom_mat=« Mathématique »)**
- **R4=restrict(R3,note=16)**
- **R5=project(R4,nom_elv)**

Liste des professeurs diplômés en Géographie

- **R1=join(professeur,est_diplome,est_diplome.id_prof=professeur.id_prof)**
- **R2=join(R1,matiere,R1.id_matiere=matiere.id_matiere)**
- **R3=restrict(R2,nom_mat=« Geographie »)**
- **R4=project(R3,nom_prof)**

Liste des élèves qui n'ont pas de note

- **La liste des élèves qui n'ont pas de note c'est:**
- **La liste de tous les élèves – la liste des élèves qui ont une note**
  - Attention la différence se fait avec 2 tables de même structure
- **R1=join(note,eleve,note.id_elev=eleve.id_eleve)**
- **R2=project(R1,nom_elv)**
- **R3=project(eleve,nom_elv)**
- **R4=différence(R3,R2)**