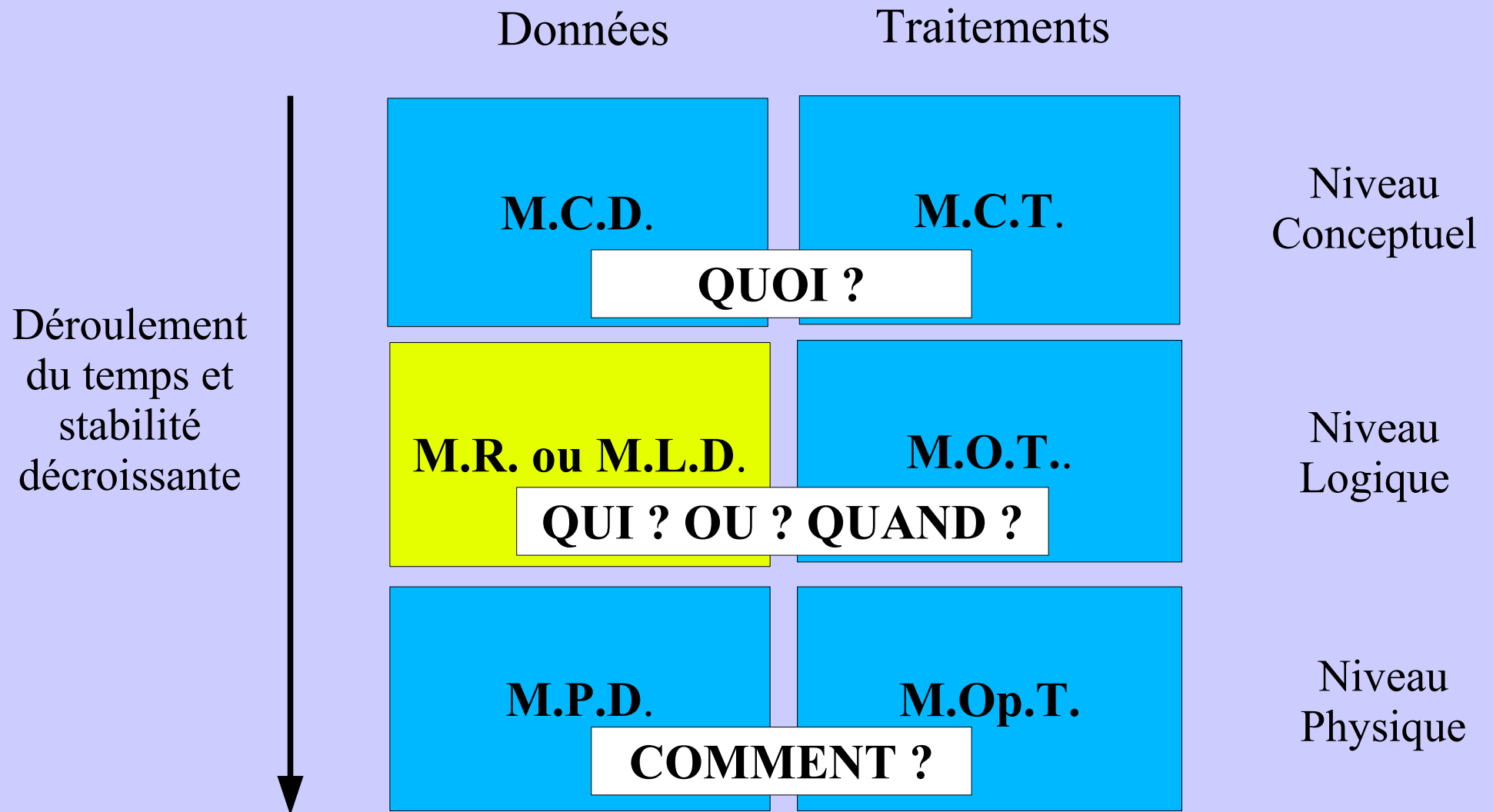


# Une approche par niveau



# Le Modèle Relationnel

- Il se situe au niveau logique
- C'est une représentation des données, issue de la modélisation conceptuelle puis organisationnelle des données
- Il est exprimé sous forme de RELATIONS

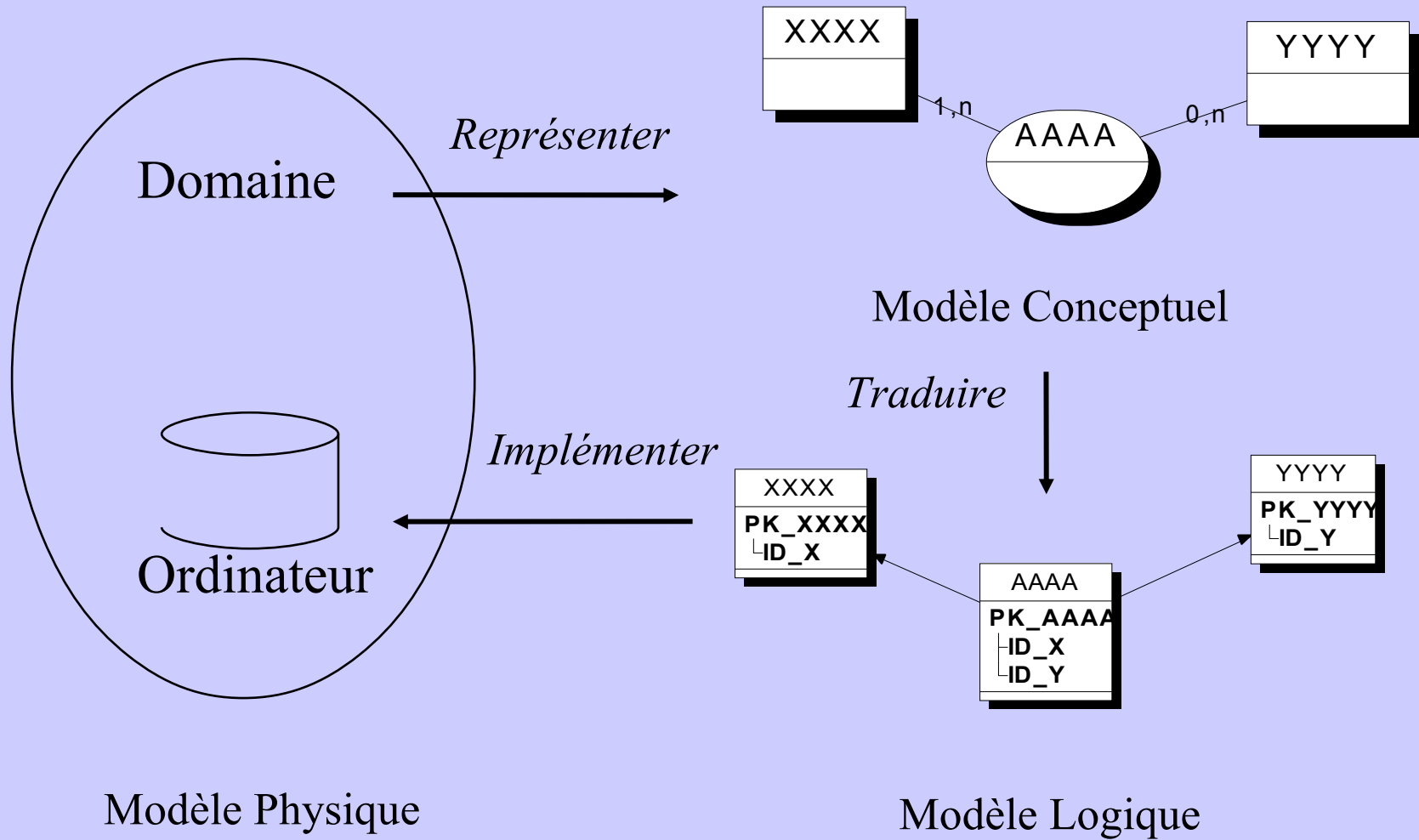
# Le Modèle Relationnel

- La modélisation logique des données conduira aux opérations suivantes :
- Transformation du MCD, en un Modèle Relationnel exprimé dans un formalisme logique adapté au SGBD envisagé
- Le Modèle Relationnel sera ensuite transformé et adapté en fonction des spécificités du langage de définition des données spécifique à l'outil retenu pour devenir Modèle Physique des Données.

# Le Modèle Logique de Données (M.L.D.)

- C'est la même chose que le Modèle Relationnel mais exprimé sous un formalisme différent
- Il est aussi exprimé sous forme de RELATIONS
- Il sera ensuite transformé pour devenir Modèle Physique des Données.

# DEMARCHE



# Le Modèle Relationnel

- Le modèle relationnel a été défini par E.F. Codd en 1970 à IBM San José
- Aspects fondamentaux du modèle relationnel :
  - une algèbre permettant de manipuler des tables et des relations
  - une démarche de conception permettant de définir une collection de relations

# Le Modèle Relationnel

- Un modèle est dit relationnel dans la mesure où il permet de parcourir la structure des données empruntant des chemins non prédéfinis, constitués en dynamique par des requêtes
- Les concepts du modèle relationnel découlent de la théorie des ensembles
- A ce type de modèle sont associées les notions suivantes :
  - domaine
  - relation
  - attribut
  - tuple (ou n-uplet)

# Le Modèle Relationnel

- Un domaine est un ensemble de valeurs ayant une signification pour l'utilisateur :
  - Ex: le domaine des noms, le domaine des âges,...
  - Ex: le domaine des entiers  $E=\{\dots-2,-1,0,+1,+2,\dots\}$
- La relation est définie comme un tableau de données
- Les colonnes sont les attributs et les lignes sont des tuples
- Un ou plusieurs attributs permettent d'identifier de façon unique chaque tuple de la table : la clé primaire de la relation
- La clé primaire est dite simple si elle est constituée d'un seul attribut et composée dans le cas contraire

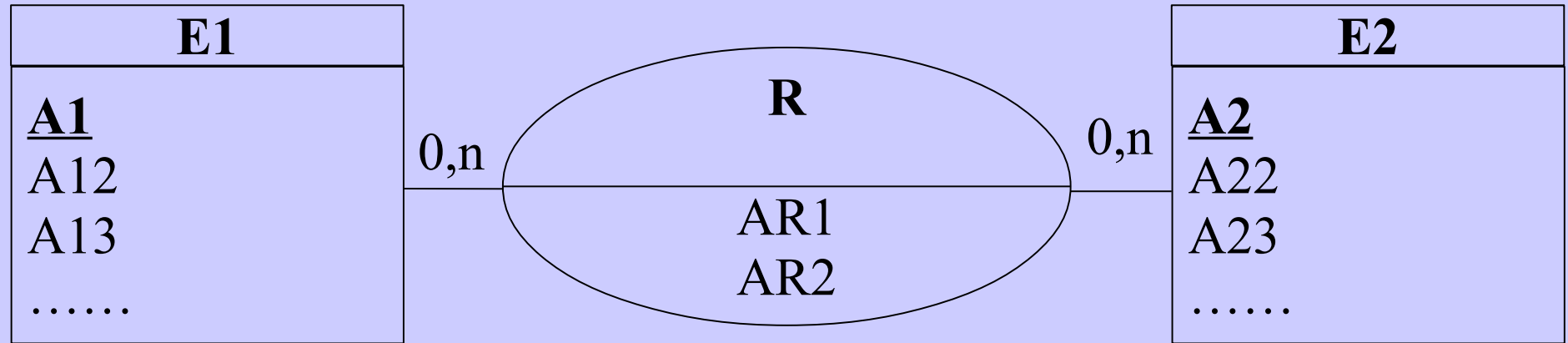


# Passage du M.C.D. au Modèle Relationnel

- Chaque entité du M.C.D. devient une relation du M.R.
- Une relation du M.C.D. ne devient pas forcément une relation du M.R. :

→ **Règles de passage du M.C.D. au M.R.**

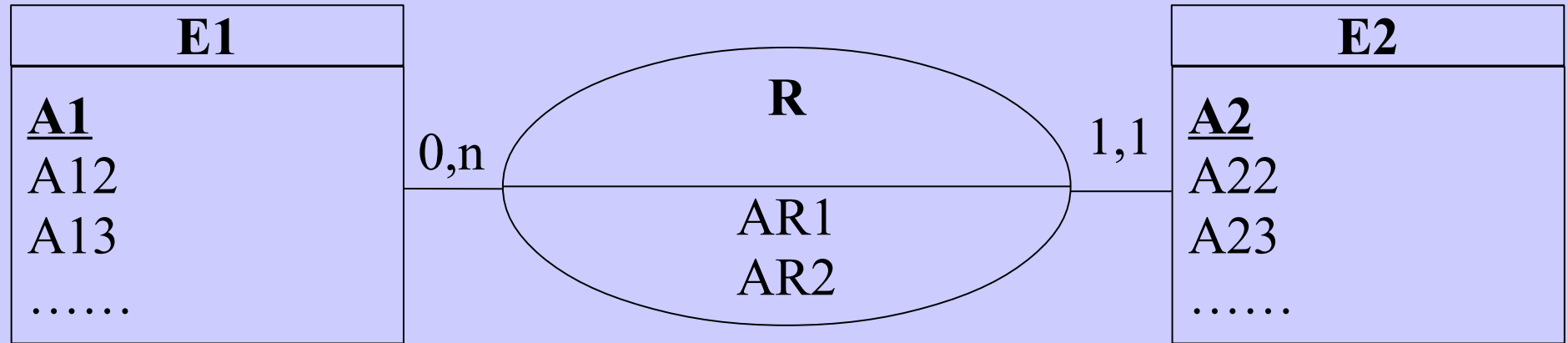
# Règles de passage du M.C.D. au M.R.



## 3 Relations :

E1(A1,A12,A13,...)  
E2(A2,A22,A23,...)  
R(#A1,#A2,AR1,AR2,)

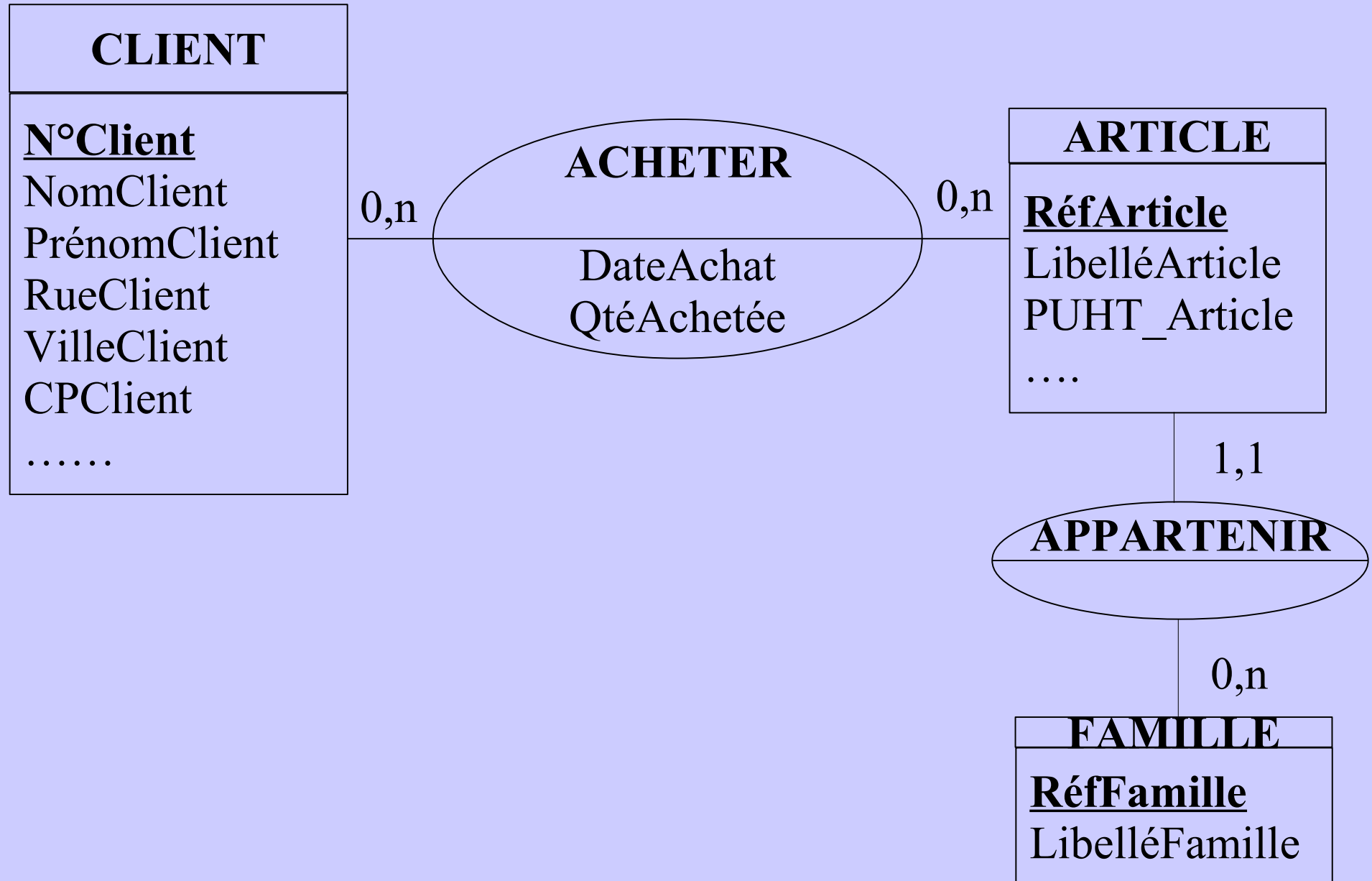
# Règles de passage du M.C.D. au M.R.



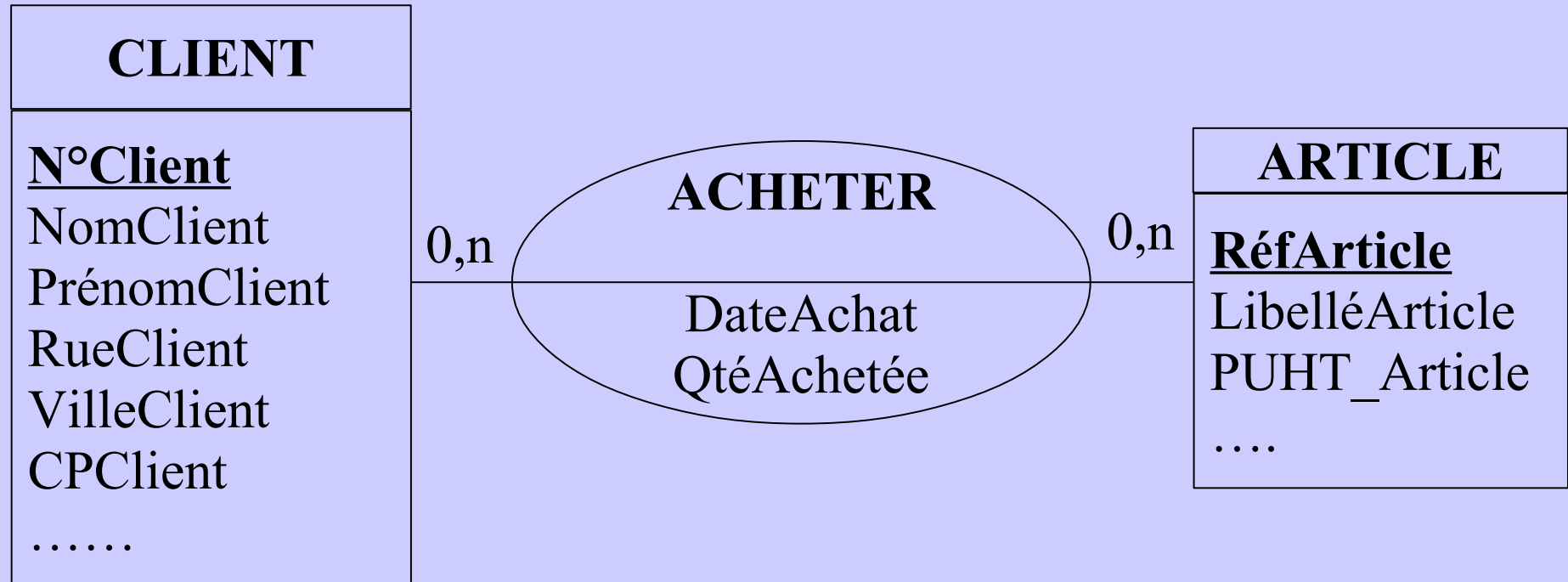
## 2 Relations :

$E1(\underline{A1}, A12, A13, \dots)$   
 $E2(\underline{A2}, A22, A23, \#A1, AR1, AR2, \dots)$

# Règles de passage du M.C.D. au M.R.



# Règles de passage du M.C.D. au M.R.



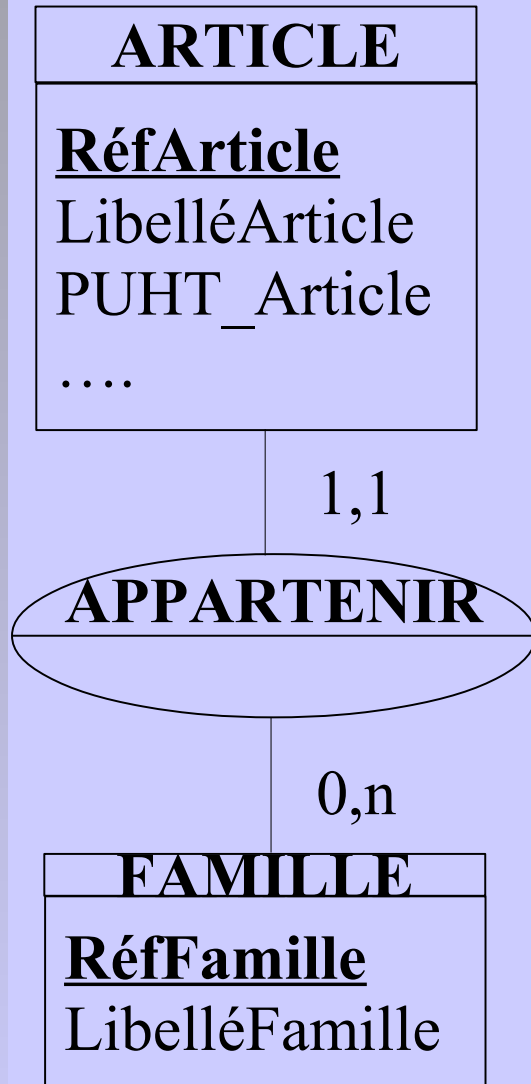
## 3 Relations :

CLIENT (N°Client, NomClient, PrénomClient, RueClient, VilleClient, CPClient, ...)

ARTICLE(RéfArticle, LibelléArticle, PUHT\_Article, ...)

ACHETER(#N°Client, #RéfArticle, DateAchat, QtéAchetée)

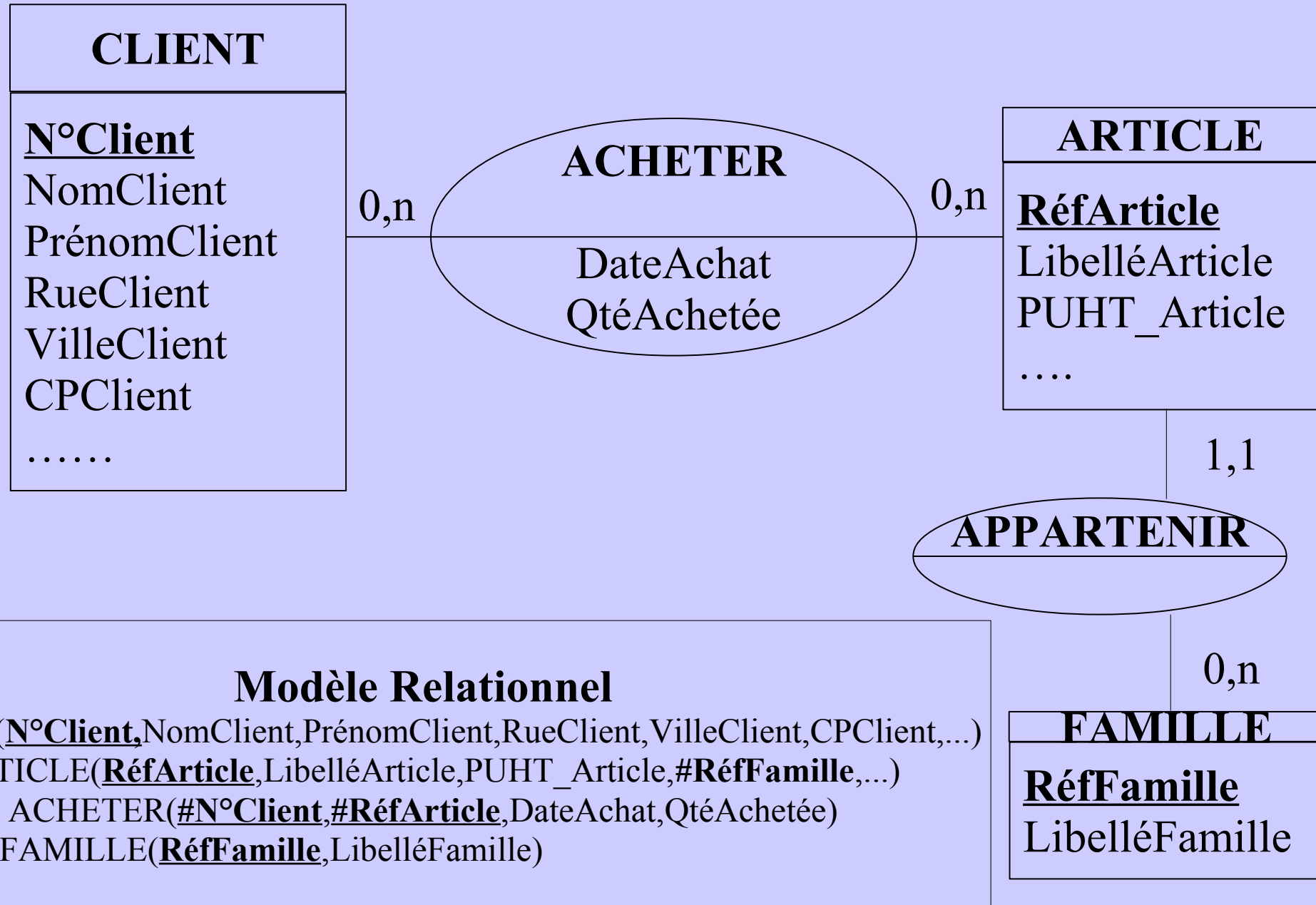
# Règles de passage du M.C.D. au M.R.



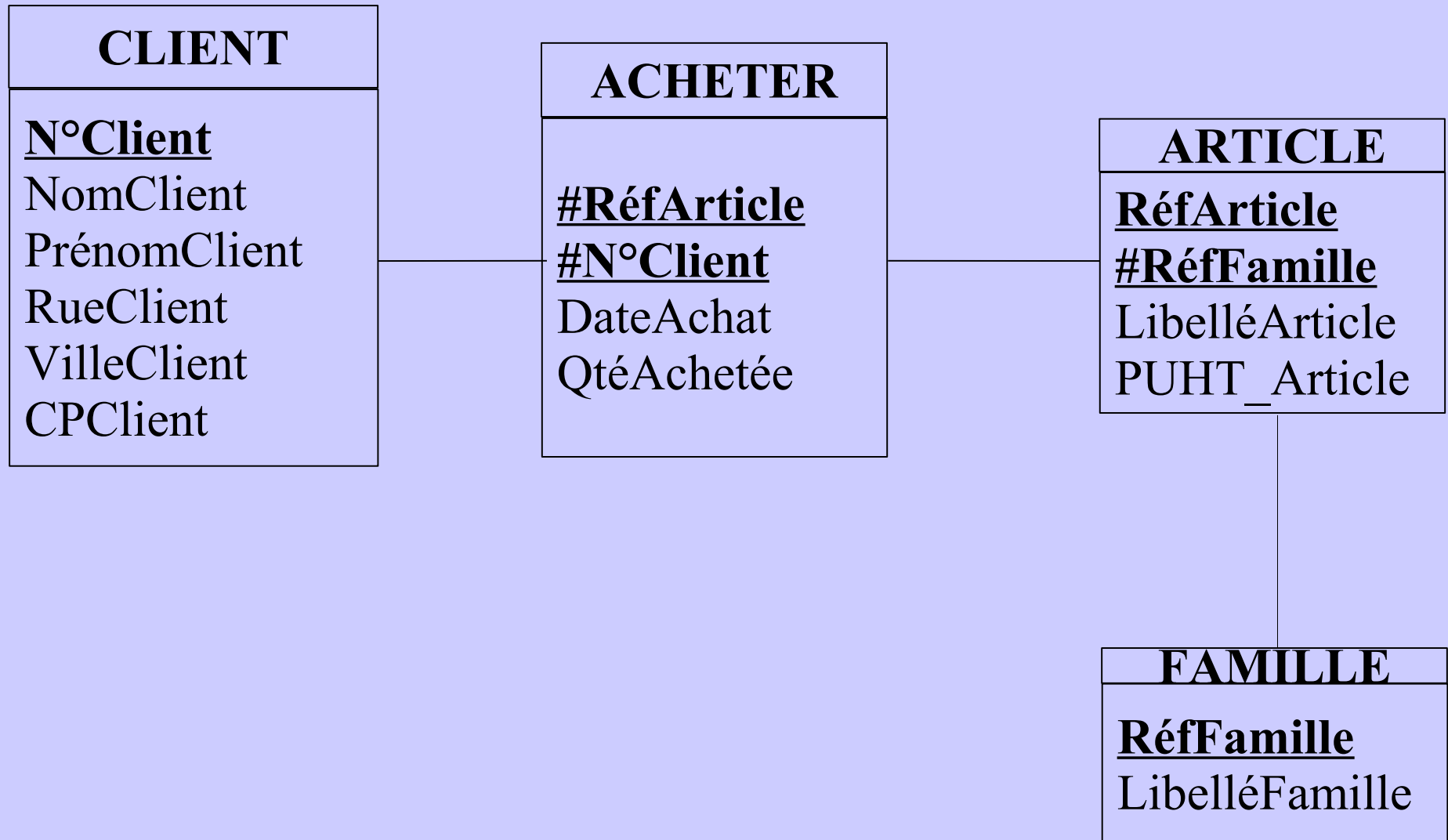
## 2 Relations :

ARTICLE (RéfArticle, LibelléArticle, PUHT\_Article, #RéfFamille...)  
FAMILLE(RéfFamille, LibelléFamille)

# Règles de passage du M.C.D. au M.R.

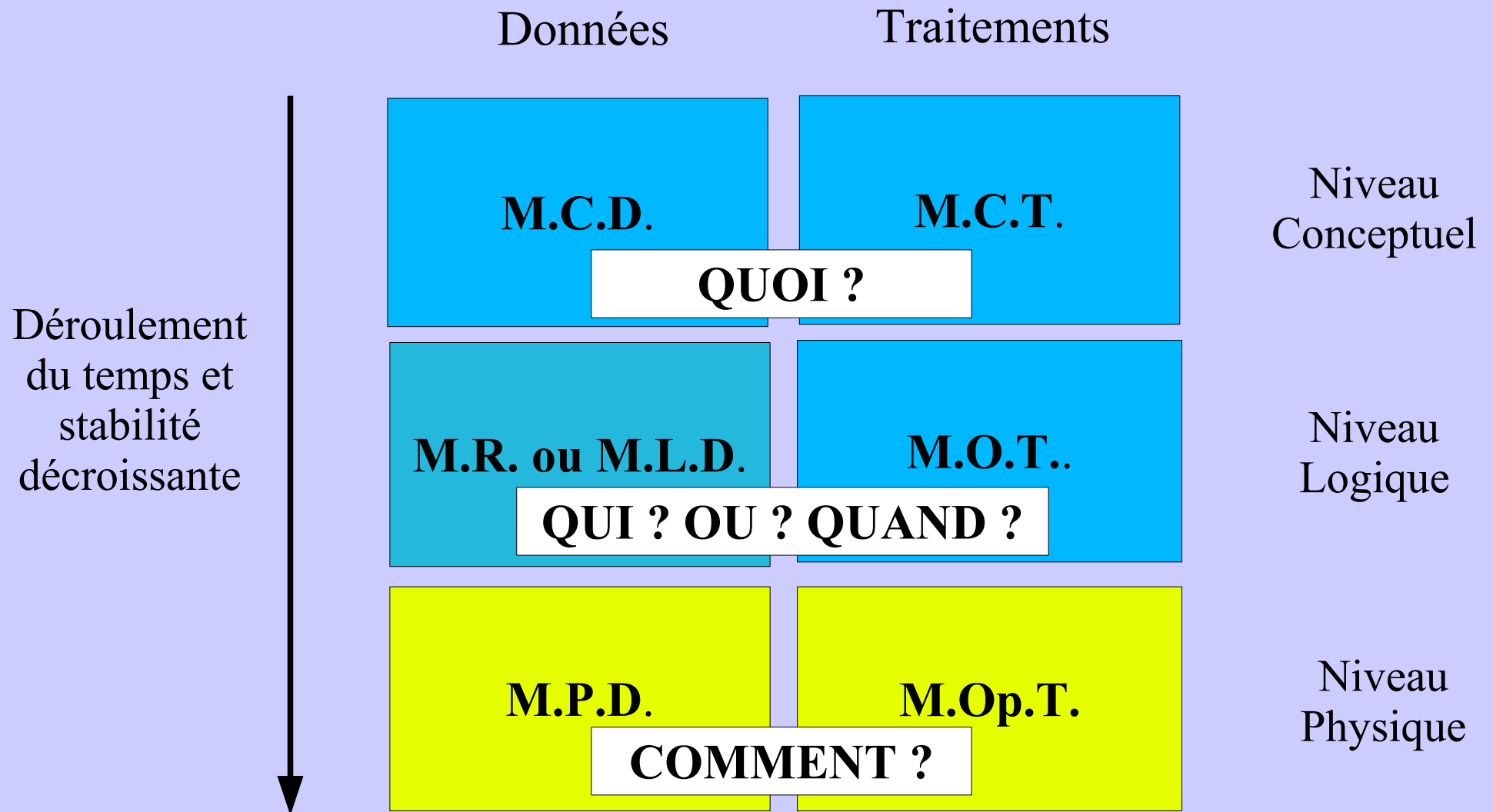


# Le Modèle Logique de Données





# Une approche par niveau



# *Modèle physique des données*

- Le MPD prépare le système de gestion des données.
- Le MPD s'intéresse à l'optimisation de la gestion des données en fonction de l'outil choisi pour cette gestion et surtout en fonction des traitements qui utilisent ces données (Vue des procédures fonctionnelles).
- Définir l'implantation physique de la base de données sur les disques, les serveurs disponibles ...

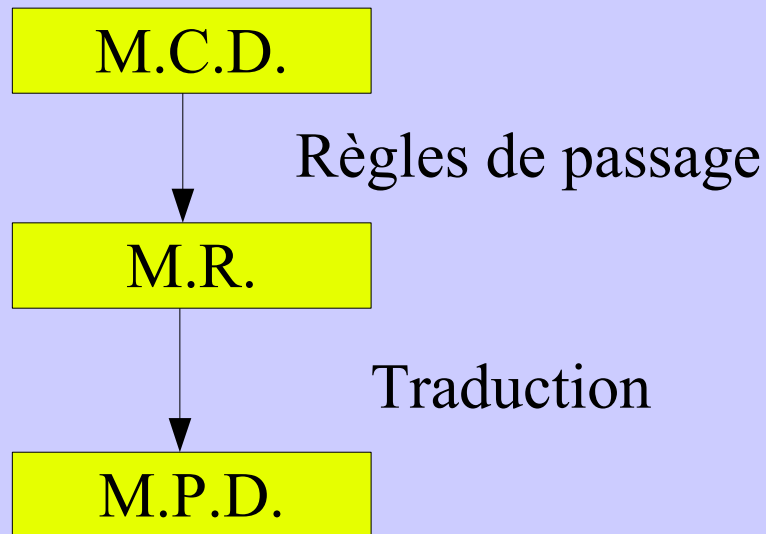
# ***Modèle opérationnel des traitements***

- Le MOPT s'intéresse à la structure interne de toutes les applications du projet. Son objectif est la préparation du développement :
  - Définir les normes de développement, si celles-ci n'appartiennent pas déjà au schéma directeur.
  - Décomposer chaque application en modules techniques :
    - Définir les données internes au module technique.
    - Définir les traitements du module technique (Procédures, fonctions) :
      - Présentation du traitement technique.
      - Appel du traitement technique.
      - Informations en entrée.
      - Informations en sortie.
      - Résultat.
      - Données internes au traitement technique.
      - Description du traitement technique (pseudo-code, algorithme ...).
  - Définir le cahier des tests.

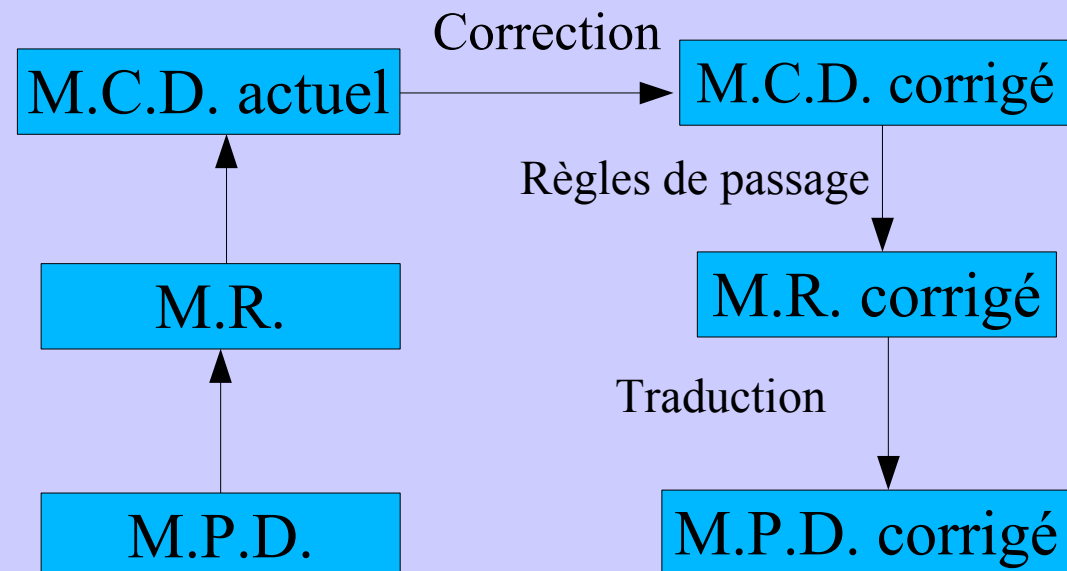
# REVERSE ENGINEERING

- Consiste à partir du schéma d'une base de données existante et à remonter jusqu'au M.C.D. afin :
  - de le corriger
  - d'intégrer de nouvelles contraintes
  - de le redocumenter

## Démarche d'analyse



## Reverse Engineering



Un garagiste qui travaille pour des clients français et étrangers désire créer une base de données avec les caractéristiques suivantes :

- Les voitures sont identifiées par leur NoChassis, l'année de construction, la marque, la couleur, la cylindrée et le type (Berline, voiture de sport, Break,...), la marque de pneumatique, la garantie d'usine.
- Le propriétaire (d'une ou plusieurs voitures) est identifié par son nom, prénom, adresse, ville, pays et NoTel.
- Les garanties d'usine dépendent uniquement de la marque et sont caractérisées par une durée et un type (exemple : pièces, pièces et main d'oeuvre, etc...).
- Pour chaque réparation sur une voiture, le Nom de l'employé, le type de réparation, la date et les frais de réparation sont mentionnés.
- L'employé est identifié de la même façon qu'un propriétaire, avec en plus son salaire mensuel, sa caisse de retraite, sa qualification.

On fait l'hypothèse suivante : Il n'existe pas de nom de ville identique dans d'autres pays.

1- Construire le MCD modélisant le cahier des charges ci-dessus (en précisant vos propres hypothèses).

2 - En déduire le schéma relationnel en mentionnant, pour chaque relation, les attributs, les identifiants ainsi que les clefs étrangères.

Un concepteur (débutant) vient de créer une base de données relationnelle pour la gestion des véhicules de la société dans laquelle il travaille. Il vous demande si celle-ci est correcte en fonction des renseignements suivants qu'il vous a donnés :

- une voiture est affectée à un ou plusieurs services
- le type de voiture : R19, 306, ...
- la consommation dépend du type uniquement
- un employé ne travaille que dans un seul service
- on désire suivre les déplacements faits par un véhicule et par qui au cours d'une journée.

Rectifiez (en justifiant vos modifications) si nécessaire, la base de données.

Structure de la base :

VOITURE(N°Immat, conso, type\_voiture, marque, date1ère\_mise\_circul, date\_chgmt\_service, #N°Service)

SERVICE(N°Service, adresseS, Nom\_Responsable)

EMPLOYE(N°Empl, nom, prénom, adresse, #N°Sservice)

UTILISATION(date, N°Empl, N°Service, N°Immat, NbKms)

Soit le schéma de la base de données suivant :

CLIENT(N°Client, raison\_sociale, adresse, téléphone)

PRODUIT(N°produit, désignation, prix, #typeProd)

TYPE\_PRODUIT(typeProd, désign\_type)

COMMANDE(N°Comm, dateComm, #N°Client)

LIGNE\_COMMANDE(N°ligne, N°Produit, N°Comm, QtéComm)

- 1 – Expliquer en langage naturel ce que fait cette base de données
- 2 – Construire le MCD associé