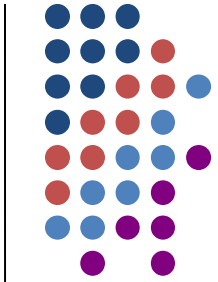
Eléments de conception des BDD relationnelles

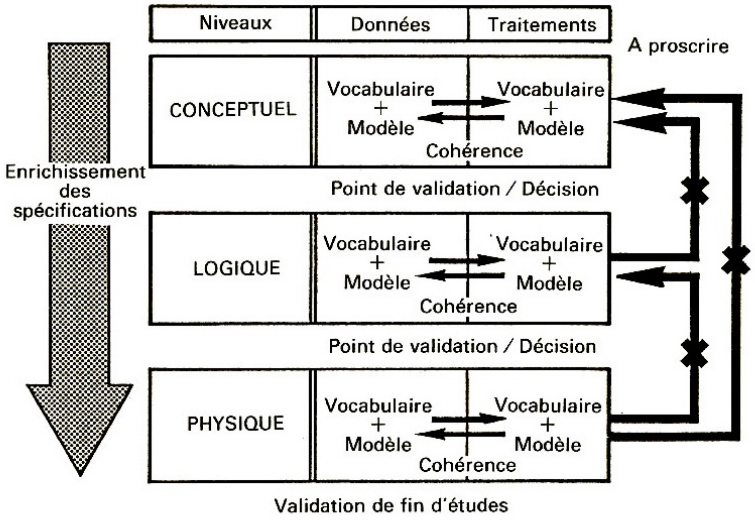
La méthode MERISE



- Méthode française, introduite en 1977 par H. Cardieu
- Motivation originelle : doter les administrations publiques d'une méthodologie d'analyse rigoureuse pour organiser leurs traitements et données de masse
- Prévue pour un spectre d'application plus large que les SGBD
- Propose une analyse séparée des données et des traitements (même si les 2 sont fortement interconnectés)
- Préconise une progression du plus abstrait (quoi faire? Quels problèmes?) au plus concret (comment implémenter?) via différents niveaux
- Produit des documents formalisés pour chaque niveau

La méthode MERISE

- Niveau conceptuel : produit le **MCD** = **modèle conceptuel des données** = représentation abstraite des données et traitements + le MCT = modèle conceptuel des traitements
- Niveau logique : produit le **MLD** = **modèle logique de données** = MCD implantable, enrichi par la volumétrie et une estimation des temps d'exécution + le MLT
- Niveau physique : produit le **MPD** = **modèle physique des données** = description des supports et méthodes de stockages + le MPT



La méthode MERISE préconise d'analyser séparément données et traitements, à chaque niveau.

(source: Wikipedia)

Les niveaux sont indépendants les uns des autres

La méthode MERISE

Dans notre cours:

- Nous ne traitons que les modèles conceptuel (MCD) et logique (MLD) de données : nous ne couvrons ni les modèles de traitement (par manque de temps), ni les modèles physiques (hors sujet)
- Un MCD consistera en un modèle Entité-Association (Entity-Relationship model)
- Un MLD consistera en un modèle relationnel.

MCD = modèle entité-assocation
MLD = modèle relationnel

On pourrait faire d'autres choix concernant le MLD (modèle objet par exemple).

Modèle entité-association

Vue d'ensemble

La production d'un MCD complet nécessite de décrire :

- Les données, et ce qu'elles représentent
- Les liens sémantiques qui existent entre elles
- Les contraintes (les plus simples) qu'elles doivent satisfaire

Le standard largement répandu pour cette description est le **modèle entité-association** (modèle E-A).

Ce modèle est **purement conceptuel** : il se contente simplement de décrire les 3 points ci-dessus sans jamais dire comment les données, liens sémantiques, et contraintes doivent être implémentés.

Modèle entité-association

Vue d'ensemble

- Il est ensuite converti en un modèle logique de données (modèle relationnel dans notre cours), qui représente cette fois-ci une solution implantable sur machine.
- Modèles E-A et relationnel ne doivent donc pas être confondus.
- Terminologie : association = relationship, mais **association ≠ relation** ! « entité-relation » est une très mauvaise traduction (bien que très répandue) de l'anglais « *entity-relationship »,* et doit être proscrite.
- Deux standards de représentation graphiques co-existent : la représentation originelle (par losanges, ellipses, rectangle, et traits) et UML. Ils codent toutefois la même chose.
- En TP, vous utiliserez le modeleur Oracle (qui suit le standard UML) ou JMerise (standard MERISE). Les DBA sont amenés à utiliser les 2 standard presque quotidiennement…

Modèle entité-association

Entités et Attributs

- Une ***entité*** est une classe abstraite représentant des objets du monde réel. Ces objets, abstraits ou concrets, peuvent être **distingués et identifiés** de façon unique.
- Ils possèdent tous les mêmes propriétés, appelées ***attributs***.
- *Exemples*
  - entité-type *constructeur*, avec attributs : id_*constr, nom*
  - entité-type *voiture*, avec attributs *num_serie, couleur.*

- Les attributs peuvent être simples ou composés
- Les attributs prennent des ***valeurs*** lorsqu'ils sont définis (ce qui n'est pas toujours obligatoire).
- Exemple
  - *couleur* peut prendre les valeurs *blanc, noir, gris, …*

- *Remarque* : les valeurs n'apparaissent **jamais** dans le modèle E-A.

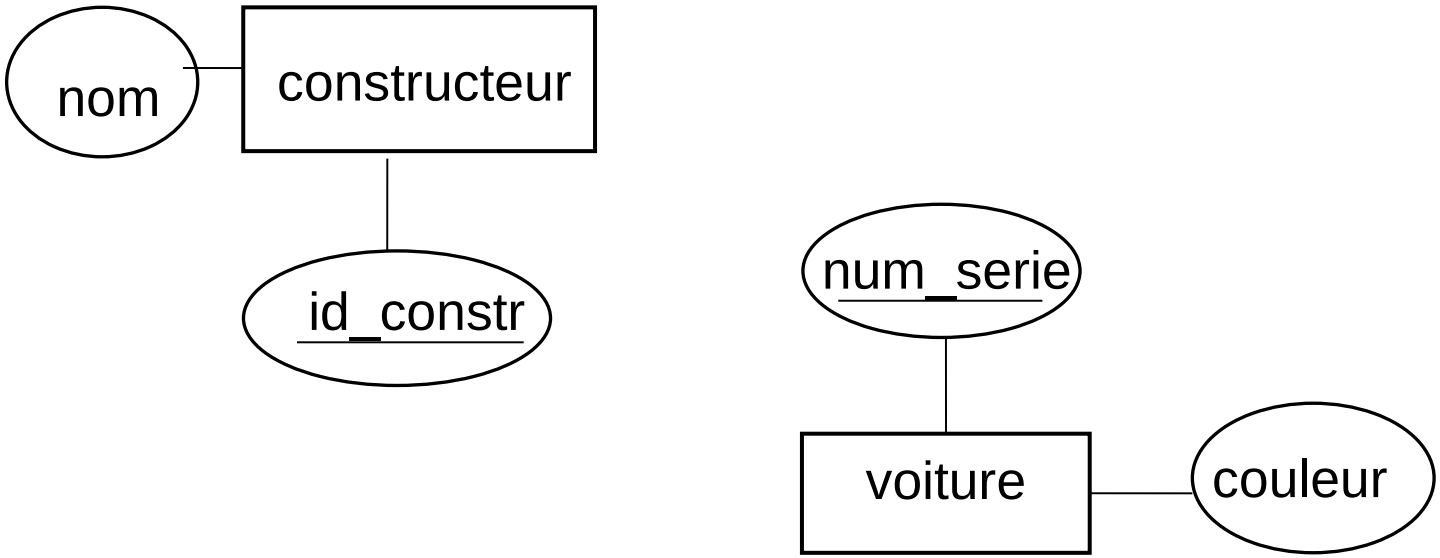
Modèle entité-association

Clé primaire d'une entité

- Un ensemble minimal d'attributs qui identifient une entité de manière unique forment ***une clé***. Parmi toutes les clés possibles, on en choisi une : c'est la ***clé primaire*** (ou **identifiant**) La clé primaire est soulignée dans la représentation graphique.
- *Exemple*
  - *id_constr* est clé primaire de l'entité-type *constructeur*
  - *nom* est une clé candidate de cette entité-type
  - *num_serie* est clé primaire de l'entité-type *voiture*.

Représentation graphique

Illustration d'entités, avec clés primaires et attributs



Modèle entité-association

Associations

- Une ***association*** est une correspondance (au sens mathématique du terme) entre au moins deux entités-types données.
- Le ***degré*** d'une association est le nombre d'entités-type qu'elle met en jeu. Une association de degré p peut donc être vue comme un ensemble (aussi grand soit il) de p-uplets vers des références à des entités particulières.
- Représentation : par un losange.

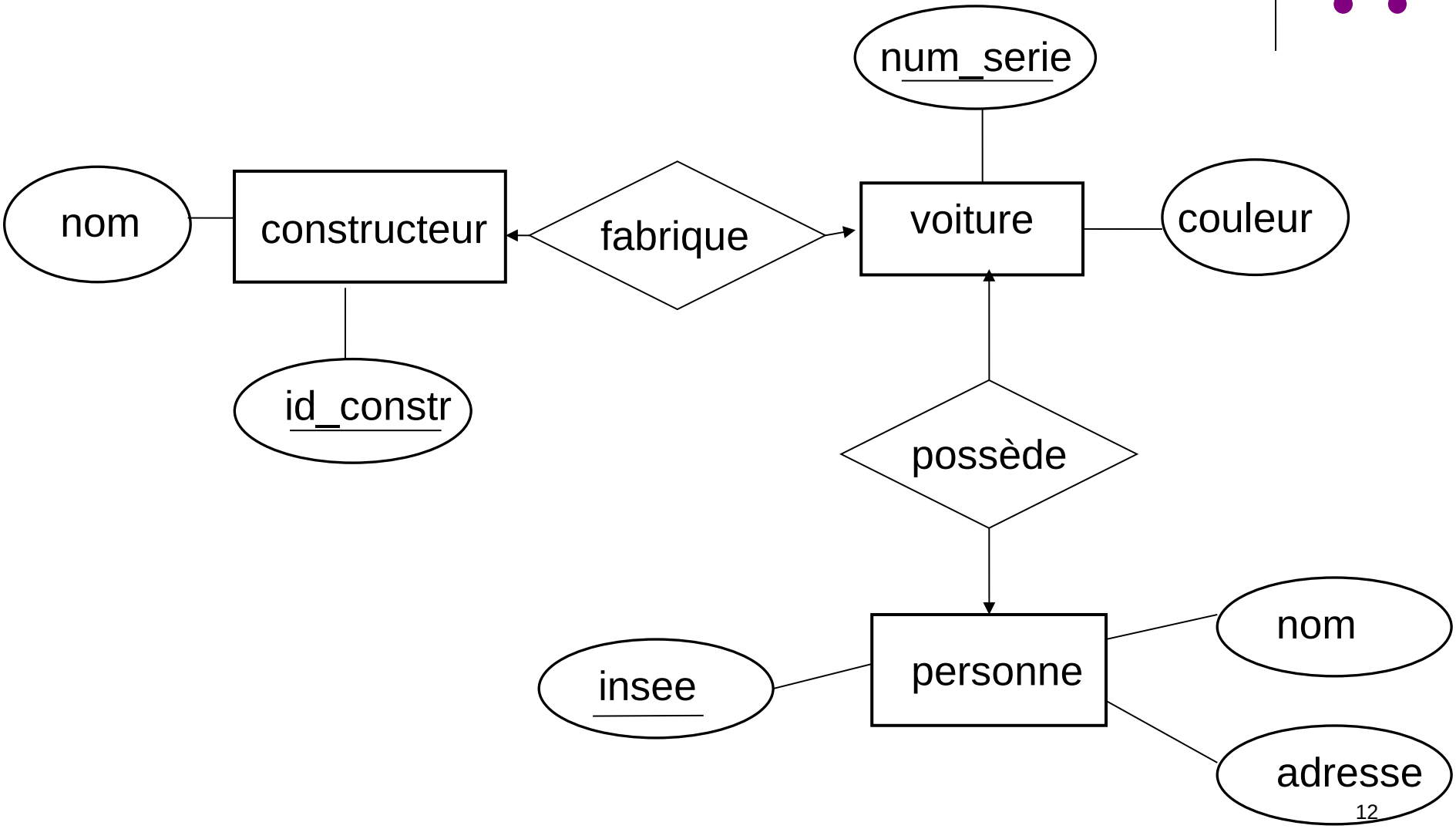
Modèle entité-association

Exemples d'associations

- Ajoutons au modèle précédent l'entité-type
  - *personne* (insee,nom,adresse)
- On peut définir deux associations
  - *possède* entre les entités voiture et personne, permet de savoir quelle personne possède quelle voiture
  - *fabrique* entre voiture et constructeur, permet de savoir quel constructeur fabrique quelle voiture

Modèle entité-association

Diagramme 1 : entités et associations



Modèle entité-association

Entités concrètes et abstraites

- Une entité ne représente pas toujours des objets physiques : c'est le cas lorsqu'on n'a pas besoin de donner les caractéristiques des objets physiques individuellement
- *Exemple*
  l'entité-type ***modèle de voiture*** avec attributs *appellation*, *carburant, puissance*, *ddc (date de commercialisation = date officielle d'autorisation de mise sur le marché), …* est une classe abstraite. Chaque modèle sera peut-être vendu à des millions d'exemplaires.
- Noter que l'attribut *ddc* est composé (jj/mm/aaa)

Modèle entité-association

Entités faibles

- Dans tous les cas, chaque objet qui se conforme à la description d'une entité est appelé ***occurrence*** ; et il doit représenter un objet unique (une référence catalogue unique dans notre cas)

- Pour qu'un modèle de voiture puisse être identifié, on doit faire référence au constructeur : l'entité-type *modèle-de-voiture* est une entité « faible »

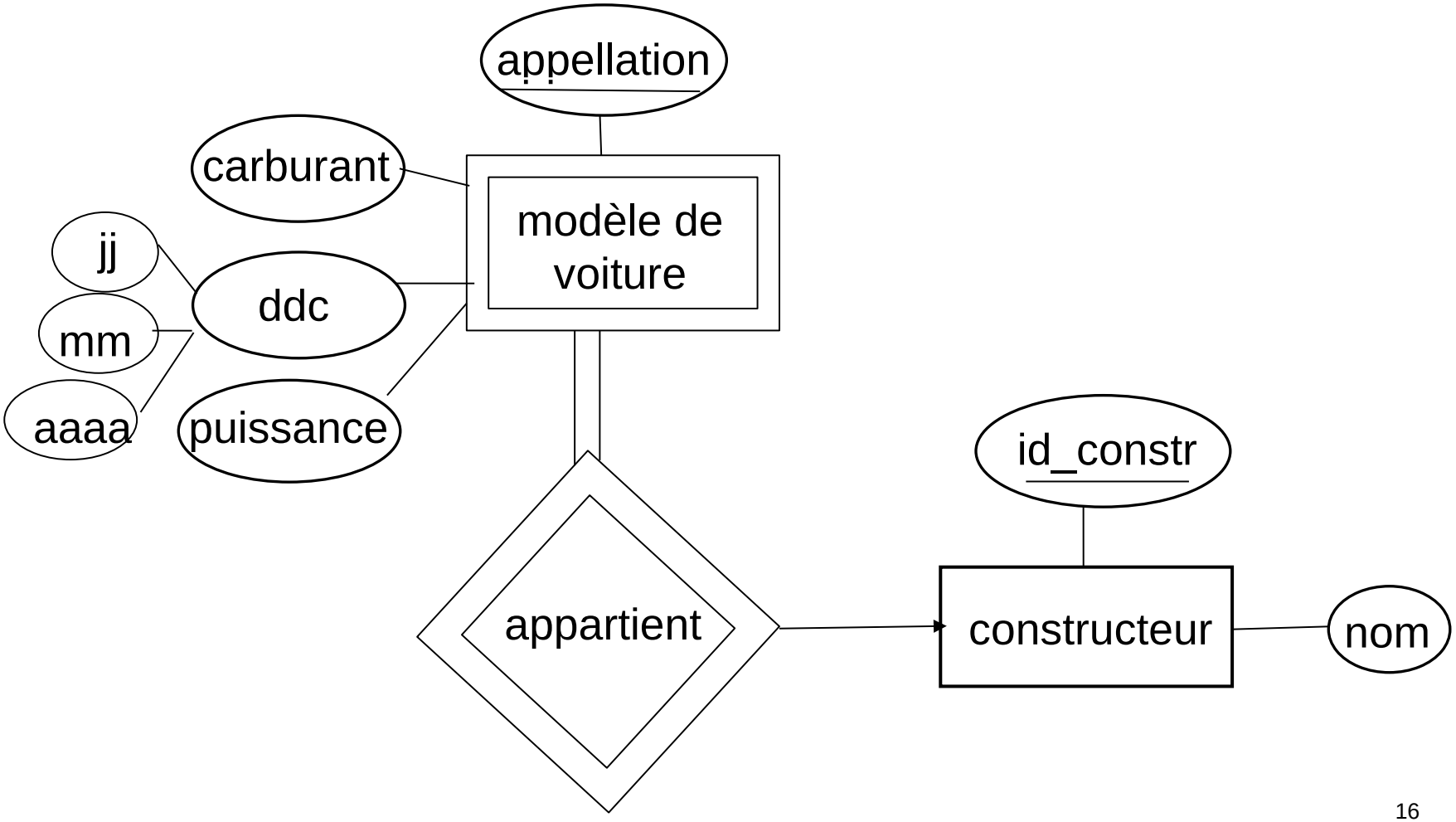
Modèle entité-association

Entités faibles

- Une entité E est dite faible si ses éléments ne peuvent pas être identifiés à l'aide de leurs seuls attributs.
- Pour identifier ses éléments on doit suivre une ou plusieurs associations à partir de E et considérer simultanément les clés des entités parcourues.
- La clé primaire d'une entité faible est obtenue en la complétant par celles des entités fortes qui la supportent
- Représentation de l'entité faible et de la relation identifiante avec double trait
- Exemple : dans le schéma E-A qui suit, la clé primaire du modèle de voitures, entité faible, n'est plus *appellation* seule ; mais le couple (*appellation,id_constr*)

Modèle entité-association

Diagramme 2



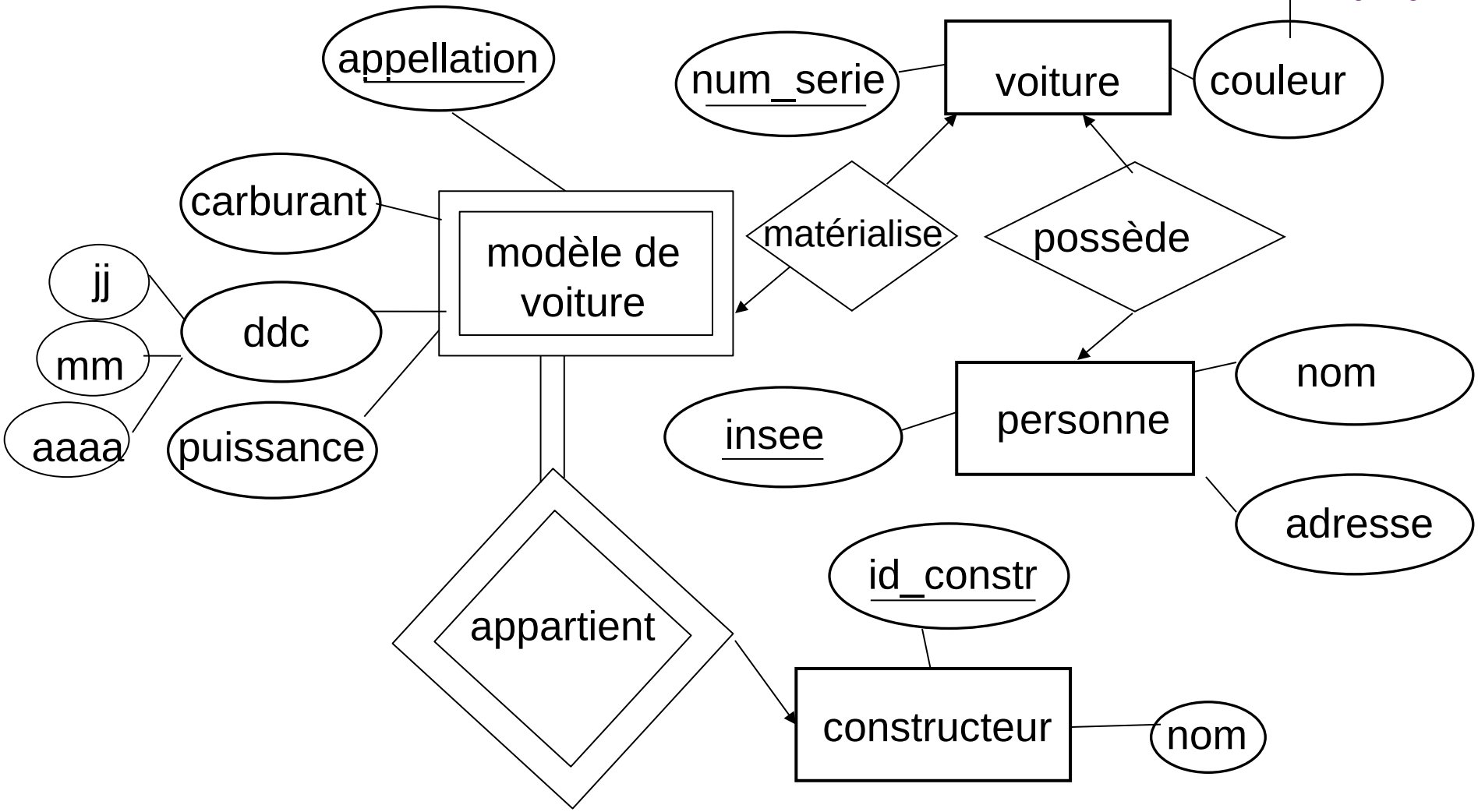
Modèle entité-association

Evolution du Modèle entité – association

- Nous pouvons désormais représenter le fait qu'une voiture matérialisée appartient à un modèle et qu'inversement un modèle peut se matérialiser en plusieurs voitures
- Nous introduisons une association entre les deux entités *voiture* et *modèle de voiture*. L'association *fabrique*, entre *voiture* et *constructeur*, devient redondante.
- Le modèle E-A correspondant est le suivant

Modèle entité-association

Diagramme 3, combine les diagrammes 1 et 2

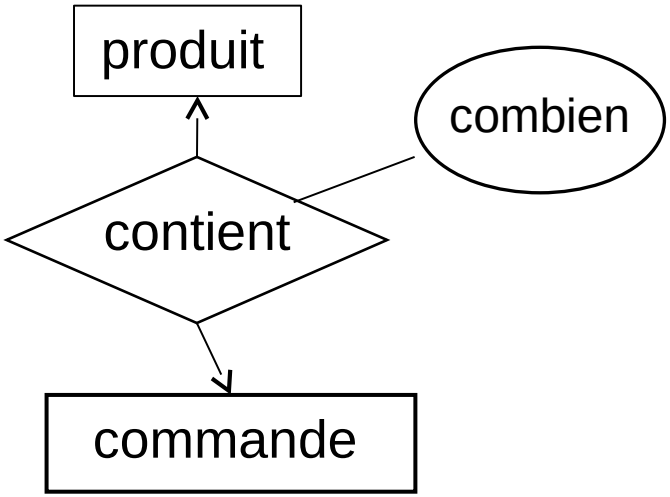


Modèle entité-association

Attribut d'association

Un attribut d'association est une propriété qui paramètre ou complète les entités liées par l'association.

Exemple 1 : une commande contient une certaine quantité de produits indistinguables. La quantité complète chaque mise en correspondance (*produit,commande*) et est nécessaire

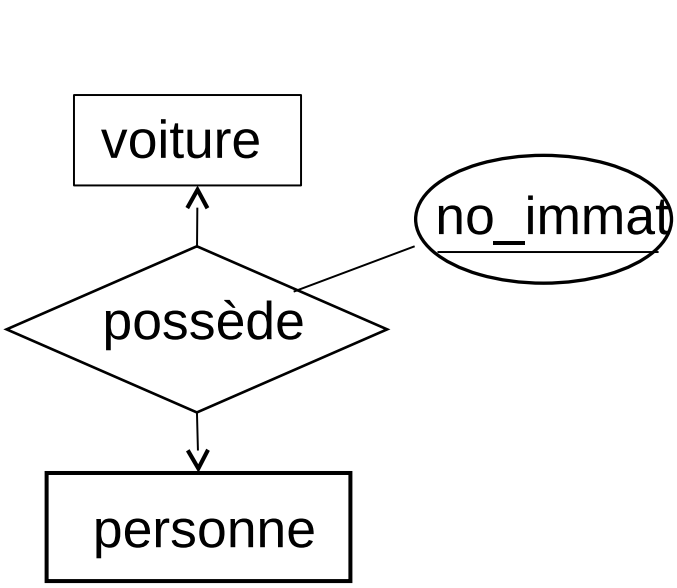


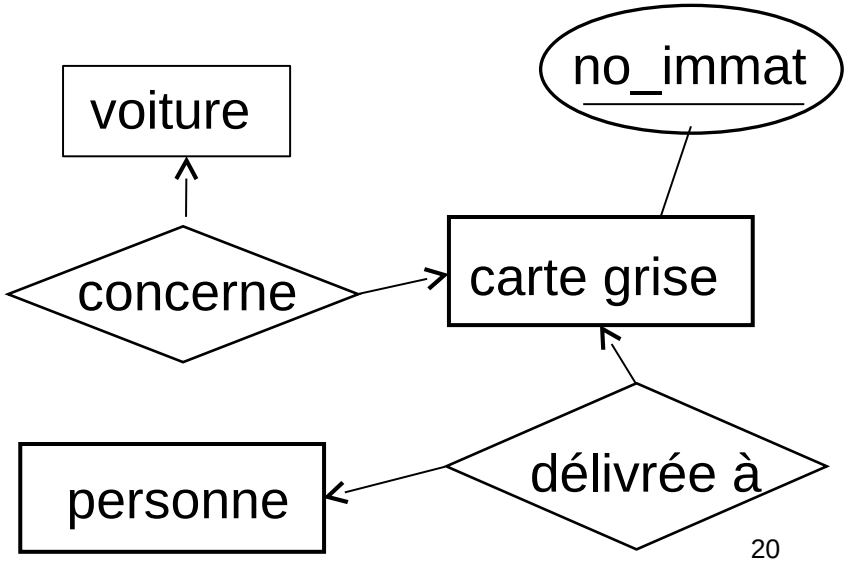
Exemple 2 (diapos suivantes) : une carte grise est délivrée à une date donnée

Modèle entité-association

Attribut d'association

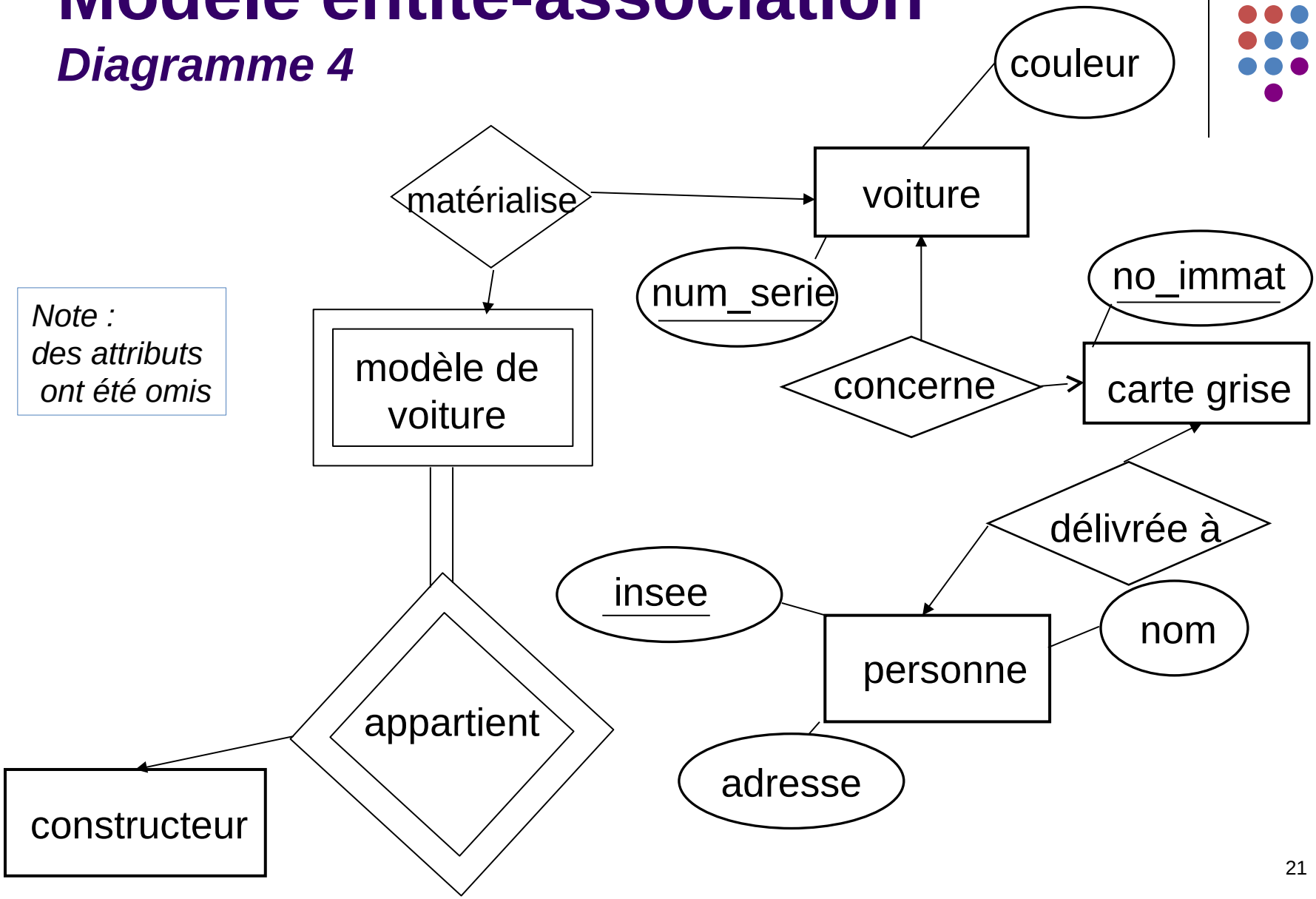
- Supposons que la possession soit matérialisée par la détention d'une carte grise par le propriétaire de toute voiture
- Ces possessions sont matérialisées par le numéro minéralogique (unique par pays) de la carte grise
- On peut être tenté soit de rajouter un attribut "no_immat" à la relation *possède* (qui en serait une clé), soit de créer une nouvelle entité "carte grise". La deuxième solution est **préférable**.





Modèle entité-association

Diagramme 4



Note :
des attributs ont été omis

Modèle entité-association

Volumétrie

Devant chaque entité impliquée dans une association, on indique un couple (m,p) appelé ***participations*** (ou ***cardinalités***) :

- m est le plus petit des nombres de fois où une instance de l'entité est susceptible d'être liée à la relation
- p est le plus grand des nombres de fois où une instance de l'entité est susceptible d'être liée à la relation

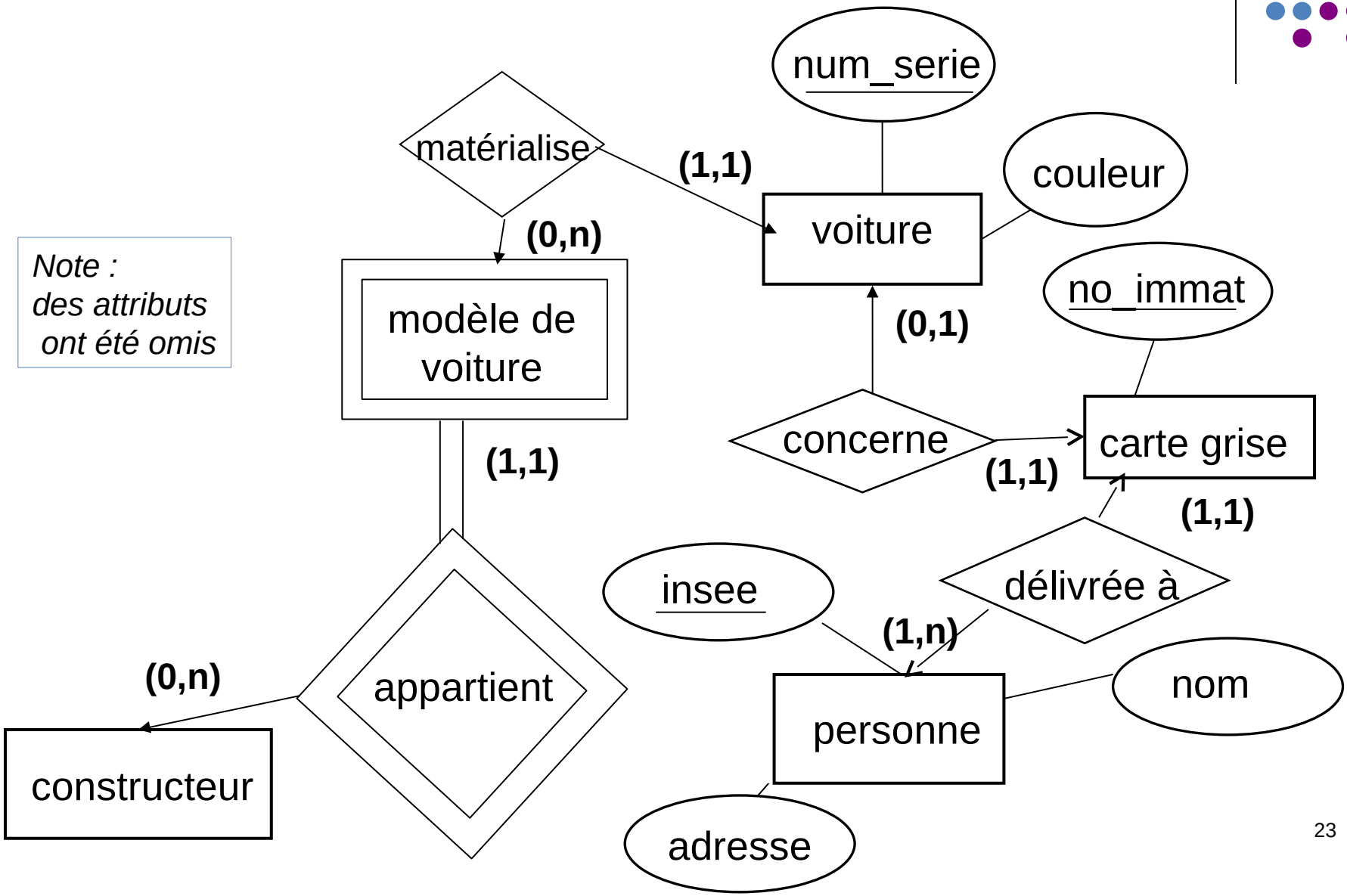
Lorsque rien ne peut être dit sur m, il doit valoir 0. Lorsque rien ne peut être dit sur p, il doit valoir l'infini (noté n, * ou ∞ indifféremment).

Exemple

Hypothèses: Toute voiture ne matérialise qu'un seul modèle, et n'est immatriculée au mieux qu'une fois. Toute carte grise ne peut être détenue que par une seule personne. Toute personne possède au moins une carte grise.

Modèle entité-association

Diagramme 5 = diagramme 4 + cardinalités



Note :
des attributs ont été omis

Modèle entité-association

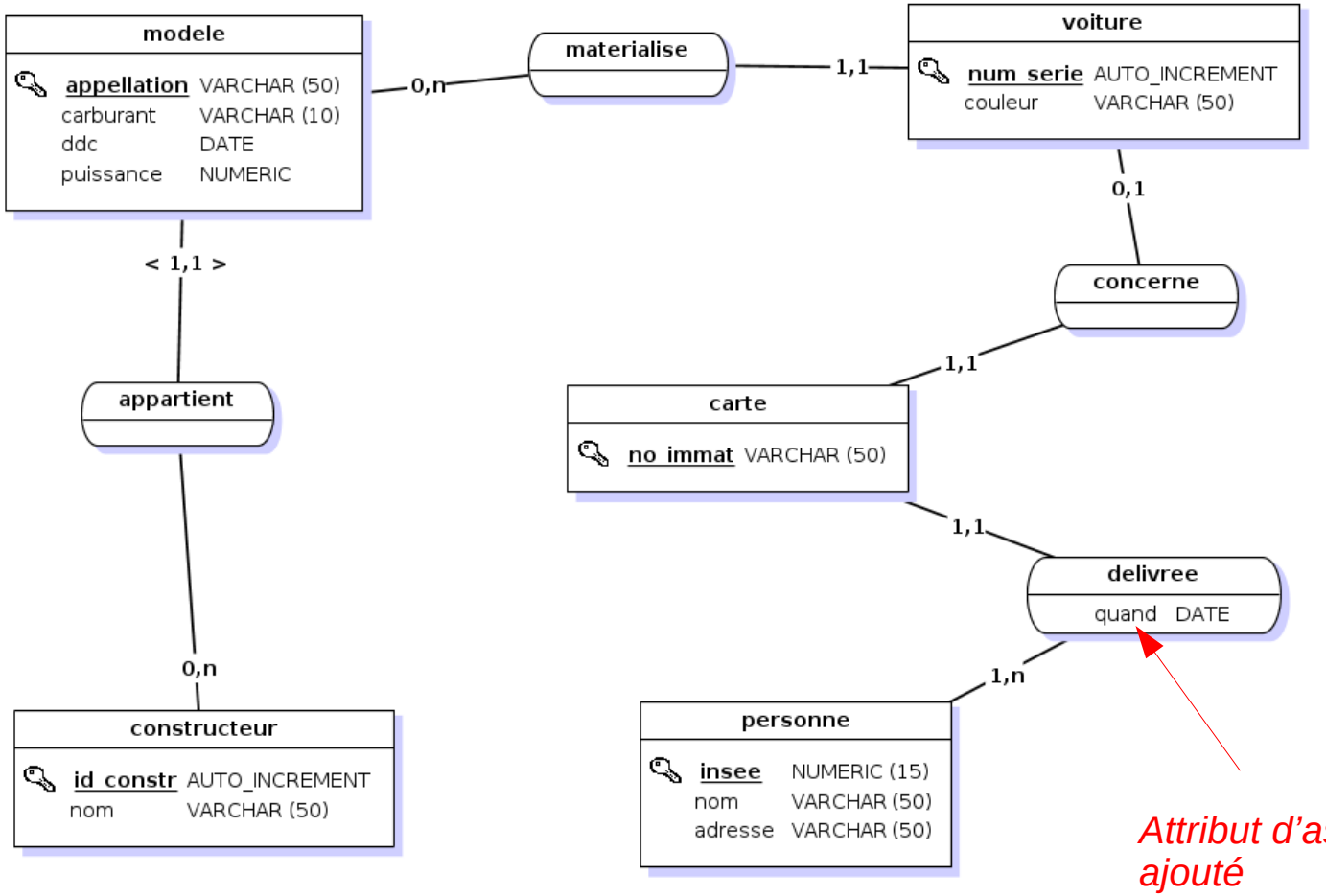
Volumétrie

Remarques

- Un seul objet susceptible de ne jamais être lié à la relation suffit à entraîner m=0 pour l'entité correspondante
- Un seul objet susceptible d'être lié n fois à la relation suffit à entraîner $p \geq n$ pour l'entité correspondante
- Les participations les plus courantes pour les relations binaires sont (0,1), (0,*), (1,1), et (1,*), mais tous les couples (m,p) $m \leq p$ sont en théorie permis.
- Des participations de (1,1) sont dites identifiantes pour la relation sur laquelle elle porte. Elles se rencontrent systématiquement du coté des entités faibles.

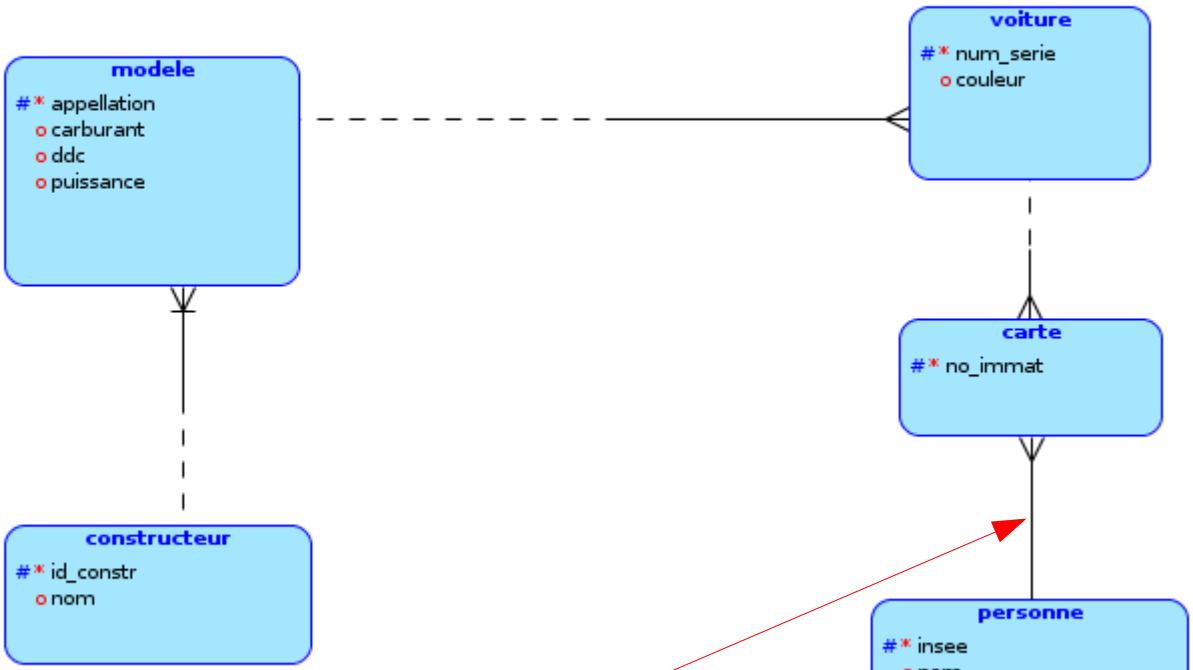
Modèle E-A (JMerise)

Modèle équivalent sur JMerise



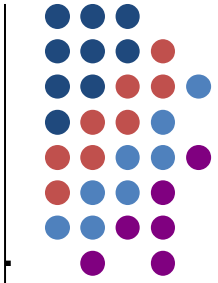
Modèle E-A (Oracle)

Modèle équivalent (ou presque) sur Oracle Data Modeler



Attribut d'association « quand » existant mais non affiché

Modèle E-A (Oracle)



Différences remarquables par rapport à la représentation MERISE :

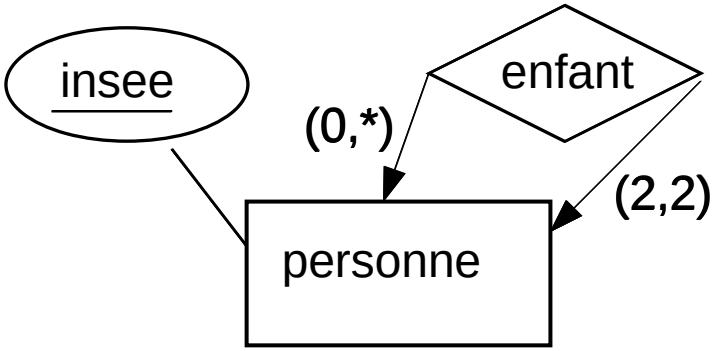
- Attributs listés à l'intérieur des entités
- Un attribut obligatoire est précédé d'un '*'
- Un attribut participant à la clé primaire d'une entité est précédé de '#'
- Dans une relation binaire :
  - Un trait plein indique une participation inf de 1
  - Un trait pointillé indique une participation inf de 0
  - Une « patte » indique une participation sup de '*' de l'autre côté de celui où elle positionnée
  - Un trait (orthogonal) indique une entité faible (relation dite « identifiante ») du côté où il se trouve

- Pas de possibilité de représenter des associations ternaires, quaternaires: il faut créer une nouvelle entité explicitement

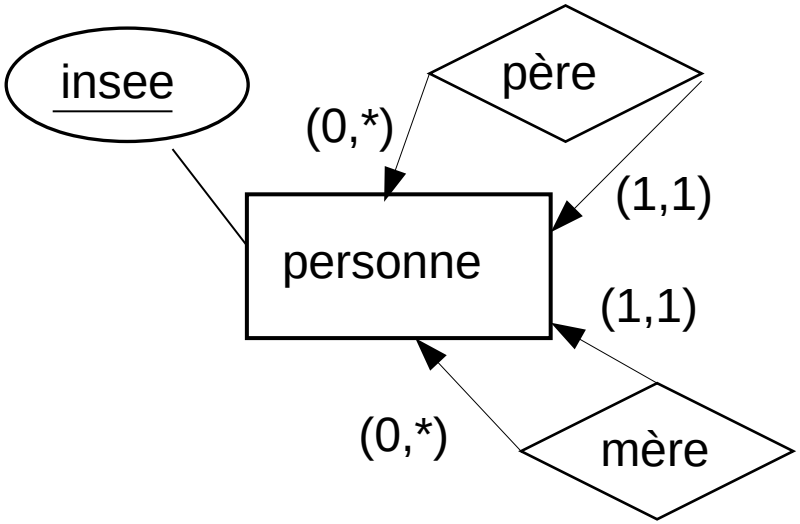
Associations réflexives

Une association peut très bien lier une entité avec elle-même.
Dans ce cas, on parle d'***association réflexive***.
Exemple typique : représentation de la filiation entre personnes ; deux solutions possibles

Solution 1



Solution 2



Remarque : la solution 2 est préférable (les bornes (2,2) ne sont pas simples à mettre en oeuvre)

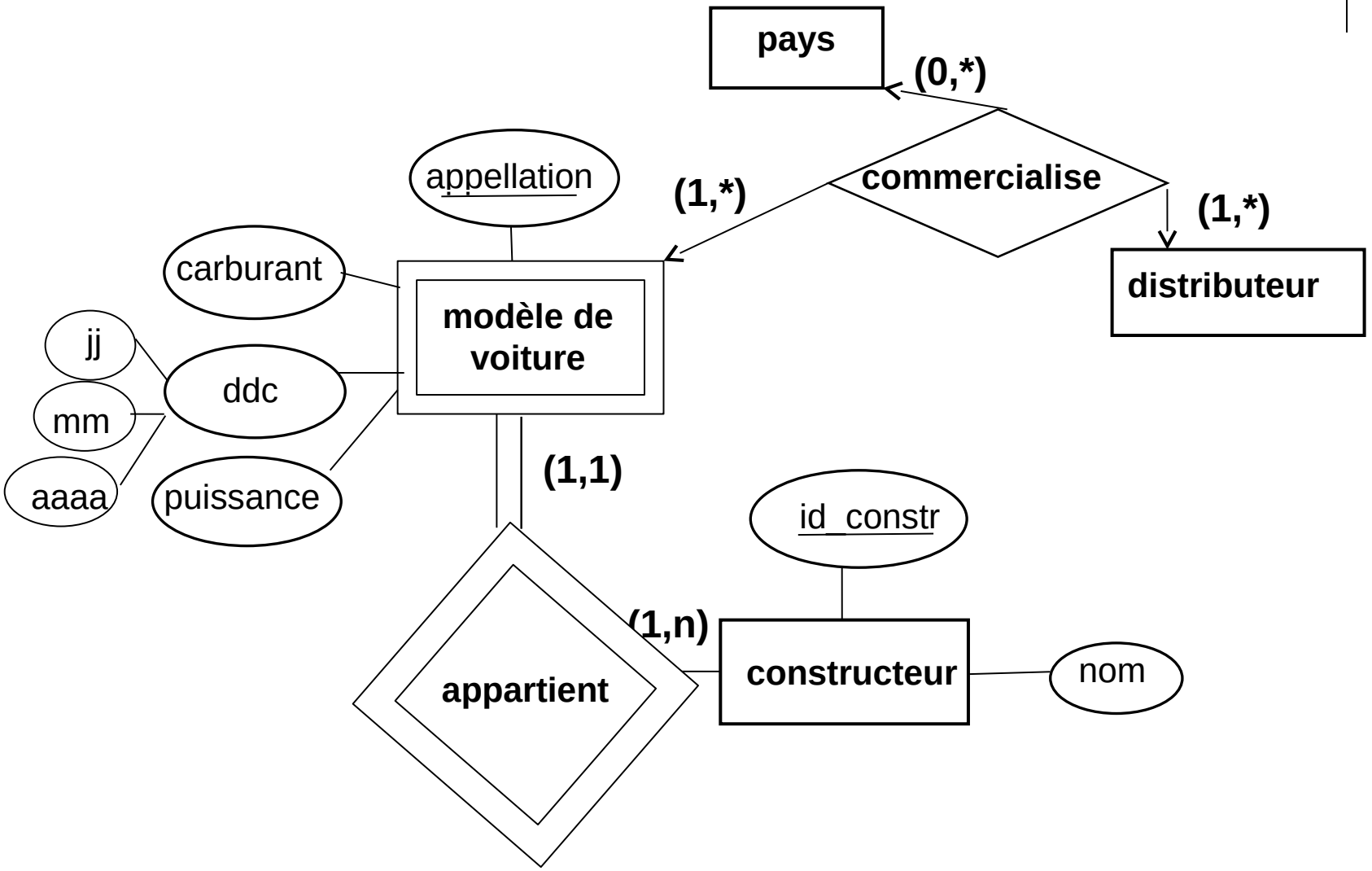
Associations de degré > 2

Lorsqu'une association porte sur plus de 2 entités, les règles pour la volumétrie restent les mêmes (diapo suivante).

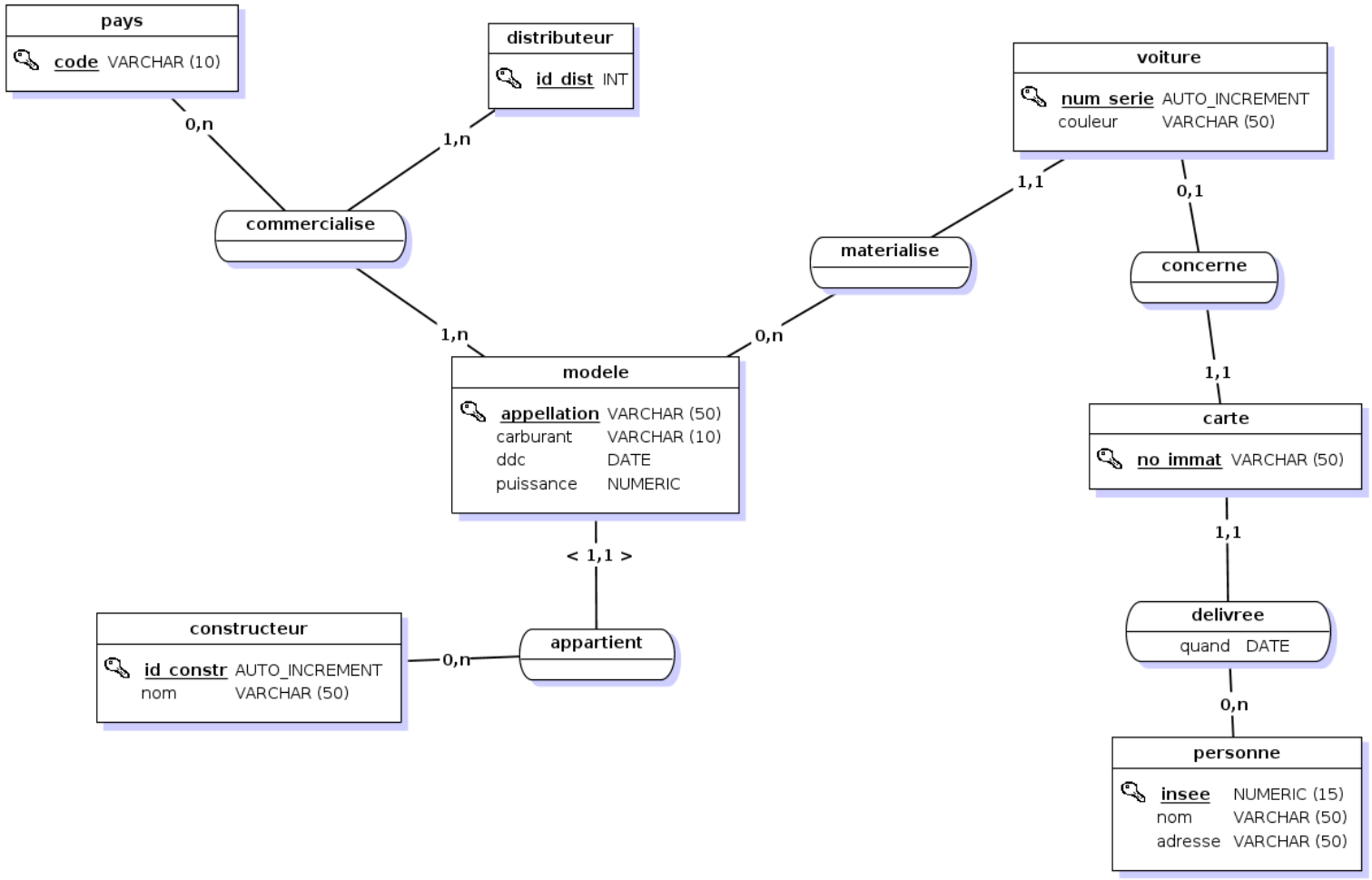
Exemple: imaginons que les modèles de voitures soient distribués par des réseaux de distributeurs internationaux: la même enseigne peut être présente dans différents pays, mais certains modèles de voitures ne seront pas forcément commercialisés dans certains pays. Il existe maintenant une association ***ternaire*** entre 3 entités : modèle de voitures, pays et distributeur.

Associations ternaires

Illustration

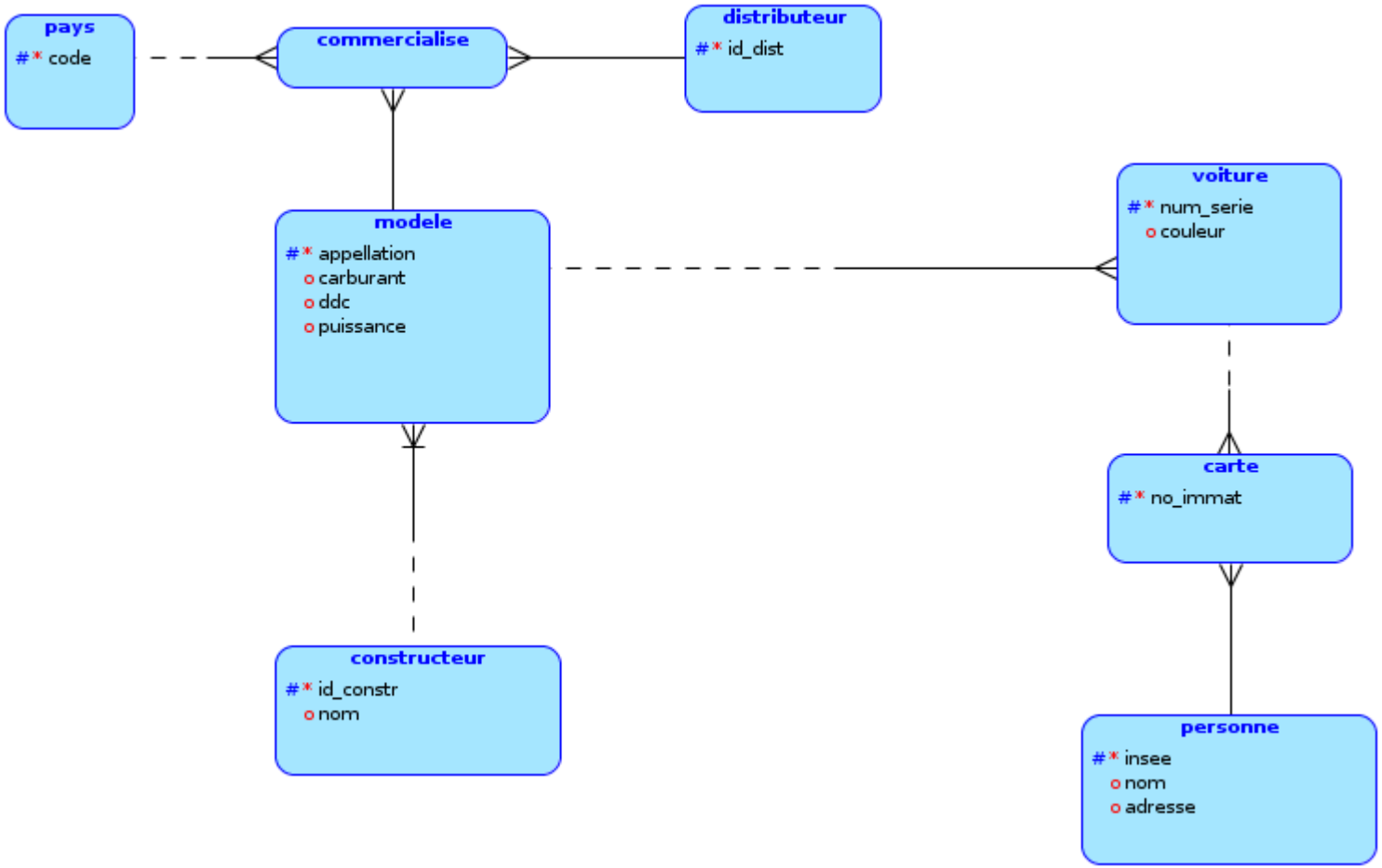


Modèle E-A final (JMerise)



Modèle E-A final (Oracle)

Modèle équivalent (ou presque) sur Oracle Data Modeler

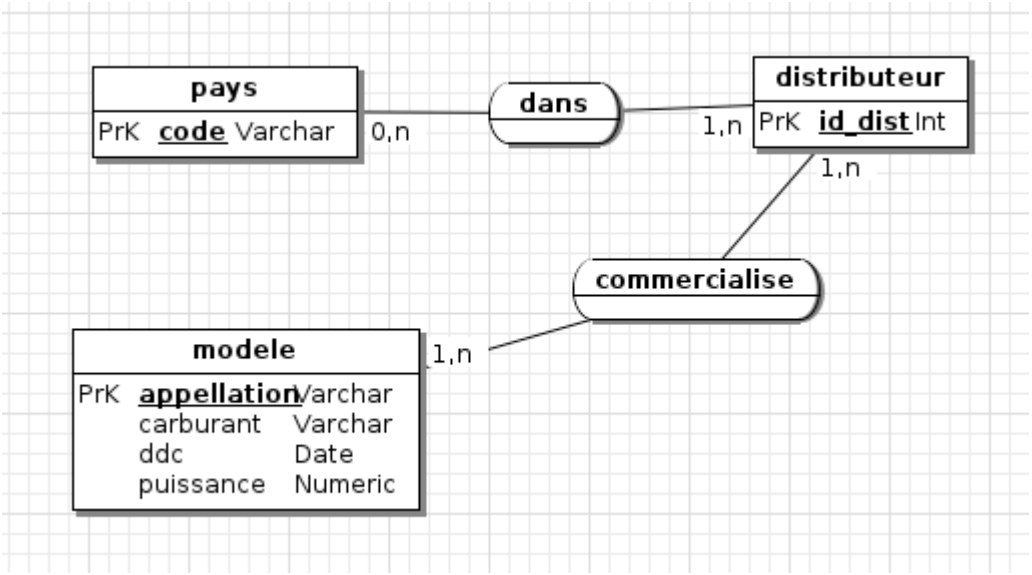


Associations ternaires

Dangers

L'association ternaire (distributeur,modele,pays) aurait été injustifiée et **invalide** dans les cas suivants:

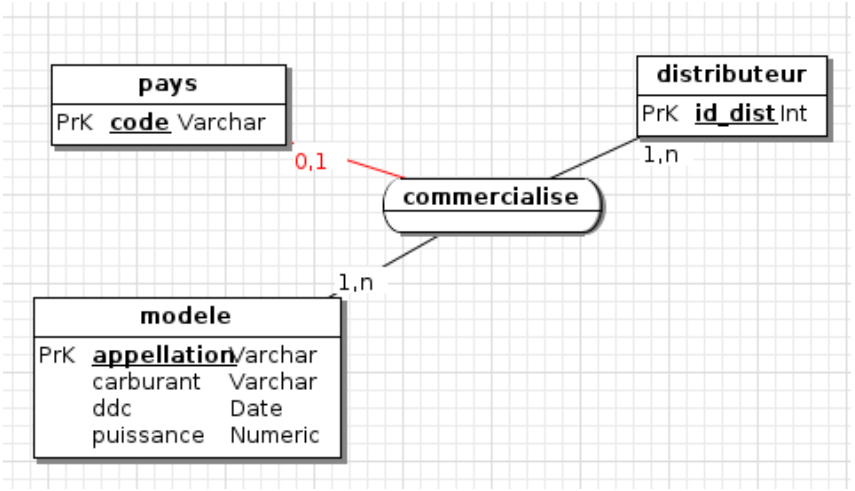
1) Si un distributeur commercialise dans un pays, alors il commercialise tous les modèles. Partie de modèle E-A correcte pour ce cas-là :



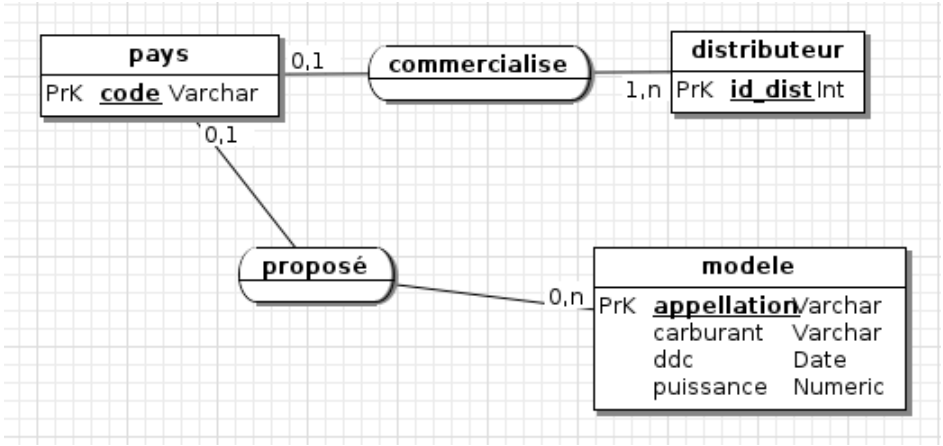
Associations ternaires

Dangers

2) Pour une raison ou une autre, l'une des cardinalités sup est à 1, ex. pour pays:



Modèle faux



Modèle correct

Méthode générale de conception

Les étapes suivantes sont conseillées pour la modélisation d'une solution à problème de BDD :

1. A partir de l'énoncé, identifier les identités physiques (personnes, compagnies d'assurance, voitures, …) ou pas (catalogue de voiture, type de contrat, …) qui ont des propriétés communes et homogènes.
2. Représenter ces identités sous forme d'entités-types. Définir la clé primaire la plus pertinente pour chaque entité.
3. Réfléchir aux associations possibles entre entités (Binaires? Ternaires? …). On a toujours intérêt à avoir :
   - le plus d'associations indépendantes possible,
   - les degrés les plus faibles possibles pour chaque association. Mais aucune hypothèse non spécifiée dans l'énoncé ne doit être prise pour les abaisser. Ex: on ne pas supposer « gratuitement » que le propriétaire d'une voiture est nécessairement son conducteur principal...

Méthode générale de conception

4. Ajouter les associations précédentes, et leurs clé primaire et attributs éventuels.
5. Ajouter la volumétrie.

ATTENTION : on ne modélise JAMAIS les entités et les associations qui:

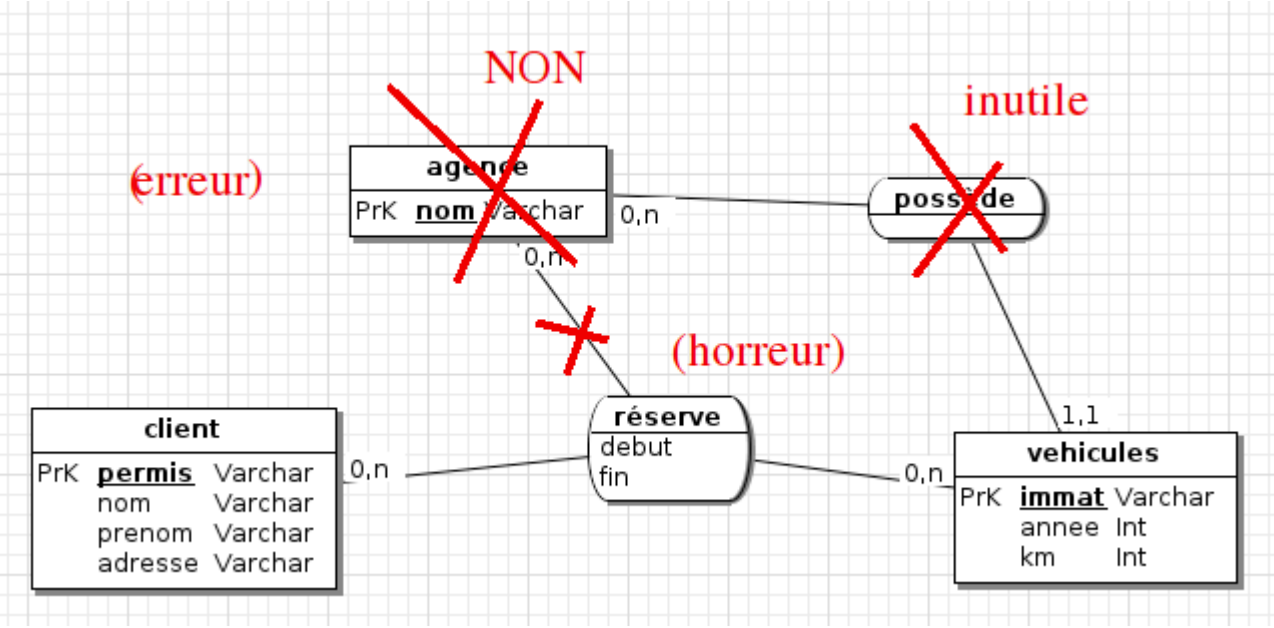
- ne servent à rien : ne contiendront jamais plus d'une instance, ou au contraire toutes les instances possibles
- plus généralement, peuvent se déduire des autres par calcul, ou par l'énoncé => risque élevé de contradiction sinon

ON NE REPRESENTE QUE CE QUE L'ON NE PEUT NI DEDUIRE, NI SAVOIR A L'AVANCE

Méthode générale de conception

Exemples d'erreurs fréquentes.

Ex 1: « Une agence de location de véhicules souhaite proposer ses services en ligne et vous sollicite pour concevoir sa BDD pour gérer ses locations. Les clients réservent à l'agence l'un de ses véhicules d'une date à l'autre. ... »

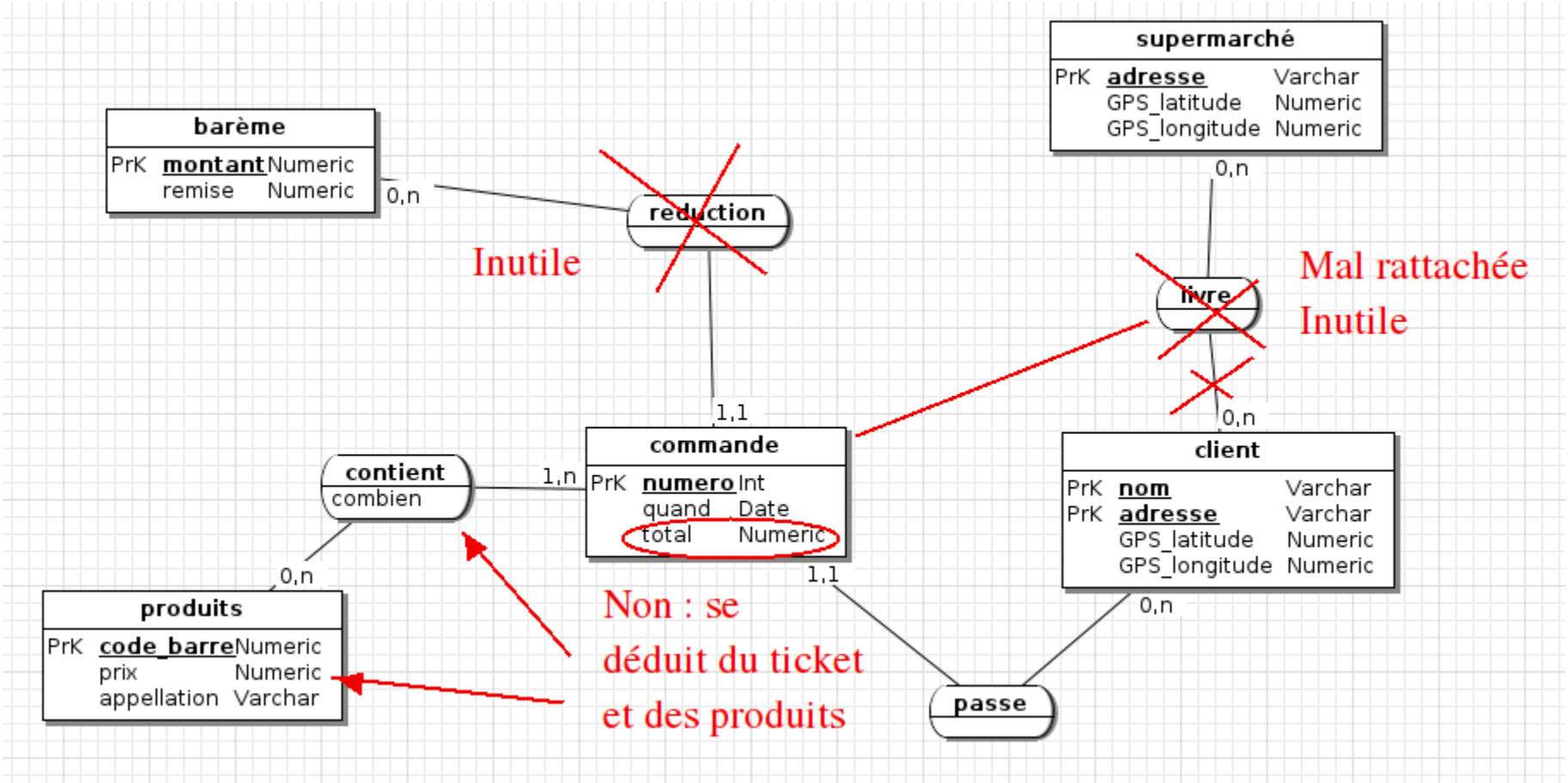


Méthode générale de conception

Ex 2: « eDrive est une chaîne de supermarchés livrant à domicile. Lorsqu'il passe commande, un client choisit les produits qu'il veut se faire livrer, et le supermarché le plus proche le livre. Lorsqu'il règle sa commande, une remise est toujours appliquée à sa commande selon un barème dégressif que eDrive doit pouvoir spécifier (ex: 50€ ⇒ 2%, 100€ ⇒5%, 250€ ⇒ 15%, etc).»

Méthode générale de conception

Ex 2 (suite)



Exercice 1

Modéliser le problème suivant sous forme de diagramme E-A (les composants n'ont pas besoin d'être distingués):

1. Tout PC possède 1 ou 2 processeurs du même modèle, et un nom d'hôte
2. Les processeurs sont soit de type Intel soit de type AMD
3. Tout PC possède une carte vidéo intégrée
4. Tout PC possède au moins 1 et au plus 6 slots PCI
5. Les slots PCI peuvent recevoir des cartes son ou des cartes réseau
6. Un slot PCI ne peut recevoir plus d'une carte
7. Un disque dur est soit de type SATA soit de type IDE
8. Tout PC possède *au plus* un disque dur

Exercice 1

Correction:

Exercice 2

Modéliser (le plus fidèlement possible) le problème suivant sous forme de diagramme E-A :

1. Un département est composé d'une ou plusieurs équipes de recherche
2. Chaque équipe de recherche est rattachée à un seul département
3. Tout membre d'un département ne peut être membre d'un autre département
4. Toute équipe de recherche comporte au moins une personne
5. Un membre du personnel peut diriger une équipe ou un département, mais pas les deux
6. Tout département est dirigé par une seule personne
7. Toute équipe de recherche est dirigée par une seule personne
8. Toute équipe de recherche développe au moins un projet
9. Tout projet est développé par au moins une équipe

Exercice 2

Correction:

Annexe

Autres exemples d'entités faibles