MODELISATION MERISE
MODELES CONCEPTUEL ET LOGIQUE DES DONNEES

Nathalie Harbonne

Processus 10 –BTS CGO –Editions Nathan Technique-Edition 2001

Jean-Patrick Matheron –Comprendre Merise –Outils conceptuels et organisationnels Editions Eyrolles-Date parution et Edition 1994

Jean-Patrick Matheron - Exercices et cas pour comprendre Merise –Date parution et Edition 1994

Plan du cours

I La structure du système d'information de gestion

1-L'information dans l'entreprise :
Que permet l'information dans l'entreprise ?

- La coordination des processus de gestion (comptable, financier etc…)
- de garder une trace des opérations effectuées par les acteurs de l'entreprise
- de prendre des décisions.

2- Présentation du système d'information dans l'entreprise :

Le système d'information de gestion assure le couplage du système opérationnel et du système de pilotage

3-Organisation du système d'information

3.1 Définition du système d'information : Un SI est un ensemble de ressources (personnel, données, procédures, logiciel, matériel) qui permet de stocker, de structurer, de traiter et de communiquer des informations sous forme de textes, images et sons

3.2 Définition du système informatique : le système informatique est un ensemble d'éléments constitué de matériels, de logiciels et d'applications mis en place dans l'entreprise pour la collecte, le stockage, le traitement , le contrôle et la diffusion de l'information

3.3 Définition d'un système d'information intégré de gestion : un système d'information est intégré si une même information n'est saisie qu'une seule fois en un point du système et répercutée dans tous les fichiers concernés crées dans le système.

II Présentation des modèles de données et des traitements

La méthode d'analyse MERISE (Méthode d'Etude et de Réalisation Informatique pour les Systèmes d'Entreprise) a été créée à la fin des années 70, par la volonté des autorités publiques (le Ministère de l'Industrie), désireuses de doter les administrations et les entreprises publiques d'une méthodologie rigoureuse tout en intégrant les aspects nouveaux pour l'époque : informatique répartie, bases de données ...

Elle est citée en France par 70% des entreprises déclarant utiliser une méthode.

La méthode Merise propose une approche de la conception séparant l'étude des données de celle des traitements, en avançant progressivement par niveaux.

Chacun de ces niveaux a pour objectif principal de fournir un certain nombre de documents (MCD, MPD, MCT, ...) permettant ainsi la synthèse textuelle d'un processus de réflexion.

Ces documents sont indispensables à l'élaboration et à la concertation autour de tout projet informatique.

La mise en place des modèles de traitements a non seulement pour but de définir les traitements à effectuer, mais également de valider les options prises lors de l'élaboration des modèles de données.

Ainsi la méthode Merise préconise, non pas d'effectuer l'analyse des données, puis ensuite celle des traitements, mais plutôt de mener en parallèle, à chaque niveau, l'analyse des données et celle des traitements.

La méthode Merise propose trois niveaux de représentation d'un système d'information :

- Le niveau conceptuel.

- Le niveau organisationnel (logique).

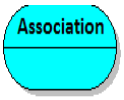
- Le niveau opérationnel (physique).

<u>modèles conceptuels</u> : ils correspondent à la finalité de l'entreprise. Ils définissent les fonctions réalisées dans l'organisation.	Ils répondent aux questions : Que fait–on dans l'organisation ? Avec quelles données ?	**MCD** : modèle conceptuel de données : quelles associations avec quelles entités **MCT** : modèle conceptuel de traitements :quels évènements entrées et résultats(succession des actions)
<u>modèles organisationnels</u> :ils mettent en évidence les choix d'organisation en termes d'automatisation, de postes de travail(utilisateurs) et de chronologie des opérations	Ils répondent aux questions Qui fait quoi ? Où ? A quel endroit Quand ? A quel moment	**MOD** :modèle organisationnel de données : Il présente les différents types d'accès aux données du MCD : droit d'insertion(I),de consultation(C),de modification(M),de suppression(S) Il informe également des différents postes de travail du système d'information Ex : Dans la société(système d'information) Neige et Soleil, Madame Duteille enregistre les écritures comptables. **MOT** :modèle organisationnel de traitements: prend en compte l'acteur concerné et le temps pour les actions décrites
<u>modèles opérationnels</u>	Comment ? Le niveau physique apporte des solutions techniques, ce qui se traduit par le choix et l'utilisation d'un système de gestion de bases de données relationnelles(SGBDR)	**MLD** : modèle logique de données : représentation de l 'ensemble des données du domaine d'étude(MCD) **MPD** :modèle physique des données : implantation de l'ensemble des données du MLD au moyen d'un SGBDR **MPT** : modèle physique de traitements : les différentes fonctionnalités d'un SGBDR(formulaires, états etc…)

III : Le modèle conceptuel de données(MCD) : première approche

1-Définition d'un MCD : le MCD a pour but de représenter un système d'information ou une partie de celui-ci(un sous domaine de gestion), sous la forme d'un schéma qui obéit à un formalisme précis.

Les trois concepts de base d'un MCD sont l'entité(

),l'association (

) et la propriété(ex : nom_client)

Un MCD est également désigné sous le terme schéma entités/associations.

Les entités du MCD deviendront des tables de bases de données.

Le MCD est donc la première étape nécessaire à la conception d'une base de données .

IV Les différentes étapes de la modélisation

1) **analyse de l'existant du SI étudié ou d'une partie de celui-ci(domaine de gestion) : on s'applique à dégager les entités naturelles dites de base** , c'est à dire celles qui apparaissent avec le plus d'évidence.

Qu'est ce qu'une entité?

Définition d'une entité :
Une entité de base est soit un individu (personne physique ou morale) soit un objet matériel (ex : une commande) ou immatériel (ex: un portefeuille d'actions) .

Parmi les objets matériels on compte notamment les objets de gestion(facture, contrats etc…)

Ex : si l'on demande d'analyser le service des commandes clients, les entités qui paraissent naturellement évidentes à créer sont l'entité commande et l'entité client.

On distingue également à côté des entités de base, les entités

- de classification (catégorie , type , famille etc…)

- de caractérisation, qui décrivent des caractéristiques des entités de base
Ex : APPARTEMENT et EQUIPEMENT(entité de caractérisation)

-de temporalité(date, période, semaine, mois, année etc…)

2)**rattacher les propriétés aux entités**

Définition **:** une propriété est une donnée élémentaire qui permet de décrire une entité

Règles à connaître :
-une entité possède au moins une propriété : son identifiant
-une même propriété ne peut pas appartenir à deux entités différentes

Définition : un identifiant est une propriété particulière de l'entité qui contribue à la détermination unique d'une occurence d'une entité type.

explication du terme "occurence": a la création de l'entité CLIENT elle possède les propriétés suivantes :

CLIENT: **code_client**, nom_client, prenom_client, rue_client, numero_rue_client, ville, code_postal

Une occurence de l'entité client est, par exemple:

11001, Durant, Pierre, Chaptal, 79, Levallois,92300

Une autre occurence serait :

12882, Dupond,Paul,Mariton, 21, Clichy,92110

On peut dire que l'entité est le modèle à suivre et une occurence un exemplaire de ce modèle

Par ailleurs, la connaissance **du code_client** permet de connaître son nom, son prénom, son adresse, etc...car le code client est unique.

Cela n'est pas le cas pour le nom du client, car si deux clients ont le même nom , on ne peut , à partir du nom, obtenir les données du client voulu.

On peut dire que le code_client est un identifiant de l'entité Client.

L'entité-type est considérée comme un modèle à suivre .

Une occurrence est un exemple de l'entité-type .

On peut à présent comprendre facilement la définition d'un identifiant.

Un identifiant est une propriété particulière pour chaque entité, qui permet de définir une et une seule occurence de l'entité

3)**Recenser les synonymes et polysémes**

synonyme : écrire la propriété : numéro_de_ commande et la propriété référence_commande, revient à écrire deux noms différents d'une même propriété .

Or on ne peut pas trouver deux fois la même propriété dans un MCD(modèle conceptuel de données); c'est le même principe pour les entités MARCHANDISE et PRODUIT qui désignent la même chose.

Définition : deux noms de propriété ou d'entités synonymes désignent la même chose mais sont différents.
Règle à connaître : On ne peut pas trouver deux propriétés ou entités synonymes dans un MCD.

polysème(=homonyme):si on crée l'entité CAFE : cela représente à la fois une boisson et un établissement; or il ne faut pas qu'il y ait d'ambiguïté au niveau du sens; donc si on veut créer une entité "CAFE" qui représente un établissement , on écrira "ETABLISSEMENT_CAFE" par exemple.

Définition : Un nom de propriété ou d'entité polysème désigne deux ou plusieurs choses différentes.

Règle à connaître : On ne peut pas trouver une propriété ou entité polysème (homonyme) dans un MCD.

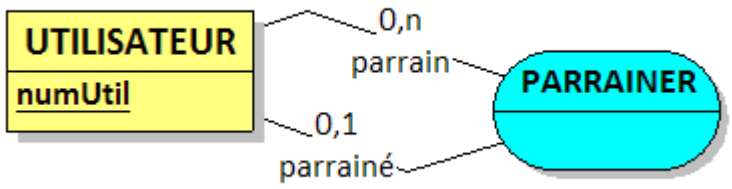
4)**Recenser les différentes associations entre les entités et éventuellement leur rattacher leurs propriétés**.

Une association est nommée par un verbe à l'infinitif qui la qualifie.

Si l'association relie deux entités on dira qu'elle est à deux pattes ou binaire.

Cas particuliers : **si une association relie une même entité à elle-même** , elle est dite unaire ou réflexive car elle lie entre elles des occurrences d'une même entité .

Tout cela sera plus clair après avoir abordé le point 5, qui explique la notion de cardinalité.

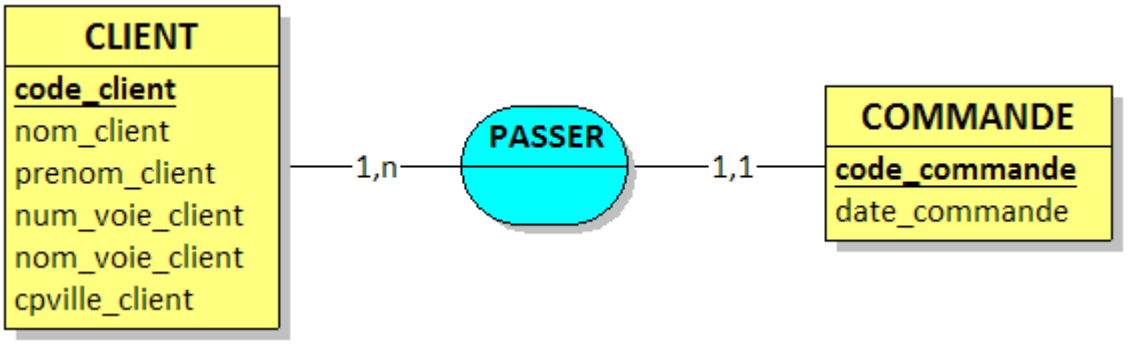


Cas particulier de l'association ternaire : c'est une association qui relie trois entités et dont les cardinalités maximales sont toutes égales à n.

5) **Déterminer les cardinalités de chaque couple Entité-Association**

Définition cardinalité minimum : nombre minimum de fois où une entité est concernée par l'association.
Elle a comme valeurs 0 ou 1 dans la grande majorité de cas.
Elle peut également avoir pour valeur n à condition que la valeur de n soit <= à celle de la cardinalité maximale.

Définition cardinalité maximum : nombre maximum de fois où une entité est concernée par l'association.
Elle peut avoir les valeurs 1 ou n(n signifie plusieurs fois sans préciser de nombre)



On peut voir qu'il y a deux groupes de cardinalités : 1,1 et 1,n
L'association se lit ainsi :
Une commande est passée par un client au minimum et un client au maximum .
En effet, une commande précise ne peut être passée que par un unique client.

L'autre groupe de cardinalités se lit ainsi : un client peut passer au minimum une commande et au maximum un nombre indéfini de commandes(autant qu'il veut).

On ne dit pas qu'un client passe au minimum 0 commande car sinon, il n'est pas un client (cas général).

L'association est binaire car elle relie deux entités.

Cas de l'association réflexive utilisateur vers utilisateur vue plus haut :
la première partie de l'association se lit ainsi : un utilisateur parraine au minimum aucun utilisateur (sens de la cardinalité 0 dans la première partie de l'association) ou parraine au maximum un à plusieurs utilisateurs (sens de la cardinalité n dans la première partie de l'association).

Le rôle associé à cet utilisateur est parrain.

La deuxième partie de l'association se lit ainsi: un utilisateur est parrainé par aucun utilisateur (sens de la cardinalité 0 dans la deuxième partie de l'association) ou au maximum par un seul utilisateur (sens de la cardinalité 1 dans la deuxième partie de l'association).

Le rôle associé à cet utilisateur est parrainé.

5.1 Associations binaires non fonctionnelles

Définition : une association binaire est non fonctionnelle si et ssi ses deux cardinalités maximales sont supérieures à 1.
Une association binaire non fonctionnelle peut porter une ou plusieurs propriétés.

L'identifiant d'une association binaire non fonctionnelle est constitué par le couple formé par chacun des identifiants des deux entités qui participent à l'association .

Exemple :

5.2 Associations binaires fonctionnelles

Définition : Une association binaire fonctionnelle est une association entre deux entités dont au moins l'une des cardinalités maximales vaut 1
Elle n'est jamais porteuse de propriétés .
On l'appelle aussi association fonctionnelle ou dépendance fonctionnelle entre entités

5.2.1 Dépendance fonctionnelle forte entre entités :

C'est une association qui possède le cardinalités 1,1 du premier côté de l'association et 0,1 ou 0,n ou 1,n de l'autre côté de l'association.

Attention : il n'est pas permis de trouver d'association avec les cardinalités 1,1 et 1,1 dans un MCD

5.2.2 Dépendance fonctionnelle faible entre entités :

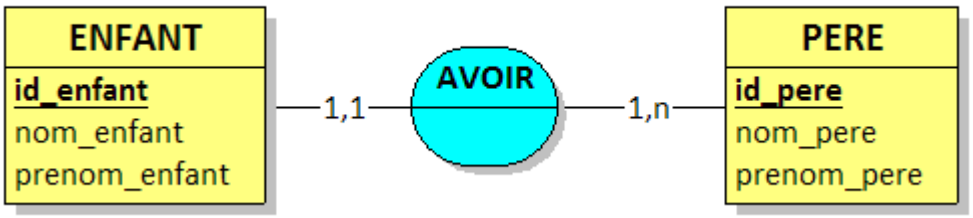
Association qui possède les cardinalités 0,1 du premier côté de l'association et 0,1ou 1,1 ou 0,n ou 1,n de l'autre côté de l'association

Si de l'autre côté de l'association on a les cardinalités 1,1, nous sommes dans le cas d'une DF forte de ce côté.
Si de l'autre côté de l'association on a les cardinalités 0,1, nous sommes dans le cas d'une DF faible de ce côté.

5.2.3 Contrainte d'unicité entre entités :

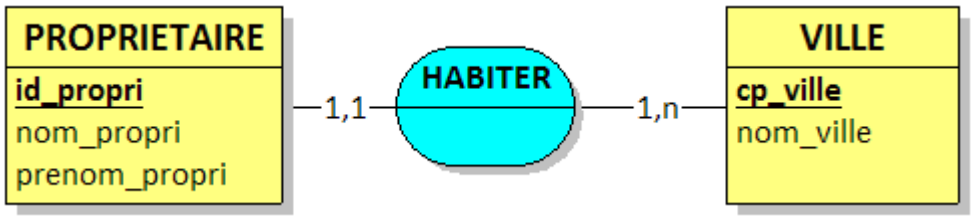
Dès qu'il y a un DF forte ou faible entre entités on parle de **contrainte d'unicité entre entités**

5.2.4 Contrainte d'intégrité fonctionnelle (CIF): elle correspond à la dépendance fonctionnelle **forte et stable**



DF Stable car un enfant ne pourra jamais changer de père ; **on ne pourra jamais mettre à jour le**

lien entre une occurrence de l 'entité enfant et une occurrence de l'entité père



DF instable car un propriétaire peut changer de ville ; on peut mettre à jour le lien entre une occurrence de l'entité propriétaire et une occurrence de l'entité ville.

5.2.5 Entité faible et identifiant relatif

Définition : une entité est dite faible lorsque son existence dépend de celle d'une autre entité.
La destruction d'une occurrence de l'entité forte entraîne la destruction des occurrences de l'entité faible associée.

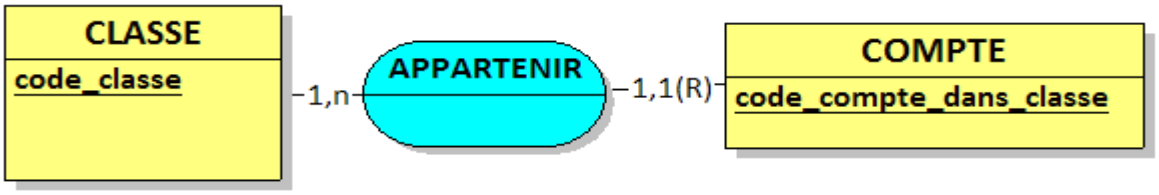
Ex : en comptabilité, les comptes appartiennent à des classes
Le compte 403 est vu comme le compte 03 de la classe 4.

Il ne peut exister que si la classe 4 existe.

La destruction de la classe 4(un exemple de l'entité forte) entraîne celle de tous les comptes de classe 4(toutes les occurrences de l'entité faible associée).
L'entité faible a alors un identifiant relatif à celui de l'entité forte.

C'est un cas particulier de CIF

(1,1) ou



Plusieurs notations existent pour exprimer la dépendance d'une entité :
- soit on met des **parenthèses** au niveau des cardinalités(1,1)
- soit on écrit la lettre (R) (comme « relatif ») à coté des cardinalités 1,1

6) **On vérifie le modèle pour s'assurer qu'il répond bien aux règles de construction**

7)**Création du dictionnaire de données qui recense :**

a)le nom des propriétés de chaque entité

b)leur définition (ce que ce nom de propriété signifie)

c) l'entité à laquelle elles appartiennent

d) leur type(numérique, alphanumérique, caractère)

e) la longueur du libellé de la propriété(ex : 5 qui signifie que c'est un nombre sur 5 caractères)

f) si elles ont valeur d'identifiant ou pas(donc oui ou non)
Ex

NOM_PROPRIETE	DEFINITION	ENTITE	TYPE	LONGUEUR	IDENTIFIANT
code_Client	code du client	CLIENT	varchar	5	0(pour oui)

On trouve dans certains dictionnaires de données une autre colonne COMMENTAIRES dans laquelle on indique le format, par exemple pour les dates.

Ex : pour la propriété date_commande , on précise en commentaires jj/mm/aaaa,
ou pour la propriété ref_commande on précise en commentaires AA-9999 : 2 caractères pour l'année et 4 pour le n° d'ordre.

8)**Normaliser le modèle : s'assurer qu'il est au moins en 3ème Forme Normale**

V Définition des formes normales

1Définition première forme normale

Un MCD est en première forme normale si toutes ses propriétés sont non décomposables(liste de valeurs d'un même type, autres propriétés), c'est à dire si elles sont **atomiques.**

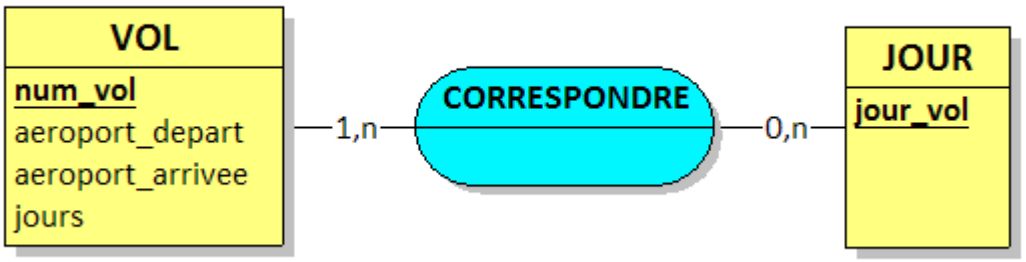
Un MCD n'est pas en 1 FN si une de ses propriétés peut prendre une liste de valeurs d'un même type. On crée alors une nouvelle entité pour représenter la multiplicité des valeurs de cette propriété.
Ex :

*précision :à partir du moment où un vol a le même aéroport de départ et aéroport arrivée, il peut avoir lieu plusieurs jours dans la même semaine

La propriété « jours » peut donc prendre plusieurs valeurs : lundi, mardi, mercredi, jeudi, vendredi,

samedi , dimanche, alors que l'identifiant permet (selon la règle de l'identifiant connue) de connaître une et une seule valeur de la propriété « jour » par occurrence concernée(1 occurrence = 1 exemplaire de l'entité « VOL »).

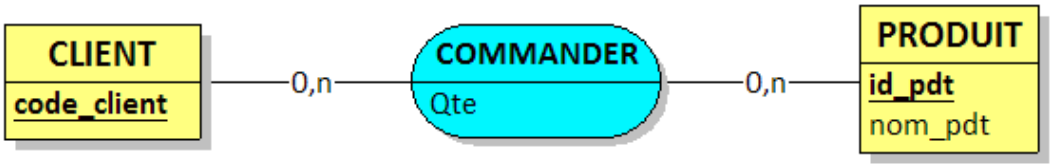
Pour corriger cela on crée une nouvelle entité et une nouvelle association, qui sont celles-ci :



Cette association se lit ainsi :

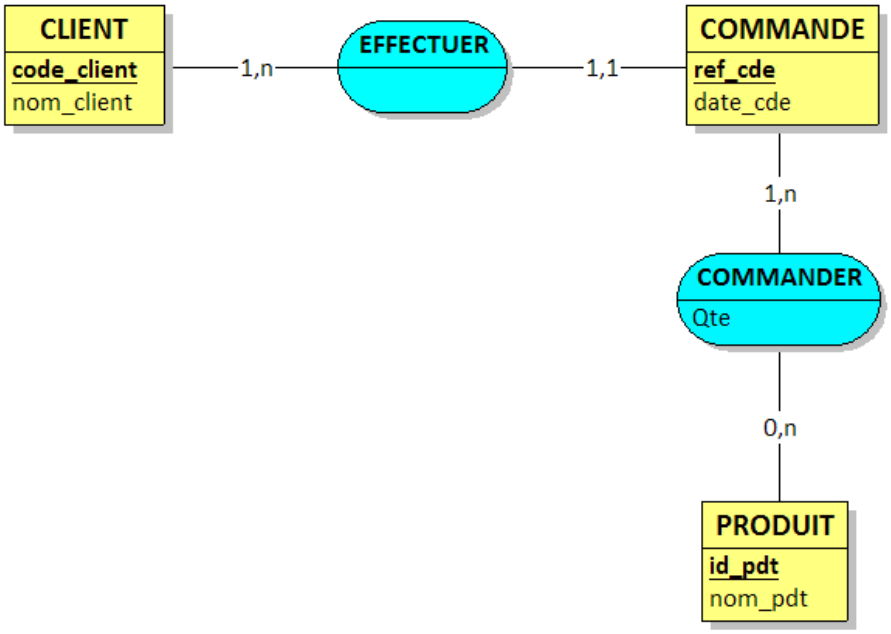
Un vol correspond au minimum à une journée de vol et au maximum à plusieurs journées de vol et à un jour de vol correspond aucun ou plusieurs vols.

Cas d'une propriété d'une association qui prend une liste de valeurs



Dans le cas ci-dessus, il existe plusieurs valeurs pour la propriété Qte, car un client peut commander plusieurs quantités d'un même produit et Qte est en fait une somme de plusieurs quantités Qte1+Qte2+Qte3 etc…

On ajoute donc une entité COMMANDE qui permet de ne donner qu'une seule valeur possible pour une Qte donnée(Qte n'est plus l'addition de Qte1+Qte2+Qte3 etc…).



Dernier cas où une entité n'est pas en première forme normale est celui où une de ses propriétés est composée de plusieurs propriétés.
La propriété « adresse », par exemple, est composée des propriétés suivantes : num_ rue, nom_rue, code_postal,ville.

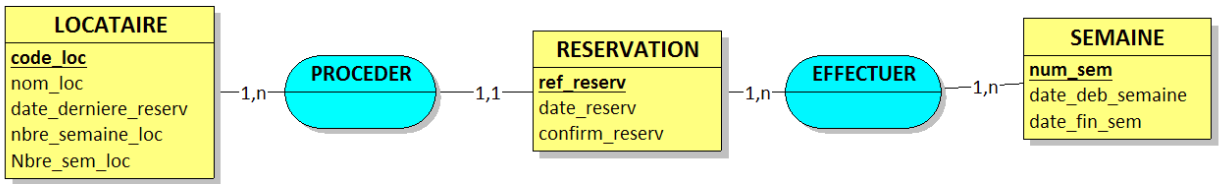
On ne peut donc pas laisser la propriété adresse ainsi .Il est nécessaire, à la place, d' indiquer toutes les propriétés qui la composent.

Cas des propriétés calculées : une propriété calculée est obtenue par calcul de propriétés existantes(comme nous l'avons vu plus haut dans le cas où Qte= Qte1+Qte2+Qte3.)
Elle n'est donc jamais saisie dans un MCD.

Cependant , sa présence se justifie exceptionnellement dans deux cas :

1-on ne conserve pas dans le temps les opérandes

Exemple du locataire dont on cumule le nombre de semaines en location et pour lequel on ne garde pas trace des réservations d'une année sur l'autre(les occurrences de l'entité **RESERVATION** et de l'association **EFFECTUER** ne sont pas conservées d'une année sur l'autre).



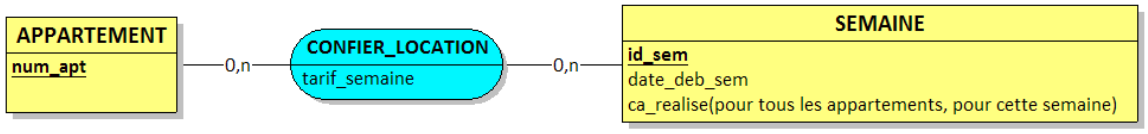
Chaque occurrence de l'association EFFECTUER correspond à une semaine de réservation pour une réservation donnée ; chaque réservation correspond à un locataire précis.

Pour respecter la règle de gestion qui dit qu'un locataire qui a loué pour 3 semaines bénéficie d'une prime (=> confirm _reserv=OUI, avec la date de dernière réservation sur les 3 dernières années), il faut cumuler le nombre de réservations confirmées dans une propriété de type numérique « nombre de semaines louées »et conserver la date de dernière réservation dans une propriété de l'entité locataire.

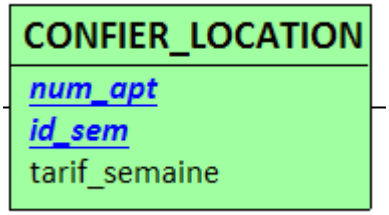
La propriété « nombre de semaines louées » sera incrémentée à chaque saisie d'une occurrence dans l'association EFFECTUER (attention : pas dans l'entite RESERVATION car cela ne permet pas de connaître le nombre de semaines réservees et à condition que la propriété confirmation= OUI)

2-dans un souci d'optimisation

Exemple d'un chef d'entreprise qui souhaite suivre l'évolution de son CA réalisé semaine après semaine
On acceptera la présence de la propriété « ca_realise » dans l'entité SEMAINE.
Elle est en fait une somme de tous les CA réalisés par appartement et par semaine.



Voici ce qu'on trouvera dans la future table CONFIER LOCATION traduction de l'association CONFIER_LOCATION



CONFIER LOCATION

num_apt,	id_sem	tarif_sem
apt1	sem1	200 euros
apt2	sem1	450 euros
apt1	sem2	500 euros
apt2	sem2	350 euros

Le CA réalisé pour la semaine sem1 est de 200+450= 650 euros= valeur calculée, normalement interdite

Le CA réalisé pour la semaine sem2 est de 500+350=

850 euros= valeur calculée, normalement interdite

1.1-Définition d'une dépendance fonctionnelle :

Nous avons vu qu'un identifiant identifie de manière unique une occurence d'une entité(qu'on appelle entité-type pour souligner la notion de modèle comme je l'ai expliqué précédemment)

Cela signifie qu'un identifiant détermine toutes les autres valeurs des propriétés d'une occurence, Par exemple, si on reprend l'entité COMMANDE qui possède la propriété date_commande

Si une commande possède l'identifiant N165, la connaissance de cet identifiant N165 entraîne celle de la date de commande.

On dira qu'il existe une dépendance fonctionnelle entre l'identifiant de l'entité COMMANDE et la propriété date_commande car la connaissance de la valeur de l'identifiant d'une occurence de l'entité COMMANDE entraîne celle de la propriété « date_commande » de cette occurrence

Définition d'une dépendance fonctionnelle :

Un propriété P2 dépend fonctionnellement d'une propriété P1, si la connaissance de la valeur de P1 détermine une et une seule valeur de P2.
Cette dépendance est notée P1->P2 et se lit P2 dépend fonctionnellement de P1.

Un identifiant détermine par dépendance fonctionnelle toutes les valeurs des autres propriétés d'une entité .

A présent , supposons que l'on ait ce cas de figure(voir dessin ci-dessous)
entité : CLIENT
identifiant : code_client,code_categorie, ensemble de propriétés P1 (l'identifiant est composé de deux propriétés dans ce cas précis).

S'il suffit de connaître le code d'un client pour connaître la valeur de la propriété nom_client correspondante, cela ne convient pas.

Il est nécessaire que les deux propriétés qui constituent l'identifiant déterminent les autres propriétés de l'entité.

On dit qu'il faut que la dépendance fonctionnelle P1->P2 soit élémentaire.

Le problème de l'élémentarité d'une dépendance fonctionnelle ne se pose donc que quand l'identifiant est composé de deux ou plusieurs propriétés

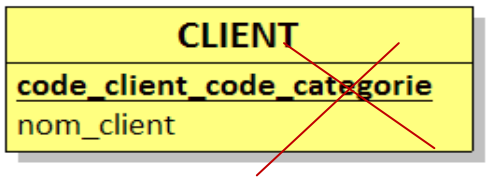
1.2 Définition d'une dépendance fonctionnelle élémentaire : une dépendance fonctionnelle P1->P2 est dite « élémentaire » si aucune partie de l'ensemble des propriétés P1 ne détermine la propriété P2(il ne faut pas qu'il suffise qu'une partie de l'identifiant détermine une valeur de propriété non identifiant).
Il est nécessaire que toutes les parties de l'identifiant déterminent les valeurs de chaque propriété non identifiante d'une entité.

2-Définition deuxième forme normale

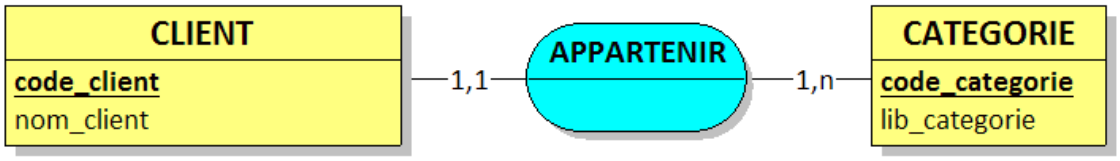
Un MCD est en deuxième forme normale si elle est en première FN (1 FN) et si toutes les dépendances fonctionnelles entre l'identifiant et les autres propriétés sont élémentaires.

Cas d'une entité qui ne respecte pas la 2 FN(2ème forme normale)

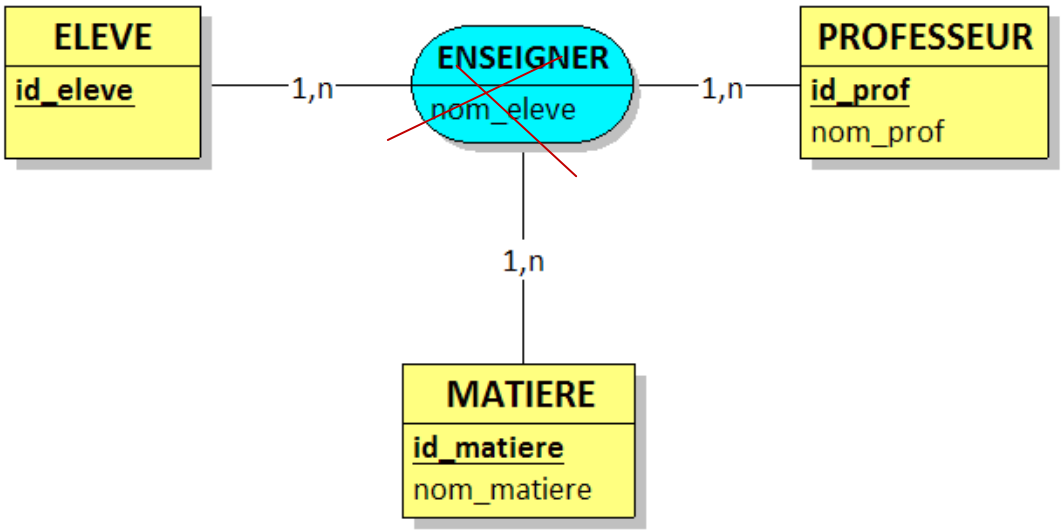
Dans le cas ci-dessous, la 2 FN n'est pas respectée car code_client, qui n'est qu'une partie de l'identifiant, suffit à déterminer une valeur de la propriété nom_client ;



Pour rectifier cette situation on procède ainsi



Cas d'une association qui ne respecte pas la 2 FN



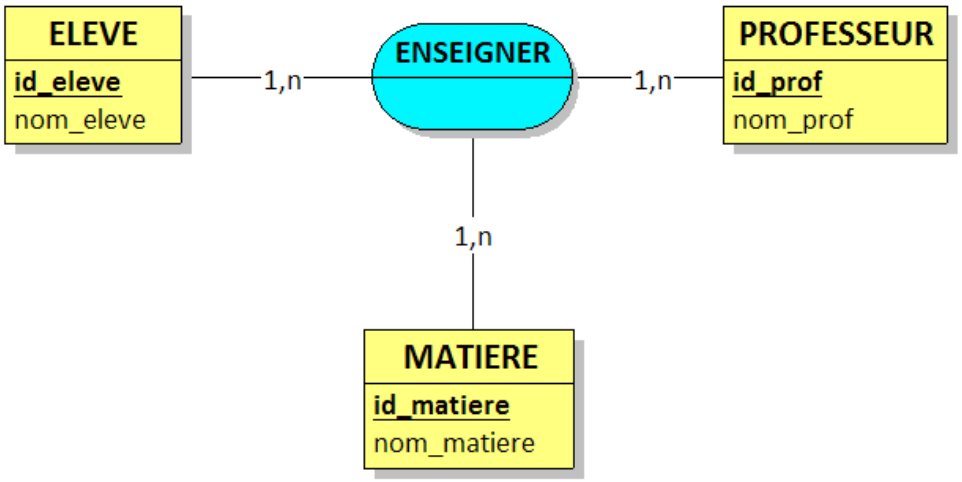
1-La propriété nom_eleve de l'association ENSEIGNER n'est pas bien placée.

En effet, dans cette association qui est non fonctionnelle(cardinalités max =n des 2 côtes de l'association), l'identifiant de l'association ENSEIGNER est composé des trois propriétés identifiant id_eleve,id_professeur id_matière des entités reliées par l'association ENSEIGNER.

Il faut donc que les trois propriétés id_eleve,id_professeur et id_matière entraînent la connaissance de la propriété nom_eleve,

Or il suffit de connaître id_eleve pour connaître nom_eleve ; il n'y a donc pas une vraie DF entre les 3 identifiants id_eleve, id_professeur , id_matière et la propriété nom_elève.

Pour rectifier cela, on fait passer la propriété nom_eleve dans l'entité ELEVE.
Ainsi il existe bien une DF entre id_eleve et nom_eleve(voir dessin ci-dessous)



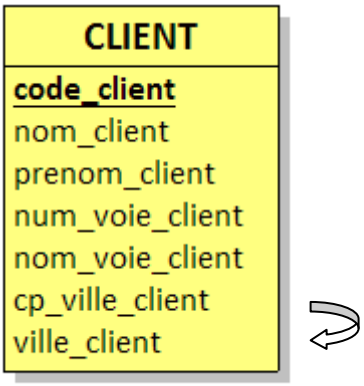
A présent, pour expliquer la notion de dépendance fonctionnelle directe reprenons le cas de la règle mathématique de transitivité.

Si on reprend l'exemple de l'entité CLIENT, son identifiant est : code_client

Supposons que ses autres propriétés soient : nom_client, num_voie, nom_voie, cp_ville, ville_client

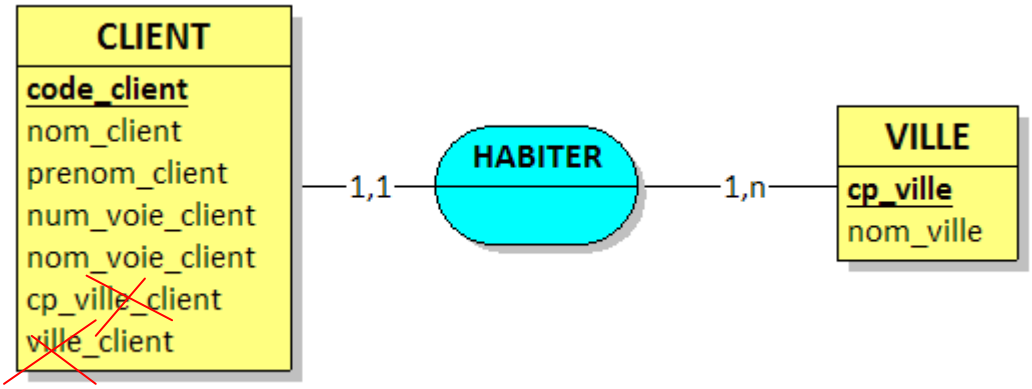
La connaissance de code_client entraîne celle de ville_client et la connaissance de cp_ville entraîne également celle de ville_client, ce qui signifie que l'on peut connaître la valeur de ville_client , pas uniquement grâce à l'identifiant code_client mais également grâce à la propriété cp_ville

Or ce cas de figure ne doit pas exister .
Il faut que seul l'identifiant puisse permettre de connaître la valeur d'une propriété, autrement dit :



DF non permise entre cp_ville_client et ville_client

Pour corriger cela on procède ainsi



Définition d'une dépendance fonctionnelle directe : Une dépendance fonctionnelle est directe si chaque propriété non-identifiant ne dépend fonctionnellement **que de l'identifiant**

3-Définition troisième forme normale

Un MCD est en 3 FN si il est en 2 FN et si toutes les dépendances fonctionnelles entre l'identifiant et les autres propriétés sont élémentaires et directes : chaque propriété non identifiant ne dépend fonctionnellement que de l'identifiant .Seul l'identifiant est source de dépendance fonctionnelle.

Lorqu'on demande à ce qu'un modèle conceptuel soit normalisé, on demande qu'il soit en 3 FN .

4 Définition forme normale Boyce Codd

Il est très rare qu'un MCD ne soit pas en Forme normale Boyce Codd; c'est la raison pour laquelle on exige qu'un MCD soit en 3 FN mais pas davantage.
Il est cependant nécessaire de connaître la définition de cette forme normale :

Un MCD est en forme normale de Boyce Codd si
-il est en 3 FN et si aucune propriété faisant partie de l'identifiant ne dépend fonctionnellement d'une propriété non identifiant.

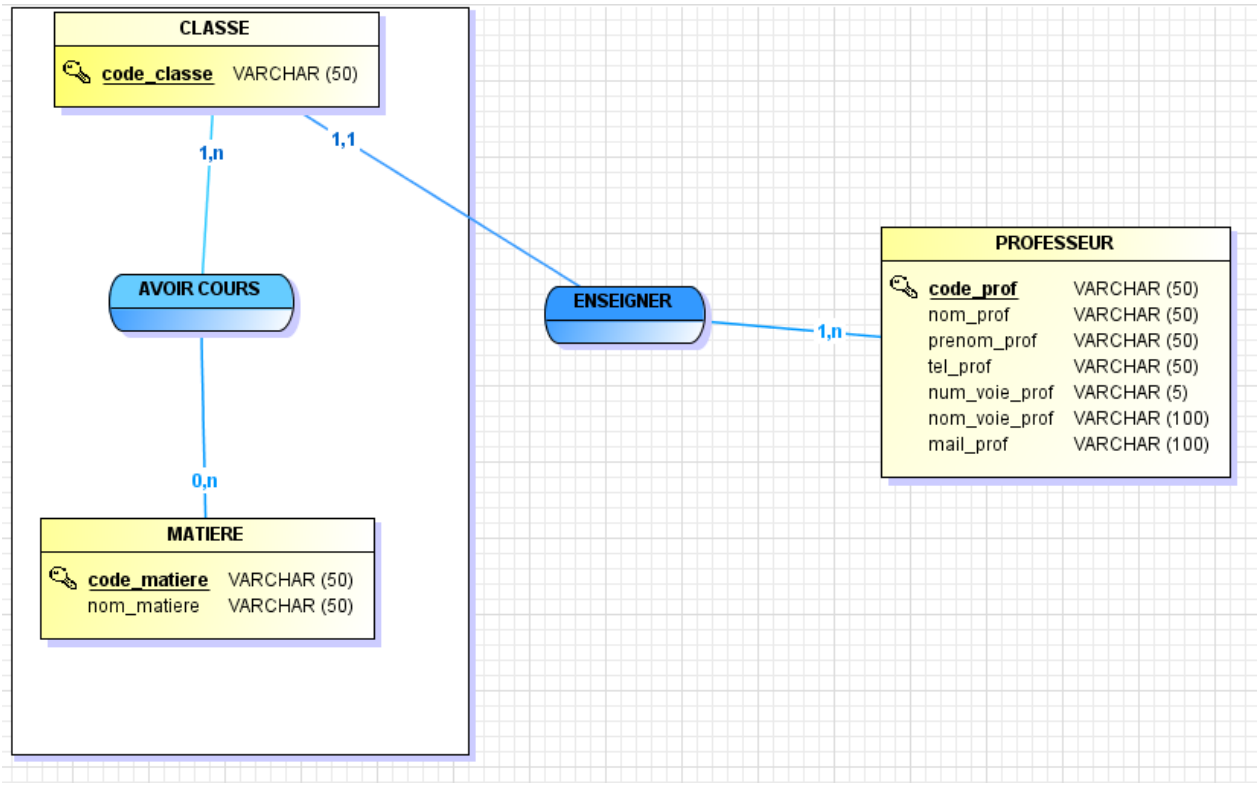
Or ce cas de figure se produit uniquement **lorsque l'identifiant est composé**, c'est à dire que l'on a par exemple ceci :

il existe une dépendance fonctionnelle interdite entre code_prof et code_matiere

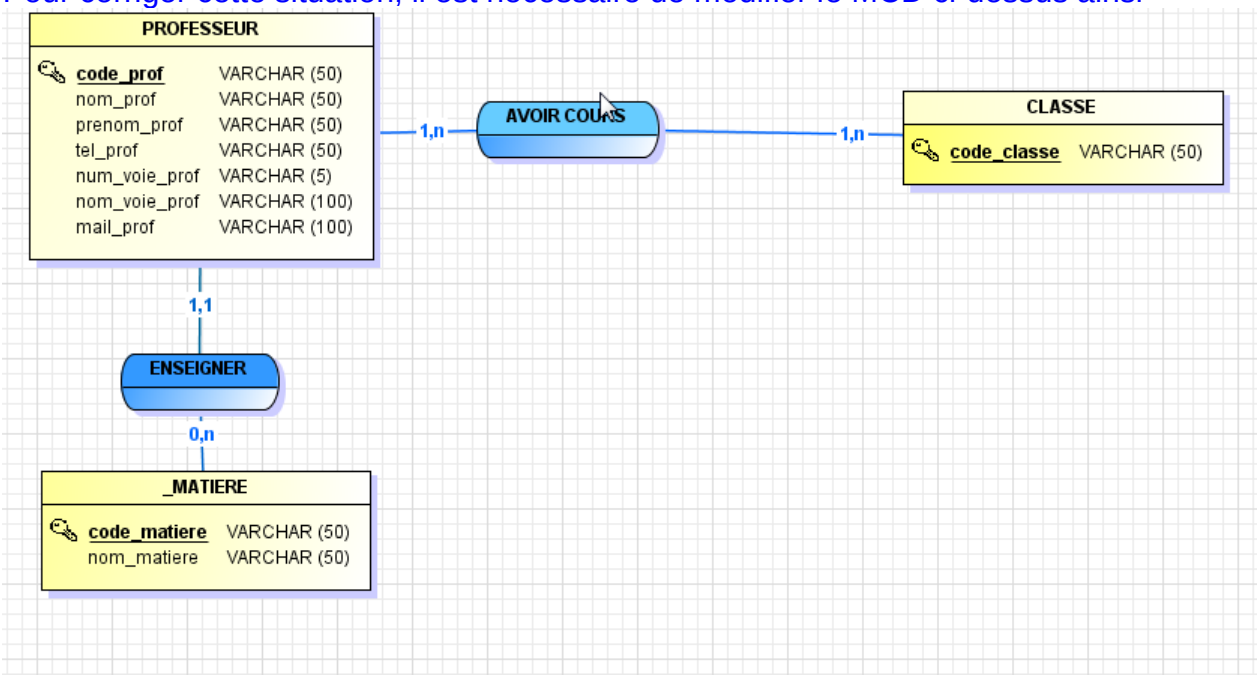
Règles de gestion : tout professeur enseigne une matière et une seule.
Toute classe n'a qu'un seul professeur par matière.

Cette règle de gestion est traduite dans un premier temps par le MCD ci-dessous(il existe les cardinalites 1,1 entre l'association AVOIR COURS considérée comme pseudo-entité et l'association ENSEIGNER)**.**

Il se trouve que la partie « code_matiere » de l'identifiant dépend fonctionnellement de la propriété « code_prof, c'est à dire que l'on peut connaître la valeur de la propriété « code_matiere » si on connaît la valeur de la propriété « code_prof », dans une occurrence de l'association « AVOIR COURS ».
Ceci n'est pas permis dans le cadre de la forme normale de Boyce Codd.



Pour corriger cette situation, il est nécessaire de modifier le MCD ci-dessus ainsi



Voici la première partie du cours à connaître pour pouvoir réaliser un MCD(modèle conceptuel des données) correctement.

VI La généralisation/spécialisation

1-définition du principe de généralisation/spécialisation

Rappel : les occurrences d'entités se caractérisent toutes par un ensemble de propriétés communes
Il peut arriver que certaines occurrences d'entités se distinguent :

-en utilisant une ou plusieurs propriétés inutiles aux autres occurrences

-par certaines associations spécifiques à ces occurrences

Pour prendre en compte ces caractéristiques spécifiques à ces occurrences d'entités on décompose cette entité :

-en plaçant les propriétés communes dans une entité générale dite « générique »

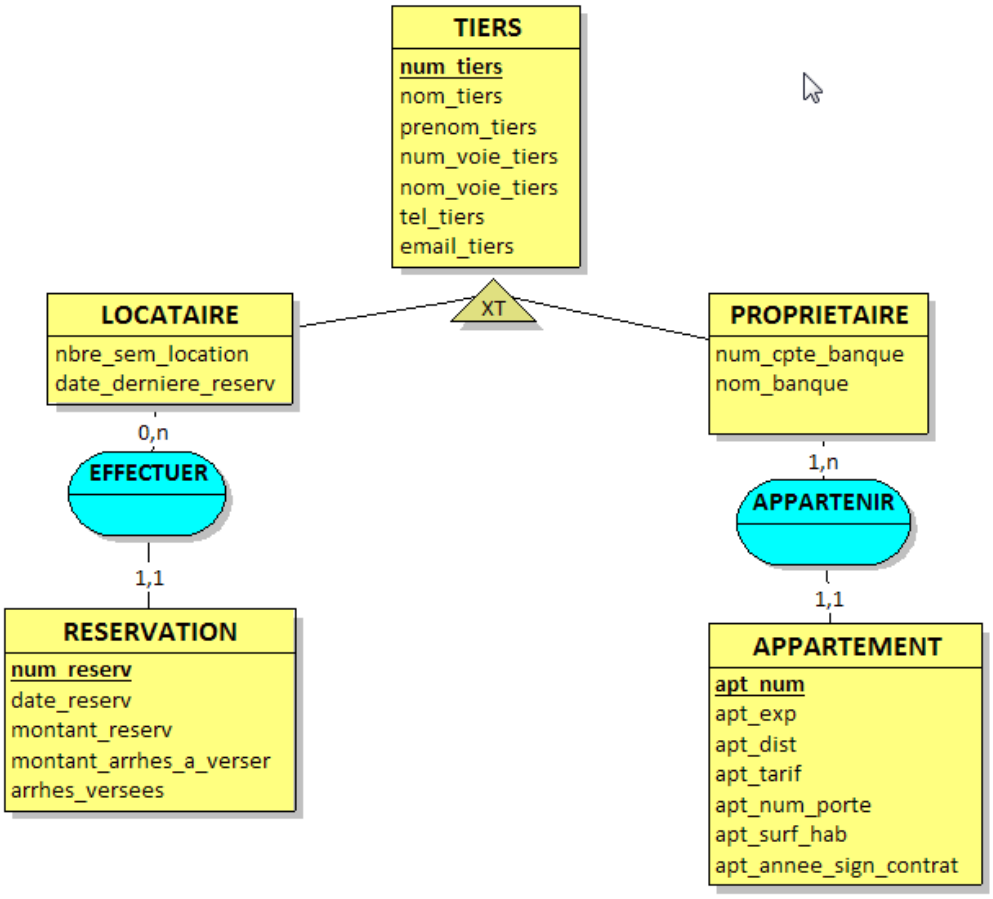
-en plaçant les propriétés spécifiques aux sous-ensemble d'occurrence dans des entités spécifiques

-en reliant par une DF forte les entités spécialisées à l'entité générique

La hiérarchie de généralisation/spécialisation est également appelée « hiérarchie de sur-type/sous – type » .

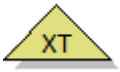
Chaque entité concernée généralisation/spécialisation peut s'associer avec toute entité du modèle(dans notre exemple, les entités locataire et propriétaire s'associent respectivement avec les entités appartement et réservation ce qui n'est pas le cas de l'entité TIERS).

Exemple (voir explications en dessous du dessin)



L'entité TIERS est une entité **générique**.
Les entités PROPRIETAIRE et LOCATAIRE sont **spécifiques** de l'entité génerique TIERS : elles héritent de toutes les propriétés de l'entité TIERS.

L'association APPARTENIR ne concerne que l'entité PROPRIETAIRE(et pas l'entité TIERS).
De même l'association EFFECTUER ne concerne que l'entité LOCATAIRE(et pas l'entité TIERS)

Le signe 

est le symbole de l'héritage : il indique une dépendance fonctionnelle FORTE entre l'entité génerique et les entités spécifiques.

2-Les contraintes sur généralisation /specialisation (dites encore sur sur-type/sous-type)

1-contrainte de partition

2-contrainte de totalité

3-contrainte d'exclusion

4-absence de contraintes

Pour chacune de ces contraintes, voir le document scanné correspondant

3- Les contraintes sur associations

Les contraintes entre associations

Pour chaque contrainte représentée , on définit
-le type de contrainte

-l'entité concernée par la contrainte , appelée **le pivot** de la contrainte

-les deux associations liées par la contrainte

1-contrainte de partition

2-contrainte de totalité

3-contrainte d'exclusion

4-contrainte d'inclusion

5-contrainte de simultaneité

VII Le modèle logique des données

1-Définition : le modèle logique de données est une image du modèle conceptuel de données

intégrant les choix d'organisation des fichiers (fichiers classiques, modèle hiérarchique ou en réseau, ou modèle relationnel).

Nous sommes à l'époque où l'utilisation des SGBDR(systèmes de gestion de bases de données relationnelles) est généralisée.

C'est donc le modèle logique relationnel des données appelé encore modèle logique de données , adapté à ces systèmes, qui sera utilisé.

modèle logique relationnel des données : les données sont enregistrées dans des tableaux à deux dimensions(lignes et colonnes).

La manipulation de ces données se fait d'après la théorie mathématique des relations.

Le modèle logique relationnel (MLR ou MLD modèle logique de données ou encore SRD : schéma relationnel de données) est un modèle **directement exploitable** par la base de données que vous voulez utiliser.

Définition d'une relation qui sera reprise lors de la règle de passage du MCD au MLD selon laquelle toute entité devient une relation. .

Qu'est ce donc qu'une relation en mathématique ?

Définitions

- **Domaine**

  Un domaine est un ensemble de valeurs.

  Exemple :

  L'ensemble □ = {0,1,2, … , □ } est le domaine des entiers.

- **Produit cartésien**

  Le produit cartésien d'un ensemble de domaines D_1, D_2, …, D_n est l'ensemble des tuples {v_1, v_2, …,v_n} tels que □ i vi □ D_i .

  On le note : D_1□ D_2□ …□ D_n

  Exemple :

  D_1={Pierre, Paul, Jacques}

  D_2={1,2}

  D_{1x} D_2 =

Pierre	1
Pierre	2
Paul	1
Paul	2

Jacques	1
Jacques	2

- **Relation**

  Une relation est un sous ensemble nommé(défini) du produit cartésien d'une liste de domaines.

  Exemple :

  D_1={Pierre, Paul, Jacques}

  D_2={1,2,3, …, 20}

  On peut composer la relation " Note " :

Pierre	8
Paul	13
Jacques	6

  Une relation peut être vue comme un tableau à 2 dimensions dont les colonnes correspondent aux domaines et les lignes contiennent les tuples(ou enregistrements d'une table).

  Une relation est parfois appelée " table ".

- **Attribut**

  Un attribut est une colonne d'une relation caractérisée par un nom.

  Exemple :

  Nom ; Valeur.

- **Schéma de relation**

  Un schéma de relation est le nom de la relation suivi de la liste des attributs avec leurs domaines.

  Exemple :

  Voiture (Marque = <Renault / Peugeot / … > , Couleur = <Bleu / Vert / … >,

  Puissance :< 10 / 20 / … >)

  Afin de simplifier, on ne précise en général pas les domaines.

- **Base de données relationnelle**

  Une base de données est relationnelle lorsque son schéma est un ensemble de schémas de relations dont les occurrences sont des tuples(enregistrements) de ces relations.

2-Règles de passage du modèle conceptuel de données ou (MCD) au modèle logique de

données ou MLD

1-Chaque entité devient une **relation** (**ou table** de base de données au niveau physique).

L'identifiant de l'entité devient **clé primaire** de la relation ou table .

Chaque propriété devient un **attribut** de relation ou de table.

2-Associations de type 1 : n (maximales 1 et n)

Chaque table possèdera sa propre clé mais la clé de la table située côté 1,n ou 0,n dans le MCD, migre vers la table située côté 0,1 ou 1,1 dans le MCD et devient **clé étrangère de cette table**.

Si une association est à trois pattes et si une des branches a une cardinalité 1,1, on fait migrer les clés primaires des autres tables dans la table située du côté 1,1

3-Associations de type 1 :1 c'est à dire de cardinalités maximales 1 et 1

-si l'association est de cardinalités 0,1 et 1,1, on considère l'association comme une association de type 1:n, **en ramenant la clé primaire de la table côté 0,1 vers la table située côté 1,1.**

Celle-ci devient alors également **clé étrangère de la relation côté 1,1**

-si l'association est de cardinalités 0,1 et 0,1: on fait migrer la clé primaire d'une des tables (celle pour laquelle on aura le plus de valeurs renseignées) vers l'autre table.

Cette clé primaire devient alors **clé étrangère.**

4- Associations de type n : n :

Une table intermédiaire dite table de jointure doit être créée .Elle doit posséder comme clé primaire l'ensemble des clés primaires des tables pour lesquelles elle sert de jointure.

5- Attributs des associations

Attention : les associations de type 1 :1 et 1:n ne doivent pas être porteuses de propriétés (c'est le signe d'une mauvaise modélisation).

-Quand le type d'association est de type n :n, les attributs de l'association deviennent des attributs de la table de jointure.