



LA METHODE MERISE :

Le Modèle Conceptuel de Données

Mme D.BOULAHROUZ



Le modèle conceptuel de données

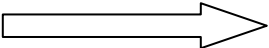
MCD

Le modèle conceptuel des données (**MCD**) décrit la signification des données sur lesquelles reposent les systèmes d'information et les structures.

Le **MCD** est l'élément le plus connu de MERISE et certainement le plus utile. Il permet d'établir une représentation claire des données du S.I. et définit les dépendances fonctionnelles de ces données entre elles.

I/ Constitution du Dictionnaire de données

Le dictionnaire de donnée est un outil nécessaire pour la construction du MCD. C'est la première étape de l'élaboration du MCD

MCC  **Etude détaillée** de chaque document

Un document = Ensemble de **rubriques**

Une rubrique = Ensemble de **données élémentaires**

Donnée élémentaire = Donnée non décomposable

Le dictionnaire de donnée est un tableau qui regroupe toutes les données du SI, pour chaque donnée identifiée il faut préciser :

- Un nom : (n_client, n_facture, nomClient, ...)
- Son type (numérique (entier, réel, date) ou alphanumérique)
- Sa longueur
- Son origine (quel document(s) ?)
- Sa nature : donnée calculée (**C**) ou non calculée (**NC**)
 - ✓ ex: passer une commande : prix (NC) * quantité (NC) = prixCommande(C)
- Des observations :
 - ✓ Des contraintes d'intégrité : exemples : salaire > smic, age > 0
 - ✓ Des règles de calcul : exemple: nouvelle valeur du salaire > ancienne valeur

Exemple : Soient les deux documents à étudier *Commande* et *Facture*.

N° client : 1250
Soumam Computer
Rue de la liberté Béjaia

Bon Commande N°120
12/12/2006

Ref	Désignation	Quantité
P010	Ecran	05
P020	Souris	20
P025	Clavier	10

Code commande : 120
Date facture : 15/01/2007
Matricule client : 1250

Facture N°30

Ref	Désignation	Qté	PU	Prix
P10	Ecran	05	12500	62500
P20	Souris	20	200	4000
P25	Clavier	10	300	3000

Montant total HT : 69500

Montant TVA : 11815

Montant TTC : 81315

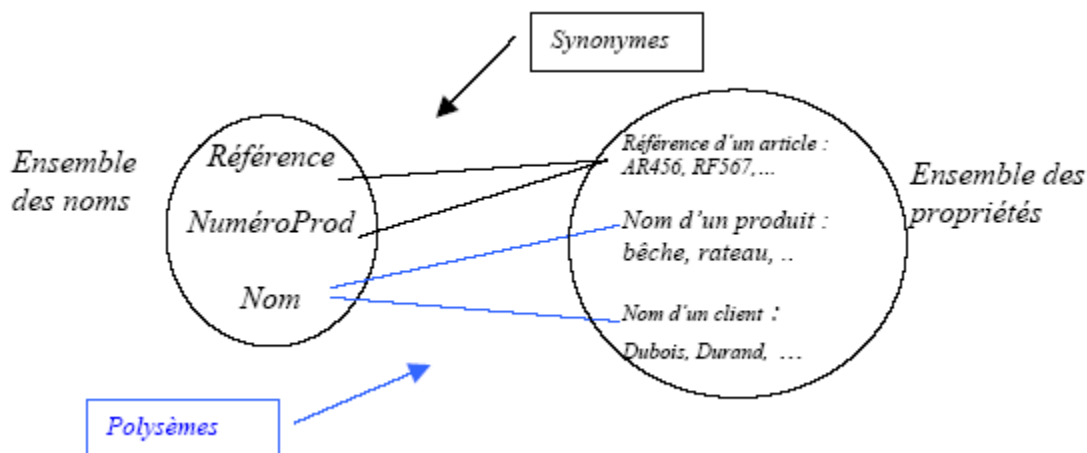
Dictionnaire de données global (brut)

Code de la donnée	Désignation	Type	Taille	Observation
Num-C	Numéro de la commande	N		JJ/MM/AAAA
Date-C	Date de la commande	Date	10	
Num-CL	Numéro client	N		
Nom-CL	Nom client	A	15	
Adr-CL	Adresse client	CC	40	
Ref	Référence produit	AN	4	
Désignation	Désignation produit	A	20	
PU	Prix unitaire produit	N		
Quantité	Quantité produit	N		
Num-F	Numéro facture	N		
Date-F	Date facture	D	10	JJ/MM/AAAA
Code-C	Code commande	N		
Mat-C	Matricule client	N		
Ref	Référence produit	AN	4	
Desgn	Désignation produit	A	20	
PU-P	Prix unitaire produit	N		
Quantité	Quantité produit	N		
Montant	Montant produit	C		
Mont-T_HT	Montant total	C		
Mont_TVA	Montant TVA	C		
Mont_TTC	Montant TTC	C		montant = PU-P * Qte-fact Mont-Total = $\sum$ Mont-P Mont_TVA = Mont_T_HT*Taux_TVA Mont_TTC = Mont_T_HT+Mont_TVA

Epuration du Dictionnaire de données brut

Certaines données doivent être éliminées de ce dictionnaire.

- Les synonymes** : noms différents pour une même valeur (Num-C et Code-C ; Num-CL et Mat-CL)
- Les **polysèmes** : deux données sont polysèmes si elles ont le même nom et expriment deux valeurs différentes (le même nom « Quantité » pour deux données différentes : Qté_cdé et Qté_fact). Il faut les renommer.



- Les redondances** : chaque propriété identifiée, n'apparaît qu'une seule fois dans le modèle.
- Supprimer les **données calculées** ;
Remarque : certaines informations calculées nécessitent d'être conservées, on parle d'informations calculées et mémorisées (**CM**) ; exemple : N°client automatique.

- e. Ajouter les informations détectées par les règles de calcul : Introduction de tauxTVA
- f. Décomposer les données concaténées : (Adresse : Rue, ville et code postal)

Dictionnaire de données épuré

Code de la donnée	Désignation	Type	Taille	Observation
Num-C				
Date-C				
Nom-CL				
Rue-CL				
Ville-CL				
Réf-Prod				
Des-Prod				
PU-P				
Num-F				
Date-F				
Num-C				
Qte-Cde				
Qte-fact				
Taux_TVA				

II/ Elaboration du modèle conceptuelle de données

Le **MCD** est un modèle chargé de représenter sous forme graphique les informations manipulées par le système d'information ; il décrit les données gérées et les liens qui existent entre elles sans tenir compte des choix d'organisation, d'automatisation, ou techniques ; il exprime le QUOI sur les données. Le modèle ne doit comporter que des informations utiles.

Le formalisme adopté par la méthode Merise pour réaliser cette description est basé sur le modèle «**Entité/Association**».

II.1/ Les concepts de base du modèle Entité/Association : Le modèle Entité/Association manipule essentiellement trois concepts : propriétés, entités et Associations

II.1.1/ La propriété (ou attribut ou rubrique) : C'est est une information élémentaire, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié. Elles sont regroupées dans le DD épuré.

* Chaque valeur prise par une propriété est appelée **occurrence**.

Exemple : Des occurrences de la Propriété « désignation article » sont: « Ecran », « souris », ...

II.1.2/ L'Entité ou individu-type :

- a. **Définition** : Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire. Elle peut représenter une notion concrète (CLIENT) ou une notion abstraite. Chaque entité est composée de propriétés élémentaires permettant de la décrire :

Exemple: l'entité **ENSEIGNANT** possède : un nom, un prénom, un âge, un nombre d'enfants, une situation familiale,...

L'entité **ARTICLE** regroupe les propriétés : Référence, Désignation et PrixUnitaireHT

Représentation graphique



Pour une université, on aura.

- Les enseignants
- les étudiants
- Les cours
- Les salles
- Les examens,...

Application 1

Le vidéo club envisage de gérer la location des cassettes à l'aide d'un SGBD. On vous communique les quelques éléments d'information suivants :

« Les cassettes sont louées aux clients. Elles sont achetées auprès d'éditeurs. Les films sont joués par des acteurs. »

Activité : Recherchez les différentes entités

- CASSETTE
- CLIENT
- EDEITEUR
- ACTEUR

Une entité peut souvent être décrite sous plusieurs appellations : dans l'exemple cassette et film représentent deux fois la même entité ; il faut donc ici ne retenir que l'entité cassette ou film (mais pas les deux)

- b. **Les identifiants** : Un identifiant est un ensemble de propriétés (une ou plusieurs) permettant de désigner une et une seule entité. La définition originale est la suivante: ***L'identifiant est une propriété particulière d'un objet telle qu'il n'existe pas deux occurrences de cet objet pour lesquelles cette propriété pourrait prendre une même valeur*** (exemple d'identifiant : Numéro de sécurité sociale, Numéro d'immatriculation de voiture, code article,...)

Une entité est complètement définie par :

- Un nom
- Un identifiant
- Une liste de propriétés

- c. **Occurrence d'entité ou individu** : D'après la définition d'une entité, on sait que la connaissance d'une valeur de la rubrique identifiante détermine la connaissance des valeurs des autres rubriques de l'entité. L'ensemble de ces valeurs est appelé occurrence d'entité.

Exemple : Dans une petite entreprise, le client est défini par les propriétés suivantes :

- N° du compte , Nom, Solde compte

« 411279 -SARL MIRAMAR - 15421 F » constitue une occurrence de l'entité client

- d. **Règles de vérification** :

- Une entité a un seul identifiant
- Une entité a au moins une propriété
- Une entité participe a au moins une association
- A chaque occurrence de l'entité, il ne peut y avoir au plus qu'une valeur de la propriété: (Si une personne possède plusieurs numéros de téléphone, il faudra éclater ces numéros sous plusieurs titres)
- Une information ne peut être que dans une seule entité. Pour être dans cette entité, elle doit dépendre de l'identifiant (notion **de dépendance fonctionnelle**)

Application 2

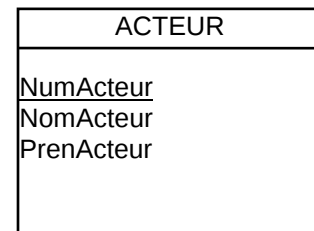
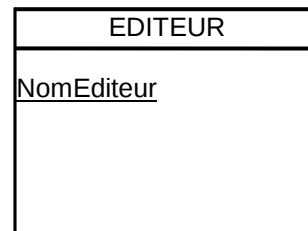
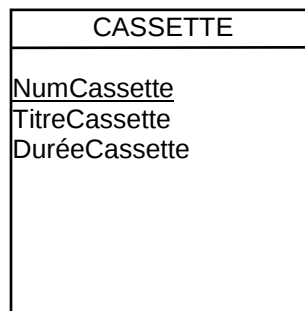
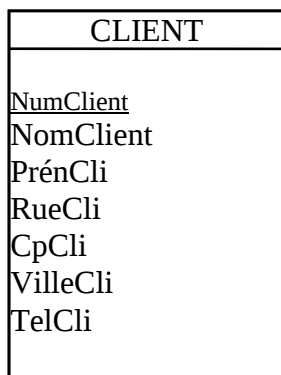
On vous communique les informations suivantes relatives à la gestion des locations de cassettes du vidéo club :

Fiche client N° : 712 NOUT Marion
 125, rue des Albères
 59500 DOUAI
 Tél : 03-27-97-50-50

Exemple de **Fiche cassette**
 N° cassette : 1745
 Titre du film : « Obsession informatique »
 105 mn
 Editeur : Warner Home Vidéo
 Acteurs principaux :
 - Harrison Ford
 - Cindy Crawford
 - John Hackerman
 Editeur : René Bateau Vidéo

T.A.F. :

Représenter les différentes entités et leurs propriétés



*Pour Acteur, on aurait pu retenir le nom+le prénom comme identifiant
 L'entité EDITEUR ne comporte qu'une seule propriété, si une cassette n'est
 Publiée que par un seul éditeur, dans ce cas on aurait pu ajouter
 La propriété NomEditeur dans l'entité CASSETTE et supprimer l'entité éditeur*

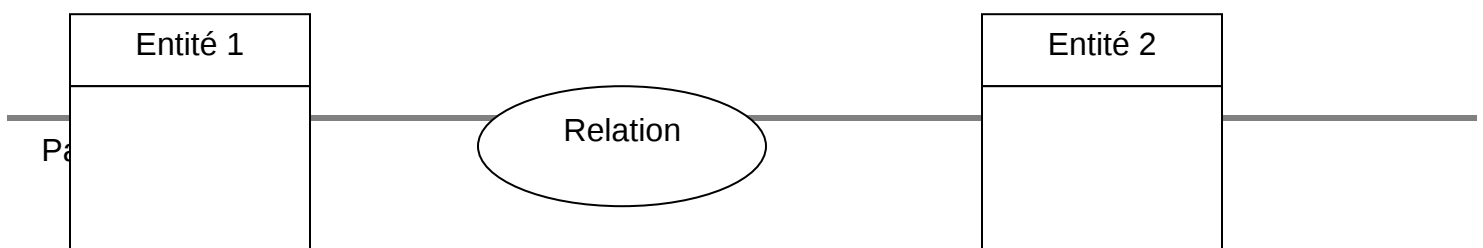
II.1.3/ L'Association (ou Relation) :

a) **Définition** : Une **association** est une relation, un lien logique (sémantique) entre deux ou plusieurs entités. L'association est représentée par un verbe.

Exemple

- Mariage est un lien entre deux personnes
- Le travail est un lien entre l'employé et son employeur
- Le client passe une commande

Représentation graphique :

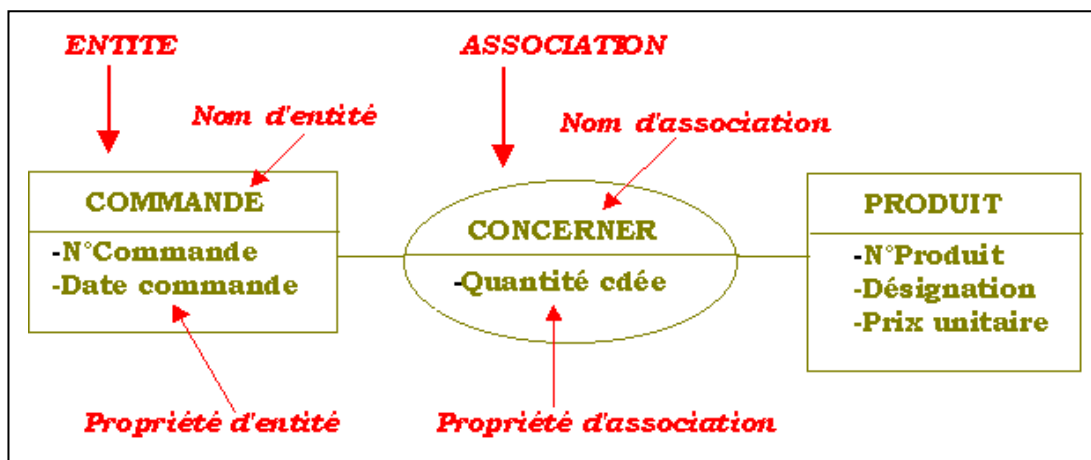
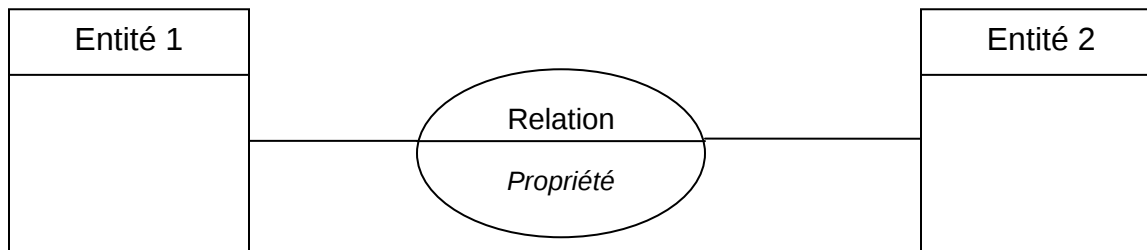


On distingue différents types d'association:

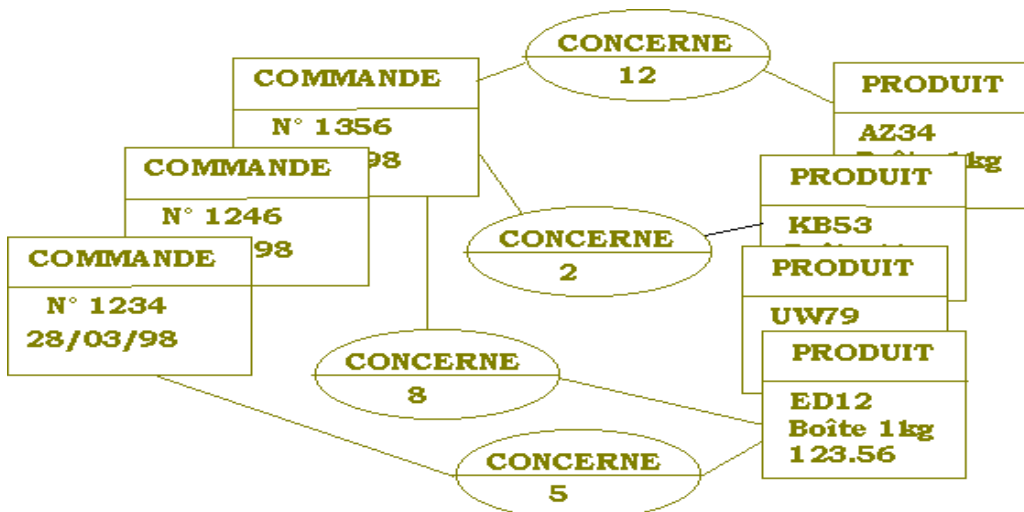
- les **associations binaires**: qui associent 2 entités
- les **associations ternaires**: qui associent 3 entités
- les **associations n-aires**: qui associe plus de 2 entités (ex: associations ternaires, quaternaires)
- les **associations réflexives** qui associent les occurrences d'une même entité

Le choix d'une association est lié à l'intérêt du concepteur.

- Une association peut être **porteuse de données (les propriétés)**



b) **Occurrences d'association** : Une occurrence d'association est un lien particulier qui relie deux occurrences d'entités.



La commande n°1234 concerne un seul produit : 5 boîtes de ED12 au prix unitaire de 123.56

La commande n°1356 concerne 3 produits : AZ34, KB53 et ED12

La commande n°1246 ne concerne aucun produit.

Le produit AZ34 est demandé par une commande.

Le produit KB53 également.

Le produit UW79 n'est pas demandé. Cela ne paraît pas être une anomalie.

Le produit ED12 est demandé dans 2 commandes.

c) Cardinalités :

La cardinalité caractérise la participation d'une entité à une association.

Elle représente le nombre d'occurrences de l'association pour chaque occurrence de l'entité On distingue:

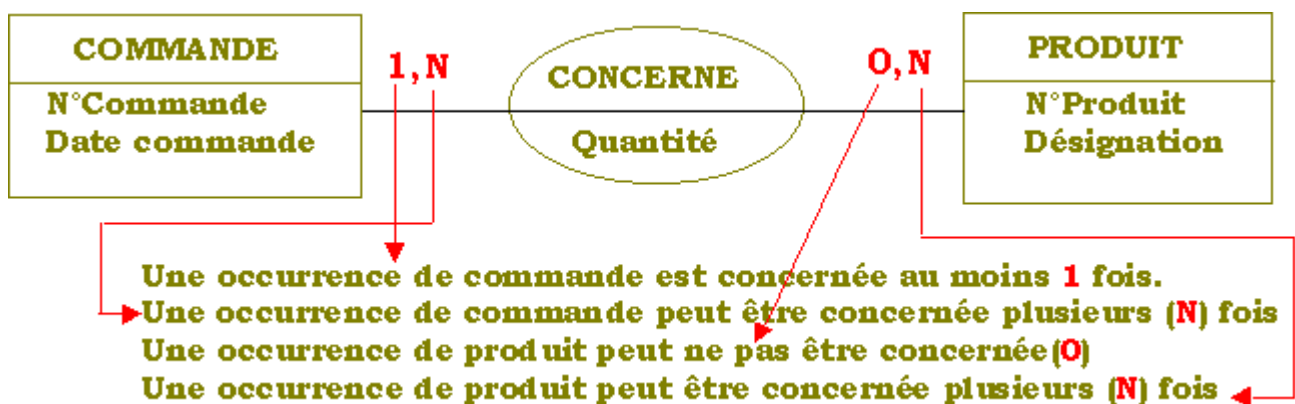
- **la cardinalité minimale**: (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation
- **la cardinalité maximale**: (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation.

Un couple de cardinalités placé entre une entité E et une association A représente le nombre minimal et maximal d'occurrences de l'association A qui peuvent être « ancrées » à une occurrence de l'association E.

Le tableau ci-après récapitule les valeurs que peut prendre ce couple :

0,1	Une occurrence participe au moins 0 fois et au plus 1 fois à l'association
1,1	Une occurrence participe exactement 1 fois à l'association
0,N	Une occurrence peut ne pas participer ou participer plusieurs fois
1,N	Une occurrence participe au moins 1 fois, voire plusieurs

Exemple :



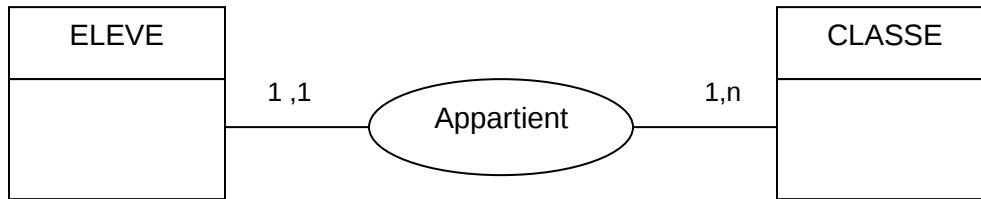
d) Les règles de gestion

Les règles de gestion déterminent des contraintes que doit respecter le Modèle. Les règles de gestion peuvent porter sur deux éléments :

- ✓ des contraintes au niveau des données, des liens entre les données

Ex : RG1 : Un élève appartient à une seule classe

RG2 : Une classe comporte au maximum 35 élèves



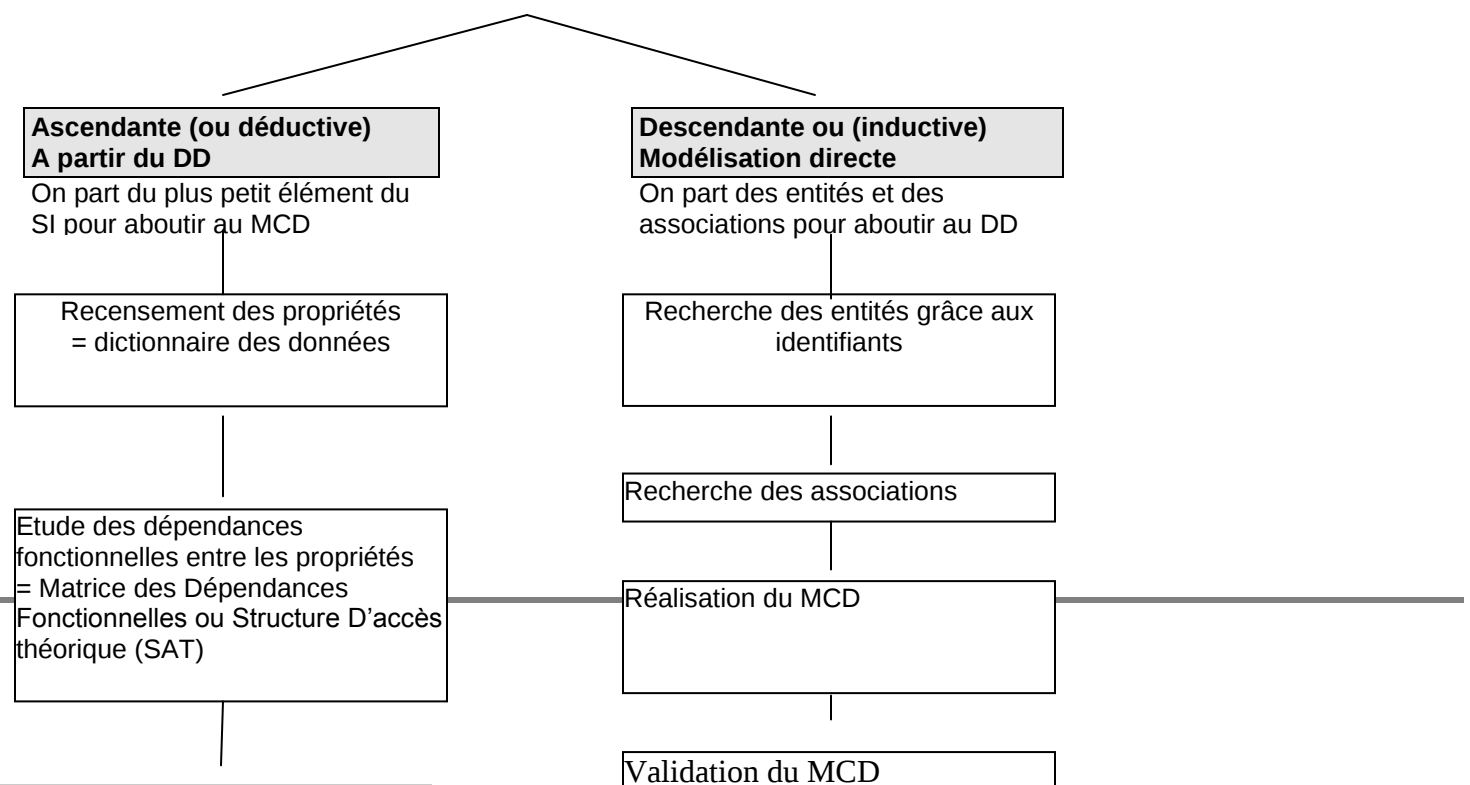
- ✓ des règles de calcul

La recherche et la formulation des règles de gestion est une des étapes les plus difficiles. La modélisation est d'autant plus facile si les règles de gestion sont détaillées.

III/Règles de construction du modèle :

Pour établir un MCD, il existe deux méthodes possibles :

Méthodes



La méthode ascendante est la méthode qui semble la plus logique, toutefois elle est plus difficile à mettre en œuvre que la méthode descendante qui nécessite simplement de connaître les structures types rencontrées dans les MCD

III.1/ La Modélisation directe :

Elle consiste à identifier, à partir d'une description exprimée en langage naturel, les entités et les associations en appliquant les règles suivantes :

- les noms deviennent des entités
- les verbes deviennent des associations

L'exemple suivant qui illustre ce propos est bien trop simple pour que cette méthode conduise à des résultats satisfaisants sur un système d'information de taille plus importante.

Une voiture appartient à un modèle particulier. Les noms sont : « voiture », « modèle ». Le verbe est : « appartient à » Ce qui donne la modélisation 1,n 1,1

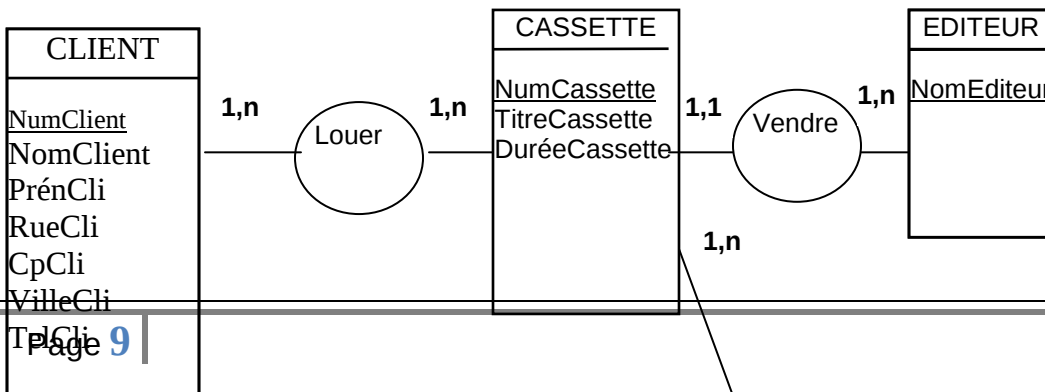
Le modèle obtenu par cette méthode est très loin de la représentation optimale et il sera nécessaire d'appliquer une phase de validation et de normalisation (élimination des situations qui induisent des redondances) pour aboutir à une solution satisfaisante.

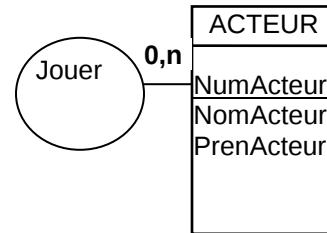


Application 4 : VIDEO CLUB (SUITE)

Reprenez l'application 2 et complétez le Modèle en y reportant les associations et en indiquant les cardinalités. Règles de gestion :

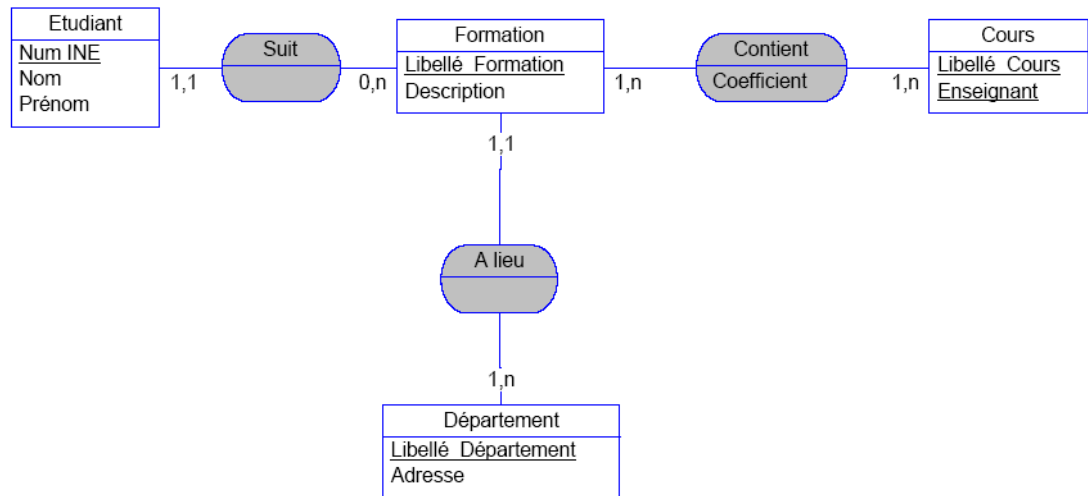
- Un film n'est commercialisé que par un seul éditeur.
- Un film comporte au moins un acteur
- Dans la base on souhaite conserver tous les acteurs même s'ils n'ont tourné dans aucun des films disponibles en cassettes au vidéoclub





Exercice d'application : Gestion des Formations.

Un département caractérisé par un libellé et une adresse dispense plusieurs formations. Un étudiant ne peut s'inscrire qu'à une formation à la fois. Une formation se compose de plusieurs cours décrits par un Libellé et le nom de l'enseignant responsable. Selon la formation un cours lui attribué un coefficient. Un cours peut être dispensé dans plusieurs formations.



III.2/ modélisation par analyse des Dépendances Fonctionnelles (DF)

Cette méthode consiste à :

1. Déterminer le dictionnaire de données épuré
2. Rechercher les dépendances fonctionnelles entre les propriétés recensées à la première étape.

a) Notion de dépendance fonctionnelle :

Définition: Une propriété B dépend fonctionnellement d'une propriété A ssi à une seule valeur de A, il n'est possible d'associer qu'une et une seule valeur de B. On note **A → B**, la réciproque est fausse.

Code_client → Nom_client : Vraie

mais Nom_client → code_client n'est pas vraie.

Astuce : trouver les identifiants (clés) : Toutes les propriétés d'une entité « dépendent fonctionnellement » de l'identifiant $\Rightarrow$ Dépendances fonctionnelles directes (**éliminer la transitivité**)

Exemple :

Num_prof $\rightarrow$ code_matière
 Code_matière $\rightarrow$ nom_matière } DF Directes
 Num_prof $\rightarrow$ nom_matière $\longrightarrow$ DF Transitive à éliminer.

Dépendances fonctionnelles directes (Exo 1 du TD)

Nom	dépend de
N° facture	N° facture
N° magasin	N° facture
Nom magasin	N° magasin
Nom client	N° client
N° client	N° facture
Adresse 1	N° client
Adresse 2	N° client
Adresse 3	N° client
Code postal	N° client
Ville	N° client
N° produit	N° produit
libellé produit	N° produit
Quantité	N° produit
Prix unitaire	N° produit
Code TVA	N° produit
Taux TVA	Code TVA
Quantité en stocks	
Entrée stock	

Remarque : introduction de l'information codeTVA : chaque produit est soumis à un taux de TVA. Plus facile pour les mises à jour éventuelles du taux de TVA.

b) Notion de dépendances fonctionnelles composées :

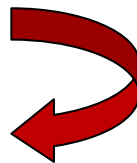
Deux possibilités :

- La propriété dépend fonctionnellement de plusieurs autres propriétés : la dépendance fonctionnelle est composée.
Qté_cdé $\rightarrow$ code_prd + num_commande ;
- L'information ne possède pas de dépendance, alors l'information dépend fonctionnellement d'elle même.

Exemple : N_facture, N_produit.

Nom	dépend de
N° facture	N° facture
N° magasin	N° facture
Nom magasin	N° magasin
Nom client	N° client
N° client	N° facture
Adresse 1	N° client
Adresse 2	N° client
Adresse 3	N° client
Code postal	N° client
Ville	N° client
N° produit	N° produit
libellé produit	N° produit
Quantité	N° produit, N° facture
Prix unitaire	N° produit
Code TVA	N° produit
Taux TVA	Code TVA
Quantité en stocks	N° produit, N° magasin
Entrée stock	N° produit, N° magasin

Dépendances fonctionnelles directes et composées

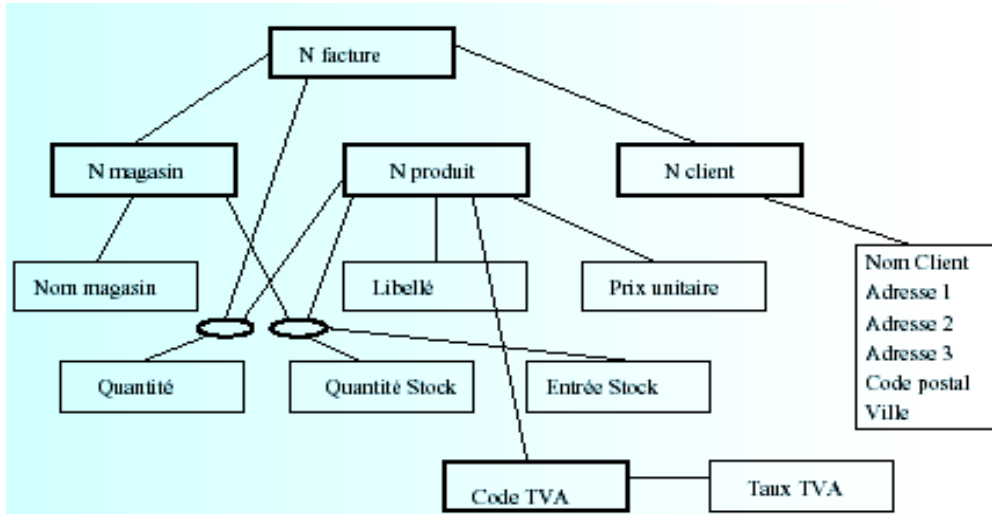


- Quelques exemples :
 - n_Magasin est un identifiant pour nomMagasin
 - n_Facture n'est pas un identifiant pour n_Magasin :
 - n_Magasin dépend bien de n_Facture (DF entre Identifiants) mais nomMagasin dépend de n_Magasin

Certaines dépendances fonctionnelles peuvent être « parasites » car elles peuvent être déduites d'autres dépendances fonctionnelles par application des propriétés remarquables telles que la réflexivité ou la transitivité. Il faut donc les éliminer pour obtenir l'ensemble minimal des dépendances fonctionnelles qui représente la même information.

c) La structure d'accès théorique (S.A.T.) ou Graphe de dépendance fonctionnelles (GDF)

La S.A.T. est une représentation graphique des dépendances fonctionnelles. La S.A.T. permet de déceler les transitivités.



d) Construction du MCD à partir du GDF :

1. Toute propriété identifiante donne naissance à une entité dont le contenu sera formé des propriétés avec lesquelles elle est en dépendance.
2. Les propriétés atteintes par des dépendances fonctionnelles multi-attributs seront intégrées à des associations porteuses (m-n) reliant les entités dont les identifiants sont spécifiés dans la source.
3. les dépendances entre identifiants se matérialiseront par la présence d'une association (1-n).

IV/Normalisation du modèle :

La normalisation a pour objectif de vérifier la non redondance de l'information dans le modèle et de proposer les transformations applicables sans perte d'informations. Pour être guidé dans la conception, nous retiendrons trois règles connues sous le nom de Formes Normales (FN). Ces FN s'appliquent aux entités et aux relations.

Il est à remarquer que ce processus de Normalisation peut se faire soit au niveau du MCD, soit au niveau du modèle relationnel.

IV.1/ 1^{ère} Forme Normale : (élémentarité des constituants)

- Une entité ou une association ne contient pas de propriété répétitive ou décomposable.
- Chaque entité doit disposer d'un identifiant qui la caractérise de manière unique.

IV.2/ 2^{ème} Forme Normale : (pas de dépendance partielle → dépendance pleine de la clé)

Une entité est en 2FN si :

- Elle est en 1FN et
- Toute propriété non identifiante dépend de la totalité de l'identifiant et pas seulement d'une partie de celui-ci. Un identifiant peut être composé de la concaténation de plusieurs propriétés.

Remarque : Si l'identifiant est formé d'une seule propriété, l'entité est toujours en 2FN.

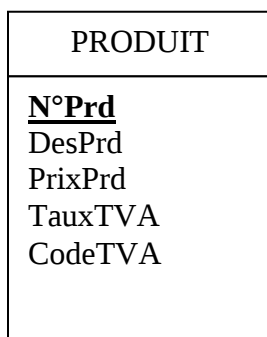
IV.3/ 3^{ème} Forme Normale : (pas de dépendance transitive)

Une Entité est en 3FN si :

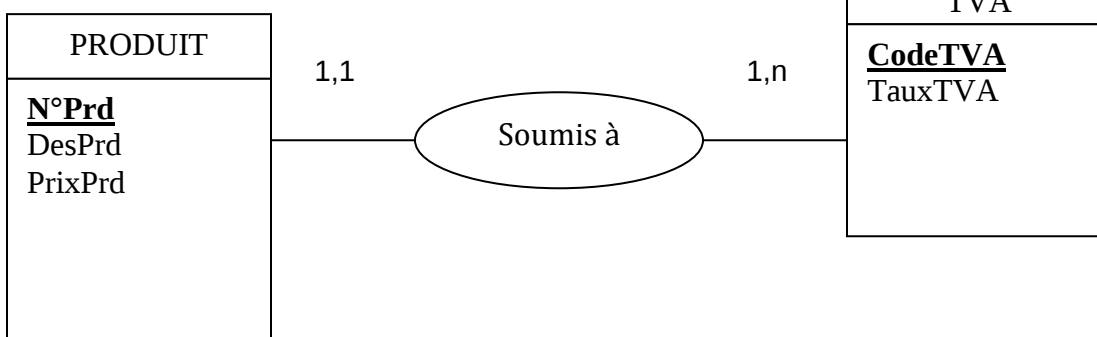
- Elle est en 2FN et
- Toute propriété non identifiante **dépend directement** de l'identifiant, et non d'une propriété non identifiante

Exemple :

Soit l'entité ELEVE suivante :



On peut se poser la question si **TauxTVA** dépend bien de **N°Prd**. Dans cette entité, **TauxTVA** ne dépend pas de **N°Prd** mais bien de **CodeTVA** ; Cette entité n'est donc pas en 3FN → il faut la décomposer comme suit :



IV.4/Normalisation des Relations :

Les propriétés des relations doivent dépendre de tous les identifiants des entités associés.



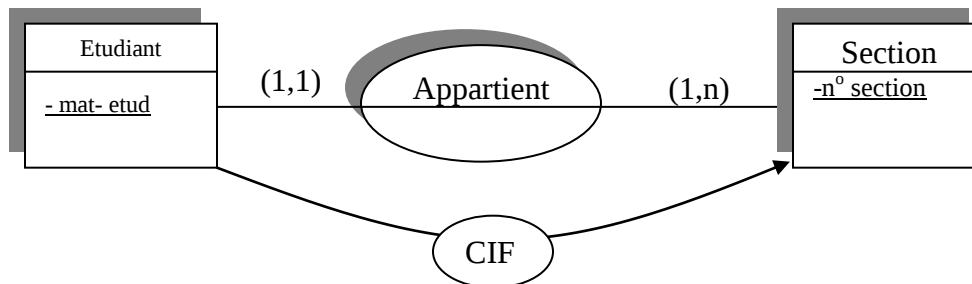
Date_BL ne dépend que de *N°BL*, alors ce n'est pas une propriété de la relation mais plutôt de l'entité **Bon Livraison**.

V/Décomposition des Relations :**a) décomposition :**

Les relations dont le nombre d'entités associées est trop important (supérieur à 3) doivent être décomposées en plusieurs relations de dimensions plus petites en utilisant les DF et les CIF que l'on peut détecter sur les relations.

b) Notion de contrainte d'intégrité fonctionnelle (CIF) :

Une CIF, sur plusieurs entités participant à une même relation, exprime que l'une des entités est totalement déterminée par la connaissance d'une ou plusieurs entités. Elle traduit un lien fort et permanent de la dépendance d'une entité par rapport à une autre entité.

Exemple1 :

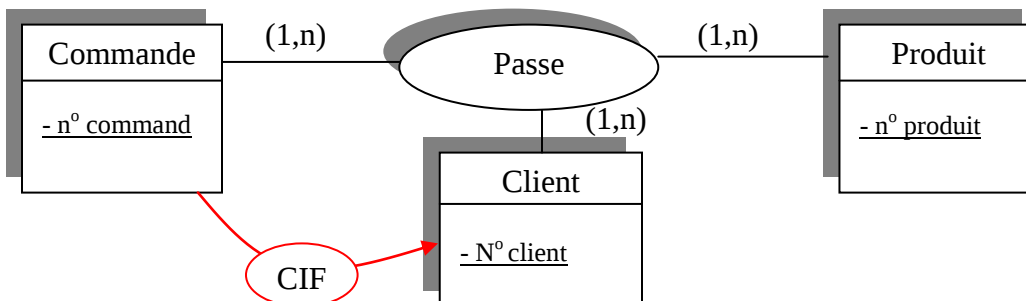
Il existe une dépendance fonctionnelle entre l'entité **Etudiant** et **Section**.

La connaissance de l'étudiant détermine sans ambiguïté la section donc il existe une CIF entre **Etudiant** et **Section**.

Remarque :

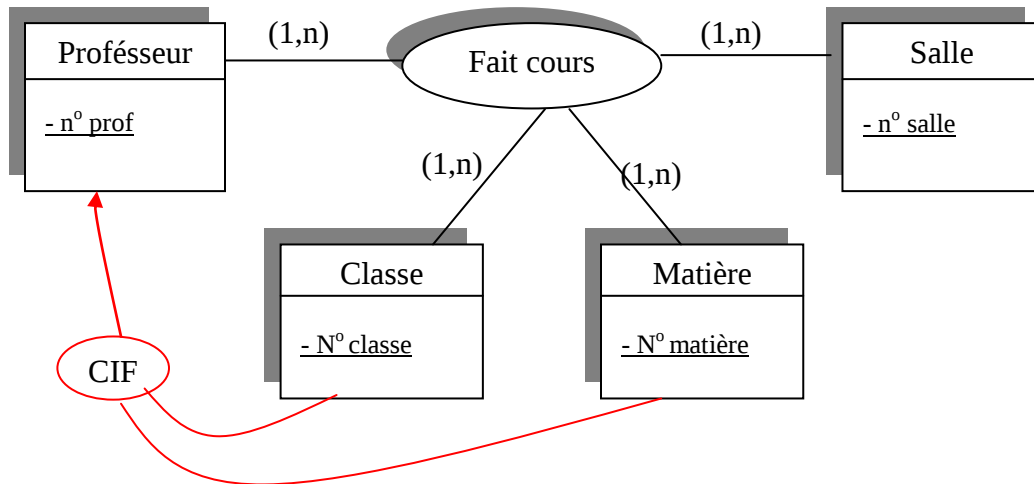
Dans une relation binaire, la présence des cardinalités (0,1) et (1,1) sur une entité exprime la présence d'une CIF pointant vers l'autre entité.

Exemple2 : soit l'association suivante :



On a : commande → client

Exemple3 :



Et soit la règle de gestion suivante : pour une classe et une matière il y a un professeur.

On a : classe + Matière $\rightarrow$ professeur

c) Partialité et totalité :

Une relation qui met en jeux des entités est dite :

- **Totale** : vis à vis de l'Entité E_i si toutes les occurrences de cette Entité participent à la relation (cardinalité Min = 1)
- **Partielle** : vis à vis de l'Entité E_i s'il existe des occurrences de cette Entité qui ne participent pas à la relation (cardinalité minimale égal 0(zéro)).

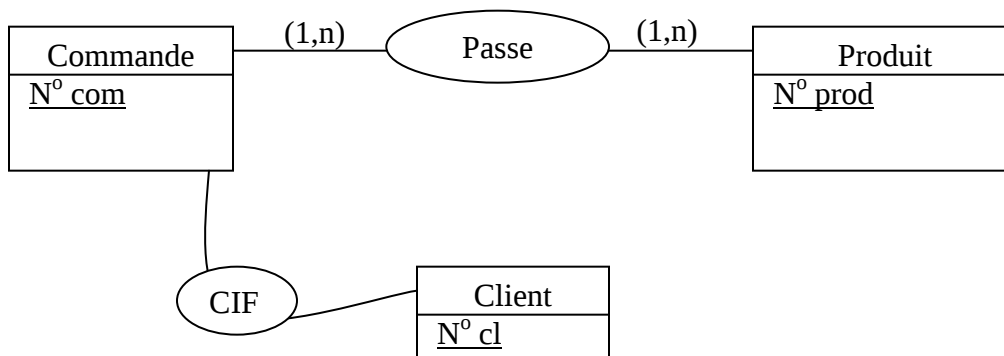
d) Condition de décomposition d'une relation de dimension > 2

La décomposition d'une relation R de dimension $n > 2$ et de la collection d'Entités qu'elle implique $(E_1, E_2, \dots, E_n)$ n'est possible que si les conditions suivantes sont vérifiées.

- 1- Il existe une CIF entre une sous collection de $R(E_1, E_2, \dots, E_n)$ et une Entité E_i .
- 2- La dimension de la sous-collection est inférieure à $(n-1)$.
- 3- La relation est totale vis à vis de la sous-collection (cardinalité minimale des entités sources d la CIF= 1).

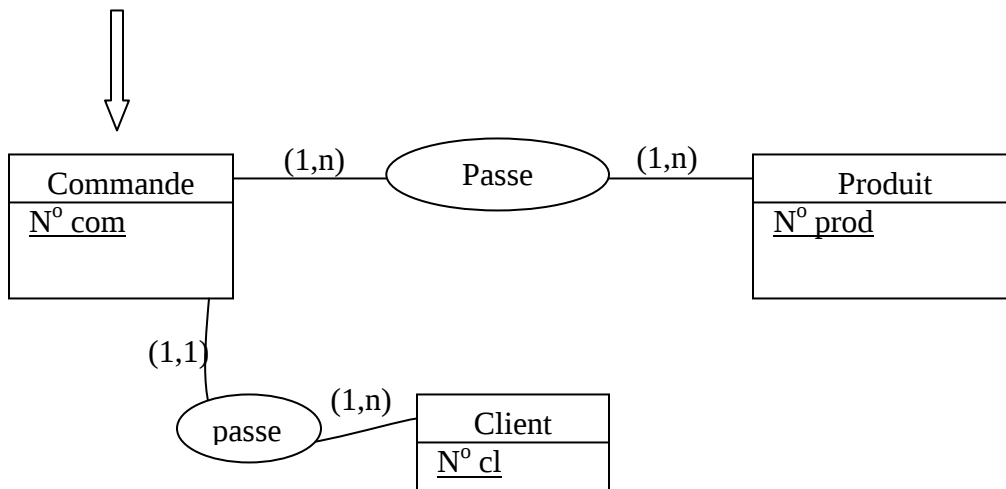
La décomposition se fait comme suit :

- 1- La CIF devient une relation entre la sous-collection et l'Entité E_i .
- 2- Supprimer la pte qui la relie la Relation R à l'Entité cible E_i .

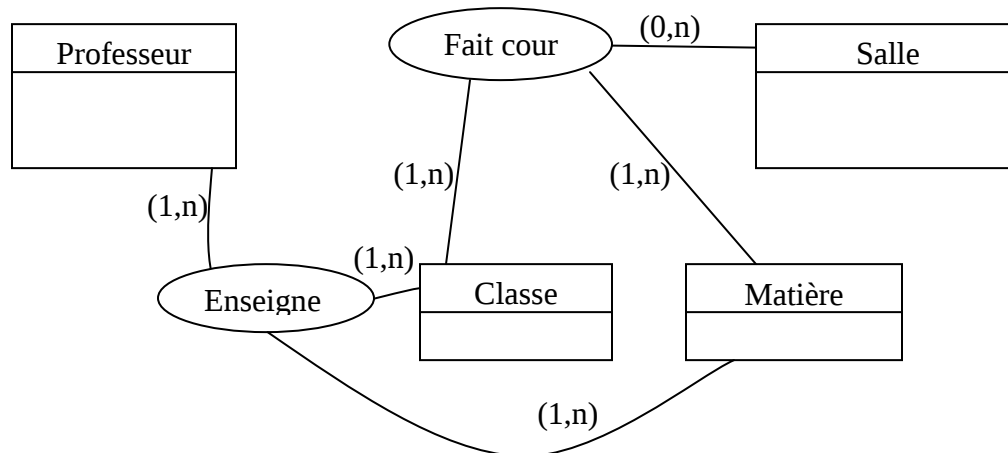
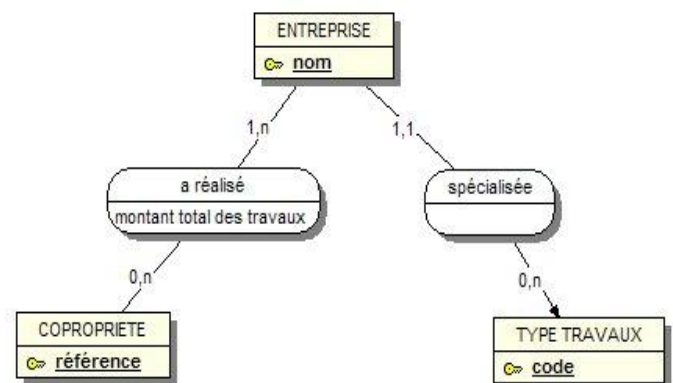


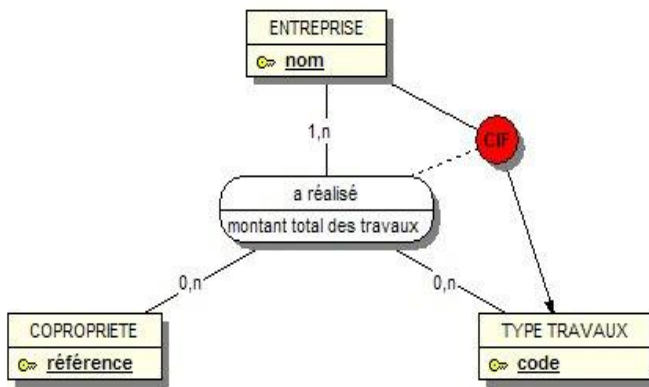
Dim $R=3$
 $K=1 < 3-1=2$
 Cardinalité Min =1

Règle de gestion :
*Une commande est
 passé par un et un*

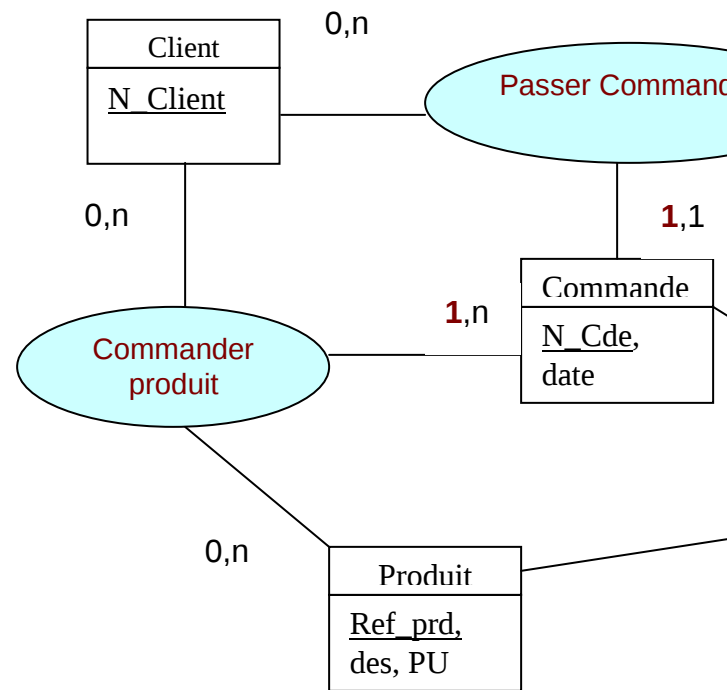
seul client**Suite de l'Exemple3 :****Condition :**1 : DIM (*fait cours*) = 4.

2 : Dim (sous collection=(Matière, Classe)) = 2 < 4-1).

3 : la relation (*fait cours*) est totale vis à vis des Entités (matière, classe).**Exemple4 :**

**Exercice d'application :**

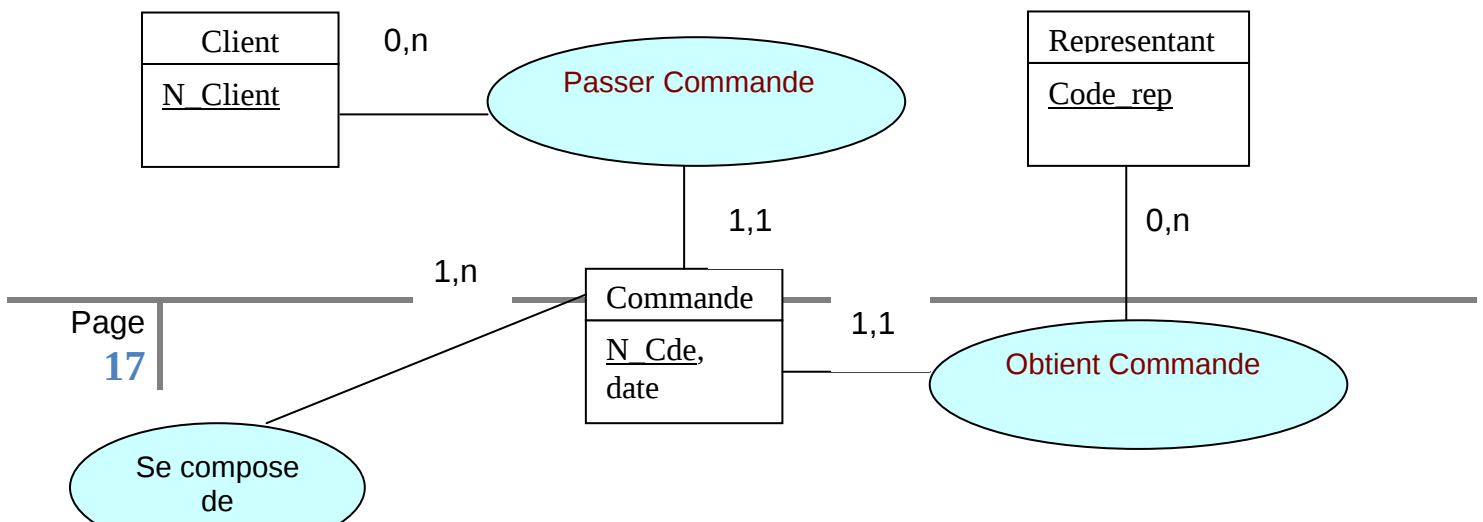
Soit le MCD suivant :

**Remarque :**

Si la DF provient d'une autre relation que celle à décomposer, il faut qu'elle concerne les mêmes occurrences d'entités que la relation à décomposer.

Dans la relation « **Commander Produit** », on a la CIF $N_Cde \rightarrow N_Client$, on peut donc la décomposer en deux relations : une entre Commande et Client : elle existe déjà il s'agit de « Passer Commande » ; l'autre entre Commande et produit : on retrouve « se compose de ». Noter que le DF utilisée provient de la relation « Passer commande » qui met en jeu les mêmes occurrences de « Client » et « Commande » que la relation à décomposer (respect de la remarque)

Pour la relation « **Passer commande** », on peut la décomposer en deux relations en se basant sur les CIF : $N_Cde \rightarrow N_Client$ et $N_Cde \rightarrow Code_rep$; on obtient alors le MCD simplifié suivant :



Série n°1: Modèle Conceptuel de données (MCD)

Exercice 1: Étude de cas de gestion des logements dans une agence immobilière.

Une agence de location de maisons et d'appartements désire gérer sa liste de logements. Elle voudrait en effet connaître l'implantation de chaque logement (nom de la commune et du quartier) ainsi que les personnes qui les occupent (les signataires uniquement).

Le loyer dépend d'un logement, mais en fonction de son type (maison, studio, T1, T2...) l'agence facturera toujours en plus du loyer la même somme forfaitaire à ses clients. Par exemple, le prix d'un studio sera toujours égal au prix du loyer + 30 € de charges forfaitaires par mois.

Pour chaque logement, on veut disposer également de l'adresse, de la superficie ainsi que du loyer.

Quant aux individus qui occupent les logements (les signataires du contrat uniquement), on se contentera de leurs noms, prénoms, date de naissance et numéro de téléphone. Pour chaque commune, on désire connaître le nombre d'habitants ainsi que la distance séparant la commune de l'agence.

- 1. À partir de l'énoncé, exprimer le choix de gestion de l'agence immobilière.**
- 2. Trouver la liste des informations retenues dans le modèle conceptuel de données.**
- 3. Établir le modèle conceptuel de données correspondant.**

Exercice 2: Étude de cas de gestion des concours d'apprentis menuisiers.

Une école désire gérer la participation de ses apprentis à divers concours de menuiserie. Chaque apprenti est encadré par un tuteur de l'école.

Dans chaque concours, l'apprenti doit réaliser un objet qu'il choisit lui-même. Le jury accorde toujours un nombre de points qui permet d'établir le classement (si 2 apprentis ont le même nombre de points, ils sont ex-æquos).

On désire connaître les concours auxquels ont participé les apprentis, l'objet réalisé, la place et le nombre de points qu'ils ont obtenus (NB : pour la place, on ne gère que les apprentis de l'école).

Les informations collectées sont :

- nom de l'apprenti
- prénom de l'apprenti
- nom du tuteur
- prénom du tuteur
- nom du concours
- lieu du concours
- dotation globale du concours
- date du concours
- nombre de points obtenus
- place obtenue
- nom objet réalisé

Établir le modèle conceptuel des données correspondant.

Exercice 3: Étude de cas de gestion des résultats au parcours du combattant

Une caserne militaire désire gérer les résultats des soldats lors du passage des obstacles du parcours du combattant. Dans sa carrière, un soldat va passer plusieurs fois le parcours du combattant.

À chaque fois qu'un soldat passe un obstacle, un instructeur lui attribue une note (note instructeur). Si le parcours comporte 20 obstacles, l'élève recevra donc 20 notes (si l'élève ne passe pas l'obstacle, la note 0 lui est attribuée).

À chaque obstacle est attribué un niveau de difficulté. (facile , moyen, difficile ...). Un bonus de points est ensuite attribué à chaque niveau (ex : bonus de 2 points pour les obstacles difficiles).

La note finale pour le passage d'un obstacle est donc égale à : note attribuée par l'instructeur + bonus relatif à la difficulté de l'obstacle.

Enfin, une note minimale à obtenir est définie pour chaque obstacle. Elle définit un niveau minimum à atteindre qui permet de dire à un soldat sur quels obstacles il doit axer en priorité son entraînement.

Exemple : soit l'obstacle « Fosse » de niveau « difficile » (le bonus attribué pour ce niveau est de 2 points). La note minimale à atteindre pour cet obstacle est de 10. Si un élève est noté 6 sur cet obstacle par l'instructeur, sa note finale sera égale à $6 + 2 = 8$. On juge donc que son niveau sur cet obstacle est insuffisant et qu'il lui faut parfaire son entraînement.

Les responsables de la caserne souhaitent obtenir la liste de tous les obstacles ainsi que leur niveau de difficulté. Ils souhaitent également obtenir la liste de toutes les notes attribuées sur chacun des obstacles. Enfin, ils désirent avoir le récapitulatif des notes obtenues par un soldat donné pour retracer sa progression, ainsi que le temps total qu'il a mis pour effectuer un parcours complet (ainsi que les temps intermédiaires).

Établir le modèle conceptuel des données correspondant.

Exercice 4: Étude de cas de gestion des courses hippiques.

On désire gérer les participations des divers chevaux et jockeys aux courses hippiques : connaître les participants d'une course et leur classement.

Une course se déroule toujours sur le même champ de course et appartient toujours à la même catégorie (exemple de catégorie : trot attelé, trot monté, obstacle, ...).

On désire connaître les catégories de course qu'un champ de course peut accueillir. On désire de plus gérer les informations suivantes :

- la désignation de la course (ex : prix d'Amérique)
- le nom du champ de course
- le nombre de places dans les tribunes
- la date de la course (cette date est variable)
- la dotation de la course en euros (cette dotation est variable)
- le nom des chevaux
- le nom et le prénom du propriétaire (on supposera qu'il n'y en a qu'un et on ne gèrera pas l'historique)
- le sexe du cheval
- le nom et prénom des jockeys
- la date de naissance de chaque cheval.
- le numéro de dossard du jockey et du cheval pour la course

NB : on désire de plus gérer les liens de parenté directs entre les chevaux. Une même course peut avoir lieu plusieurs fois dans la même saison sur le même champ de course et les dotations ne sont pas toujours les mêmes.

Ex : le trot monté d'Auteuil se déroule au mois de mars avec une dotation de 5 millions, au mois de juillet avec une dotation de 3 millions et au mois de décembre avec une dotation de 4 millions.

Établir le modèle conceptuel de données correspondant.

2 ^{ème} année Management		Informatique de Gestion: Base de Données
-----------------------------------	--	--

Exercice 5: Étude de cas de gestion d'un zoo.

Synthèse des entretiens avec le directeur du zoo :

Chaque animal qui est accueilli reçoit un nom de baptême qui sert à le repérer par rapport aux autres animaux de son espèce. Un nom ne peut pas être réutilisé pour un animal de la même espèce, même après le décès de l'animal qui portait ce nom.

Avec l'aide d'un expert, on identifie son espèce :

On aura donc :

- LEO le lion d'Afrique
- TITI le canari

Mais aussi :

- TITI le ouistiti

À côté de chaque enclos ou cage est affiché l'arbre généalogique sur 2 générations (parents et grands-parents), et ceci pour tous les animaux du zoo.

Quand il y a des travaux dans le zoo, on est amené à loger certains animaux d'espèces différentes dans le même enclos. Il faut donc connaître les espèces qui peuvent cohabiter ainsi que l'adéquation entre l'espèce et l'enclos.

En ce qui concerne la nourriture, chaque animal reçoit un repas conformément à un menu type, spécifique de son espèce. Les menus types sont en fait un aliment composé qui est défini par une proportion de protides, glucides et de lipides.

Pour chaque menu est spécifiée la quantité recommandée d'aliment composé à distribuer.

On note enfin à chaque repas la date et l'heure des repas ainsi que les quantités effectivement distribuées.

Établir le modèle conceptuel des données correspondant puis.

Corrigé de la Série n°1: Modèle Conceptuel de données (MCD)

Exercice 1: Étude de cas de gestion des logements dans une agence immobilière.

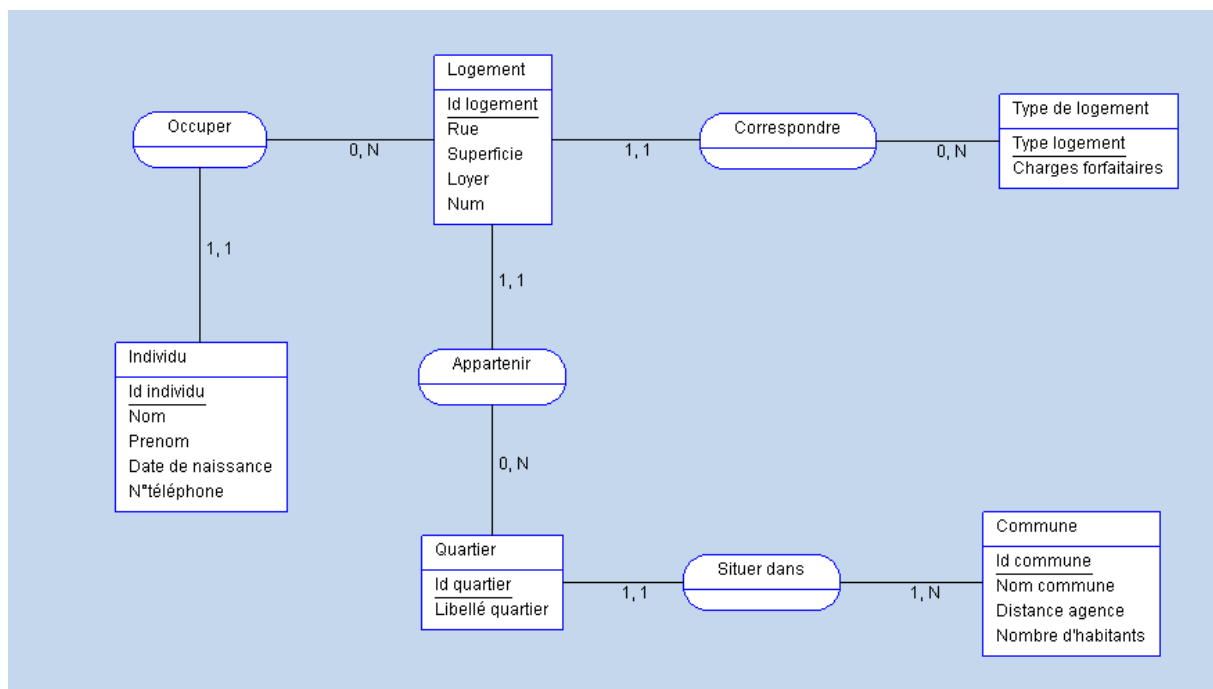
1)-Choix de gestion :

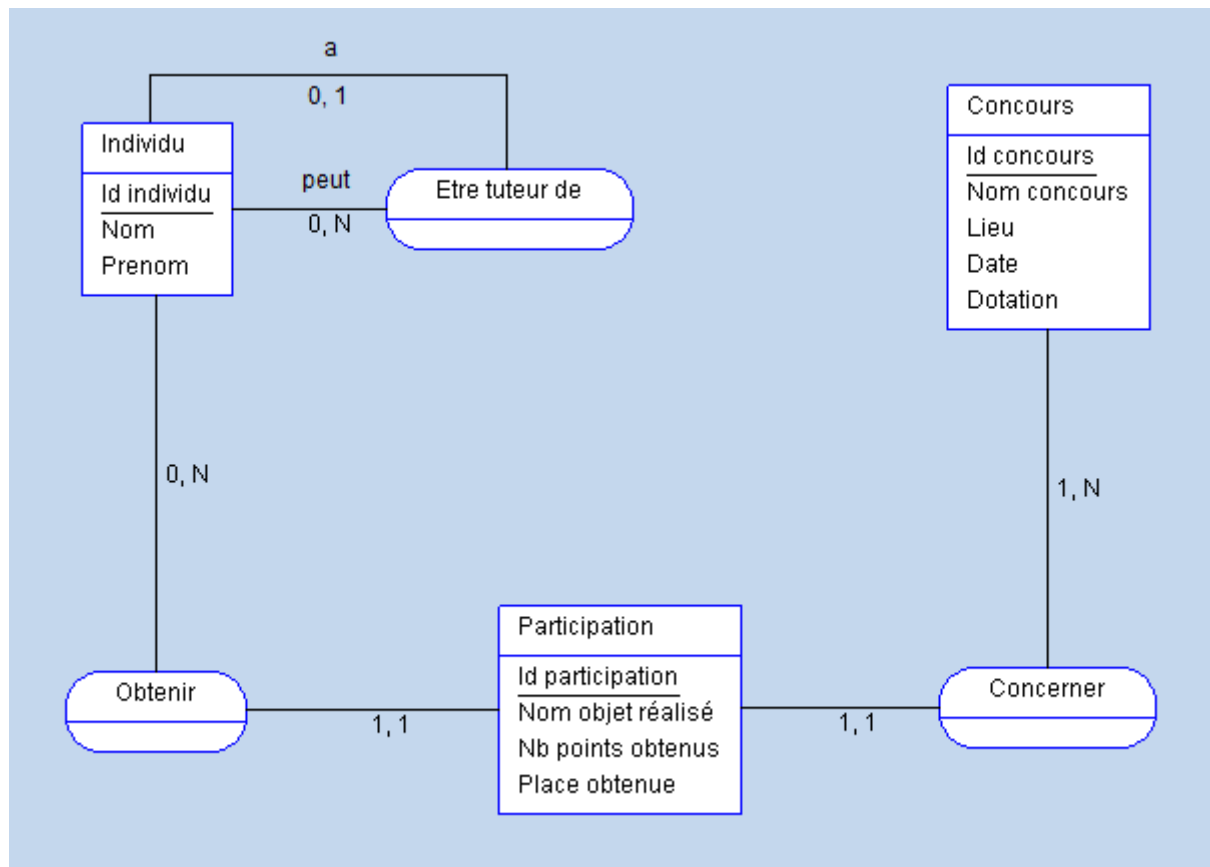
- L'unité géographique retenue pour la gestion des logements est le quartier et on considère que chaque commune possède au moins un quartier.
- On ne s'intéresse qu'aux signataires du contrat uniquement et pas aux locataires.
- Les logements inoccupés font également partie de la gestion.
- L'historique des occupations des logements n'est pas utile.

2)- la liste des informations retenues dans le modèle conceptuel de données:

- Id commune, Nom commune, Distance agence, Nombre d'habitants,
- Id individu, Nom, Prénom, Date de naissance, N°téléphone,
- Id logement, num, Rue, Superficie, Loyer,
- Id quartier, Libellé quartier,
- Type logement, Charges forfaitaires.

3)-



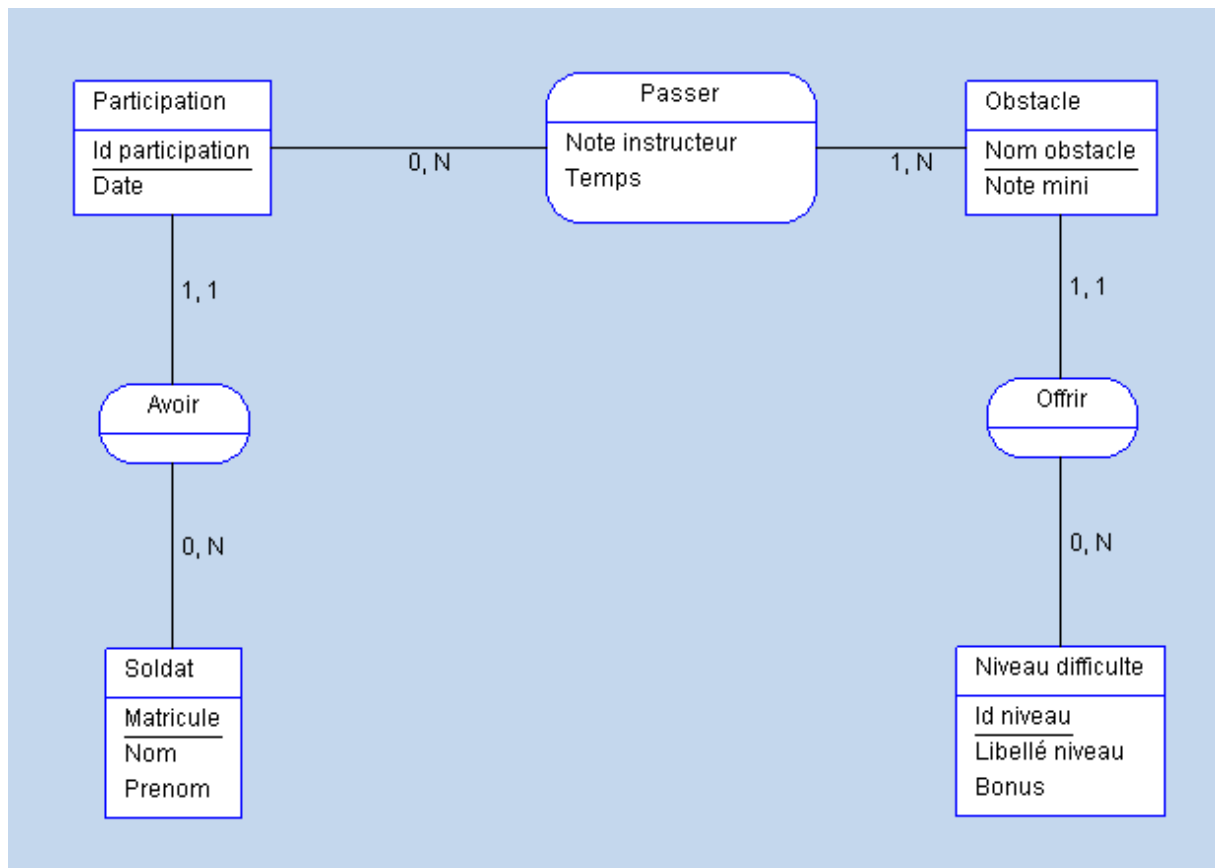
Exercice 2: Étude de cas de gestion des concours d'apprentis menuisiers.

Exercice 3: Étude de cas de gestion des résultats au parcours du combattant**Choix de gestion :**

Il n'y a pas de transition entre les obstacles. Le temps total pour le parcours est donc la somme des temps obtenus sur chacun des obstacles.

Liste des informations retenues dans le modèle :

- Matricule, Nom, Prénom,
- Id participation, Date,
- Note instructeur, Temps,
- Nom obstacle, Note mini,
- Id niveau, Libellé niveau, Bonus.



Exercice 4: Étude de cas de gestion des courses hippiques.