

## Grille de structuration d'un cours en ligne

Conception et Propriétés des BD
<b>Enseignant</b> COULIBALY MOUSSA Enseignant vacataire Ingénieur conception des SI au CHU-T moussa.coulibaly97@gmail.com 70278896
<b>A- Système d'entrée</b>
<ul style="list-style-type: none"> <li>▪ Présentation</li> </ul> <p>Ce cours sur la conception et propriétés des Bases de données permet de comprendre l'information, de connaître comment structurer les informations relatives à un système de gestion et enfin les adapter aux SGBD (Système de Gestion des Bases de données). Tous les aspects véritablement importants sont abordés, notamment les définitions sur les bases de données, les concepts de base de l'information, les règles de construction du MCD (Modèle Conceptuel des Données), la génération du MLD (Modèle Logique des Données) à partir du MCD et introduction aux SQL (Structured Query Language).</p>
<ul style="list-style-type: none"> <li>▪ Objectifs du module</li> </ul>
<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>○ Objectif général</li> </ul> <p>Au terme de ce module, vous serez à mesure de :</p> <ul style="list-style-type: none"> <li>• faire une généralité sur les Bases de Données ;</li> <li>• mémoriser les différents concepts de l'information ;</li> <li>• faire le MCD en appliquant les règles de construction ;</li> <li>• générer le MLD à partir du MCD ;</li> <li>• Appliquer les notions de SQL pour générer une Base de données dans un SGBD.</li> </ul> </li> </ul>
<ul style="list-style-type: none"> <li> <ul style="list-style-type: none"> <li>○ Objectifs spécifiques</li> </ul> <p>Au terme de cette séquence, vous serez capable de :</p> <ul style="list-style-type: none"> <li>• <b>OG1 : faire une généralité sur les Bases de Données</b> <ul style="list-style-type: none"> <li>○ Définir une base de données ;</li> <li>○ Définir un SGBD;</li> <li>○ Énumérer les caractéristiques, les qualités et les propriétés d'une base de données ;</li> <li>○ Décrire l'architecture des bases de données du point de vu interne, conceptuel et externe ;</li> <li>○ Citer les fonctionnalités d'une base ;</li> <li>○ Connaître les différents langages utilisés par un SGBD ;</li> <li>○ Décrire le rôle du gestionnaire d'une base de données.</li> </ul> </li> <li>• <b>OG2 : mémoriser les différents concepts de l'information</b> <ul style="list-style-type: none"> <li>○ <i>Décrire et retenir une propriété ;</i></li> <li>○ Savoir regrouper les propriétés en entité</li> <li>○ <i>Appliquer les différentes règles de dépendances fonctionnelles pour bonne structurations des données ;</i></li> <li>○ <i>Décrire et lire une association ou relation et les cardinalités existant entre les entités ;</i></li> <li>○ Appliquer les fonctionnalités d'une relation binaire ;</li> </ul> </li> <li>• <b>OG3 : faire le MCD en appliquant les règles de construction</b> <ul style="list-style-type: none"> <li>○ <i>Connaître quelques démarches de construction d'un MCD ;</i></li> <li>○ <i>Appliquer la construction par dépendance fonctionnelle ;</i> <ul style="list-style-type: none"> <li>○ <i>Construire le dictionnaire de données ;</i></li> <li>○ <i>Construire la matrice de dépendances fonctionnelles ;</i></li> <li>○ <i>Faire le MCD à partir de cette matrice dépendance fonctionnelle ;</i></li> </ul> </li> <li>○ <i>Appliquer les différentes formes normales.</i></li> </ul> </li> <li>• <b>OG4 : générer le MLD à partir du MCD</b> <ul style="list-style-type: none"> <li>○ <i>transformer une entité en une table ;</i></li> <li>○ <i>transformer les relations binaire du type 1-1 ;</i></li> </ul> </li> </ul> </li> </ul>

- transformer les relations binaire du type 1-n ;
- transformer les relations binaire du type m-n ;
- transformer les relations n-aire  $n \geq 3$  ;
- cas particulier des relations réflexives ;
- cas de l'héritage ;
- **OG5 : Appliquer les notions de SQL pour générer une Base de données dans un SGBD.**
  - Utiliser le Langage de Définition de Données
    - CREATE DATABASE
    - CREATE TABLE
    - ALTER
  - Utiliser le Langages de Manipulation des Données
    - INSERT INTO
    - UPDATE
    - DELETE
  - Utiliser le Langage d'Interrogation des Données
    - SELECT FROM WHERE

N

## B- Système d'apprentissage

- Unité d'apprentissage 1 (OS1) : généralité sur les Bases de Données

- contenu

- Objectif

Définir et distinguer une Base de Données et un système de gestion de base de données et connaître de façon générale une base de données.

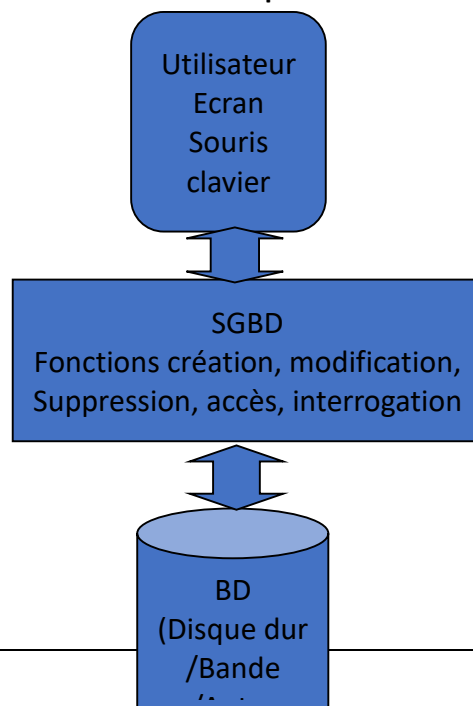
- Définitions

**D'après C.DELOBEL et M.ADIBA** : Une base de données est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun.

**D'après G.WIEDERHOLD en anglais** : When we talk informally about a database, we refer to a collection of mutually related data, to the computer hardware that is used to store it, and to the program that are used to manipulate it.

**Un SGBD (Système de Gestion de Base de Données)** est un système qui permet de créer, de modifier, de supprimer, d'accéder, d'interroger les données contenues dans la BD

- Architecture montrant La place d'un SGBD entre un utilisateur et la base des données



- Caractéristiques, qualités et propriétés d'une BD

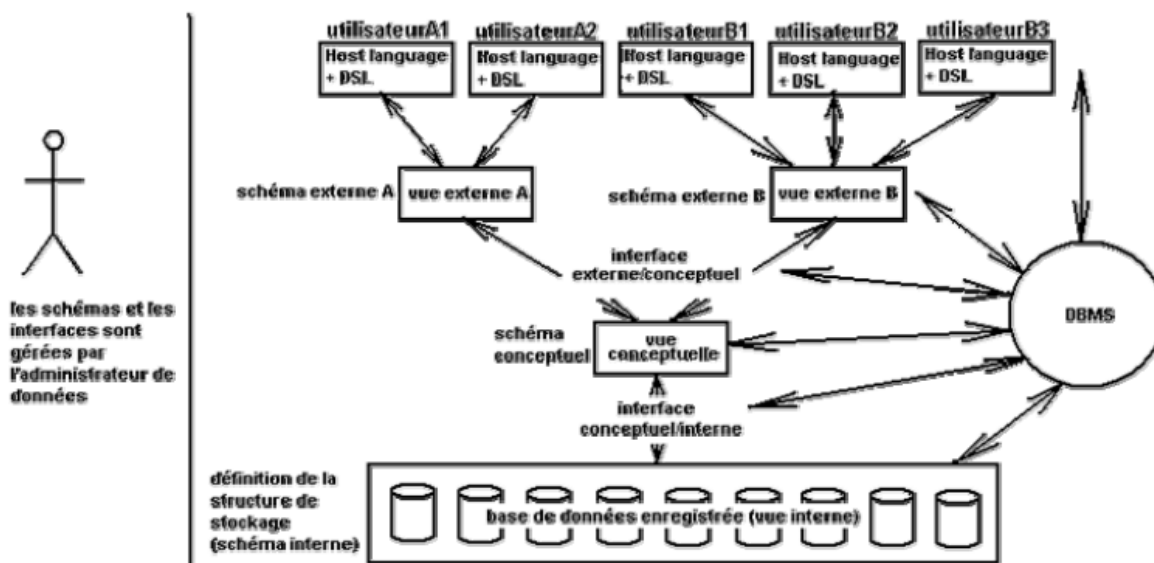
2. exhaustivité des informations (Informations complètes),
3. non-redondance des informations (Pas de répétition d'information),
4. partage des informations (accès simultané, centralisation des données. Problèmes : sécurité, accès concurrent),
5. standard d'organisation (Autorité d'un administrateur de BD ou DBA),
6. sécurité des informations ( droits d'accès),
7. intégrité des informations (exactitude)
8. indépendance des informations vis à vis des applications (ne doit pas être liée au programme).

- Architecture d'une base de données

Cette architecture (ci –dessous) découle de l'étude de l'ANSI (American National Standards Committee on Computers and Information Processing) dans les années 1972. Cette architecture est composée de 3 niveaux (interne, conceptuel, externe) :

1. Interne définit le stockage de l'information sur les mémoires auxiliaires (fichier, index, adressage),
2. Conceptuel (externe ou vue) définit le schéma conceptuel ( les règles de description des données et des relations entre ces données)
3. Le niveau externe contrairement aux précédents niveaux correspond à plusieurs schémas externes qui ne sont autres que les vues(partielles) qu'ont les différents utilisateurs de la base de données.)

**NB** : On notera que la division en 3 niveaux permet une double indépendance des données d'une part des applications et d'autre part les moyens de stockage.



Il existe 4 types de modèles de Base de données :

1. le modèle hiérarchique,
2. le modèle réseau,
3. le modèle relationnel,
4. le modèle objet.

Une base de données est gérée par un SGBD qui doit avoir les fonctionnalités suivantes

- Les fonctionnalités d'un SGBD

1. assurer l'interactivité entre les utilisateurs et la base de données soit :
  - a. dialoguer avec l'utilisateur,
  - b. effectuer les opérations de recherche, de sélection, de mise à jour des informations,
2. décrire l'ensemble des données sous la forme des trois niveaux décrits ci-dessus et assurer les passages entre ces niveaux,

3. maintenir l'intégrité des données par vérification des contraintes d'intégrité.
4. maintenir la sécurité et la confidentialité des données par vérification des droits d'accès des utilisateurs,
5. gérer les accès de plusieurs utilisateurs à une même information,
6. prévoir des solutions de redémarrage en cas d'incidents.

- Les Langages des SGBD

Pour assurer ces fonctionnalités les SGBD utilisent les langages suivants :

un langage de description de données (LDD) qui définit les différents objets de la base de données ( nature, caractéristiques)ainsi que les contraintes d'intégrité qui portent sur eux,

un langage de manipulation de données (LMD) permettant de réaliser les différentes commandes souhaitées par l'utilisateur sur les données.

Un langage d'interrogation des données (LID) permettant à l'utilisateur de faire des requêtes sur les données.

- Rôle de l'administrateur de la Base de données

le bon fonctionnement d'une base de données est assuré, en général, sur les gros systèmes, par des personnels spécialisés: plus particulièrement l'administrateur de la base de données (ABD). Le ABD est une personne ou un groupe de personnes responsables de tout le système d'information, dont les responsabilités portent sur :

1. le choix du contenu de la base de données
2. le choix de la structure de stockage
3. le choix de la stratégie d'accès
4. les relations avec les utilisateurs
5. le choix des autorisations d'accès et des procédures de validation
6. la stratégie de recopie et de sauvegarde
7. le choix des modifications à apporter.

- situations d'apprentissages/activités locales

### Exercice 1

Réorganiser par exemple a-b-e-... pour donner la définition d'une base de données :

a- enregistrées

b- Une base de données est un ensemble

c- sur

d- structuré de données enregistrées

e- pour satisfaire simultanément plusieurs utilisateurs de façon sélective et en un temps opportun

f- des supports accessibles par l'ordinateur

### Exercice 2

Un SGBD est

a- le modèle conceptuel de la Structure Générale de la Base de Données

b- la vue interne du Stockage générique de la Base de Données.

c- un Système de Gestion de la Base de Données.

Un SGBD permet

a- de créer, de modifier, de supprimer, d'accéder, d'interroger les données contenues dans la BD

b- de donner le schéma de la structure des données contenues dans la BD

c- de schématiser seulement le modèle logique des données contenues dans le BD

### Exercice 3

Cocher les bonnes réponses :

1- Les qualités d'une base données sont :

a- standard d'organisation

b- la redondance des données

c- la sécurité des informations

d- l'information non partagée

e- intégrité des données

2- les niveaux proposés par l'architecture ANSI sont :

- a- partiel, interne et externe
- b- total, conceptuel et interne
- c- interne, conceptuel et externe

3- parmi les modèle de base de données qui fait l'objet de notre apprentissage ?

- a- modèle réseaux
- b- modèle objet
- c- modèle hiérarchique
- d- modèle relationnel

4- les différents langages utilisés par SGBD sont :

- a- LDD, LMD et LID
- b- LDM, LMD et LID
- c- IID, LMD et LID

5- Un Administrateur de Base de Données est

- a- une personne ou un groupe de personnes responsables de tout le système d'information, dont les responsabilités portent sur les qualités d'une base de données. (support, accès,...)
- b- personne chargée de la formation en bureautique
- c- personne chargée de l'administration du réseaux

▪ Unité d'apprentissage 2 (OS2) : mémoriser les différents concepts de l'information

○ contenu

L'objectif principal de ce module est de disposer et de maîtriser les différents concepts rentrant dans la construction d'un MCD.

○ Concept sur l'information

○ La propriété (ou attribut ou rubrique)

La propriété est une information élémentaire, c'est-à-dire non déductible d'autres informations, qui présente un intérêt pour le domaine étudié. Par exemple, si l'on considère le domaine de gestion des commandes d'une société de vente par correspondance, les données : « référence article », « désignation article », « prix unitaire HT », « taux de TVA » sont des propriétés pertinentes pour ce domaine.

○ Occurrence d'une propriété

Chaque valeur prise par une propriété est appelée occurrence. Des occurrences de la rubrique « désignation article » sont par exemple : « râteau », « bêche », « scie »,

○ Propriété atomique

Une propriété est dite simple ou encore atomique si chacune des valeurs qu'elle regroupe n'est pas décomposable. La propriété « Adresse » n'est pas élémentaire car elle peut être décomposée en trois propriétés : la rue, le code postal et la ville.

○ Propriété paramètre

Une propriété paramètre est une propriété qui, à un instant donné, contient une seule valeur. Par exemple au BF la TVA est une propriété paramètre pour le taux qui est 18 %.

Remarques : le nom d'une propriété doit être le plus explicite possible : à sa seule lecture on doit donc pouvoir se faire une idée de ce que représente la propriété. En outre, l'identification de chaque propriété consiste à garantir une bijection entre l'ensemble des noms et l'ensemble des propriétés à gérer. On devra donc exclure les synonymes qui correspondent à deux noms différents pour identifier la même propriété et les polysèmes qui représentent deux propriétés différentes ayant le même nom.

Enfin, le principe de non-redondance impose que chaque propriété, correctement identifiée, n'apparaisse qu'une seule fois dans le modèle

- L'entité ou individu-type

- Définition

Une entité est la représentation d'un élément matériel ou immatériel ayant un rôle dans le système que l'on désire décrire.

Au premier abord, on peut définir l'entité comme étant un regroupement bien pensé, donc sensé, de plusieurs propriétés. Par exemple, on considère l'entité ARTICLE qui regroupe les propriétés : Référence, Désignation et PrixUnitaireHT.

- Représentation

<nom entité>	Article
<propriété1>	<u>reference</u>
<propriété2>	designation
	prix unit

- La relation de dépendance fonctionnelle

Considérons deux propriétés P1 et P2. La création d'une entité E regroupant ces deux seules propriétés n'est envisageable que si l'une des deux conditions suivantes est satisfaite :

À toute valeur de la propriété P1 doit correspondre au plus une valeur de la propriété P2. Ce fait traduit l'existence d'une dépendance fonctionnelle monovaluée entre P1 et P2 notée : P1 -> P2. On dit encore que P1 détermine P2. P1 est alors rubrique identifiante de l'entité E.

E
<u>P1</u>
P2

ou à toute valeur de la rubrique P2 doit correspondre au plus une valeur de la rubrique P1. P2 est alors en dépendance fonctionnelle avec P1 et l'entité E doit être représentée ainsi :

E
<u>P2</u>
P1

- 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. Le modèle conceptuel des données propose de souligner les identifiants. 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 :

Article			
<u>reference</u>	134ER	354TY	452GT
designation	Rateau	Bêche	Scie
prix unit	150 F	68,50 F	45 .

- Notion de dépendance fonctionnelle directe

Considérons le SI suivant :

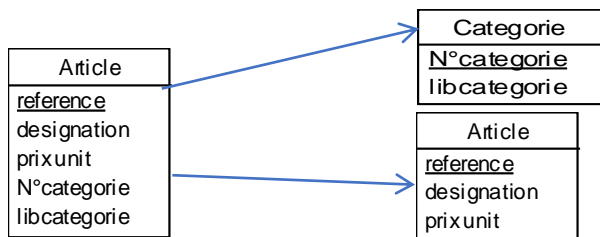
Article	134ER	354TY	452GT
<u>reference</u>	Rateau	Bêche	Scie
designation			

Nous constatons que lib catégorie n'est pas directement lié à référence.

Pour supprimer de telles redondances, on devra veiller à ce que toute dépendance fonctionnelle entre la propriété identifiante de l'entité et une propriété non identifiante de l'entité soit directe. Une dépendance fonctionnelle monovaluée  $x \rightarrow y$  est directe s'il n'existe pas de propriété  $z$  telle que :  $x \rightarrow z$  et  $z \rightarrow y$ .

Dans l'exemple précédent la dépendance fonctionnelle Reference  $\rightarrow$  LibCategorie n'est pas directe car il existe la propriété NoCategorie telle que :

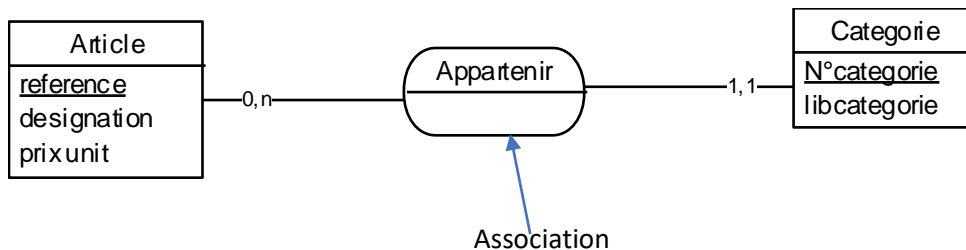
Reference  $\rightarrow$  NoCategorie et NoCategorie  $\rightarrow$  LibCategorie



#### ○ L'association (ou relation-type)

##### ▪ Définition

Une association (appelée aussi parfois relation) est un lien sémantique entre plusieurs entités. Une classe de relation contient donc toutes les relations de même type (qui relient donc des entités appartenant à des mêmes classes d'entité).

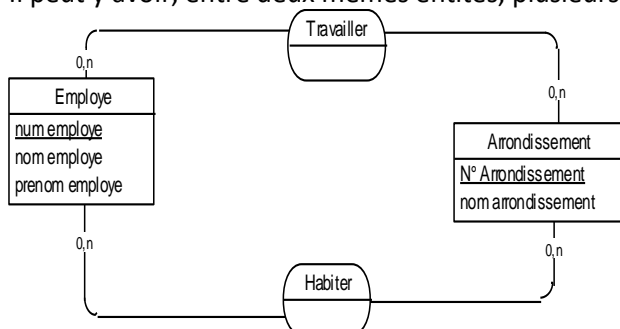


##### ▪ Dénomination des relations

Une classe de relation peut lier plus de deux classes d'entité. Voici les dénominations des classes de relation selon le nombre d'intervenants:

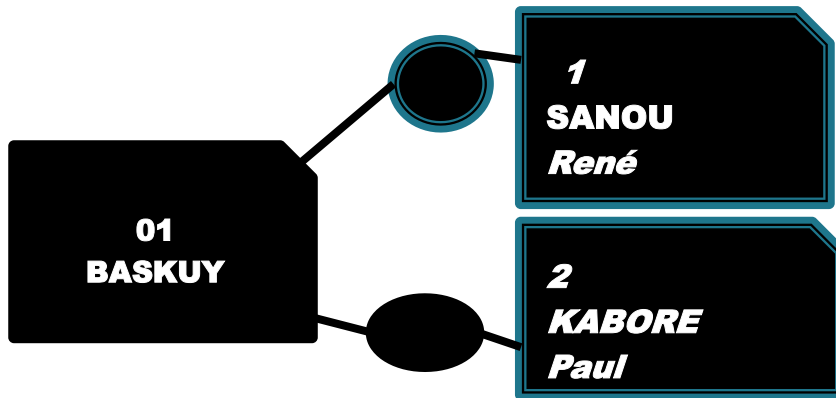
1. une classe de relation récursive (ou réflexive) relie la même classe d'entité
2. une classe de relation binaire relie deux classes d'entité
3. une classe de relation ternaire relie trois classes d'entité
4. Une classe de relation n-aire relie n classes d'entité

Il peut y avoir, entre deux mêmes entités, plusieurs associations qui représentent chacune des réalités différentes.



### ▪ Occurrence d'association

Une occurrence d'association est un lien particulier qui relie deux occurrences d'entités. deux Exemples d'occurrences de l'association « Habite » :



**Remarque** : certains auteurs définissent l'identifiant d'une association comme étant la concaténation des identifiants des entités qui participent à l'association.

Exemple de l'association « Habite » : num\_employe + N°Arrondissement

### ▪ Cardinalité

Les cardinalités permettent de caractériser le lien qui existe entre une entité et la relation à laquelle elle est reliée. La cardinalité d'une relation est composé d'un couple comportant une borne maximale et une borne minimale, intervalle dans lequel la cardinalité d'une entité peut prendre sa valeur :

1. la borne minimale (généralement 0 ou 1) décrit le nombre minimum de fois qu'une entité peut participer à une relation ;
2. la borne maximale (généralement 1 ou n) décrit le nombre maximum de fois qu'une entité peut participer à une relation.

Les cardinalités usuelles existantes sont :

	<b>Pour chaque occurrence de E, le modèle admet :</b> <ul style="list-style-type: none"> <li>– soit l'absence de lien</li> <li>– soit la présence d'un seul lien</li> </ul>
	<b>Pour chaque occurrence de E le modèle admet la présence d'un et un seul lien</b>

	<b>Pour chaque occurrence de E le modèle admet la présence d'un seul ou de plusieurs liens</b>
	<b>Pour chaque occurrence de E le modèle admet :</b> <ul style="list-style-type: none"> <li>– soit l'absence de lien</li> <li>– soit la présence de plusieurs liens</li> </ul>

### ▪ Caractéristiques d'une association

#### La dimension d'une association

La dimension d'une association indique le nombre d'entités participant à l'association. Les dimensions les plus courantes sont 2 (association binaire) et 3 (association ternaire) :

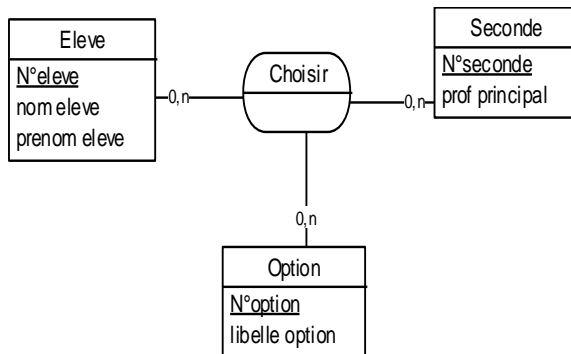
1. L'association binaire exprime la présence de liens sémantiques entre les occurrences d'une entité A et les occurrences d'une entité B. L'association « Habite » entre les EMPLOYE et l'arrondissement est binaire.
2. L'association ternaire exprime la présence de liens sémantiques entre les occurrences de 3 entités.



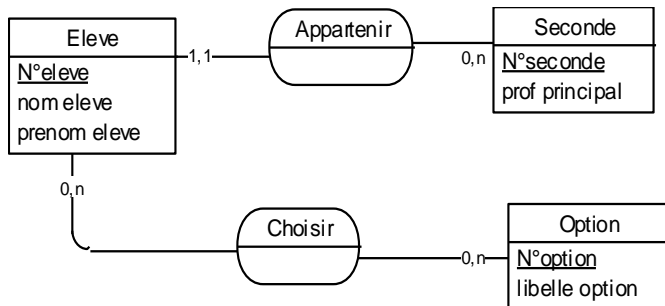
### La décomposition

L'opération de décomposition consiste à éclater une relation de dimension n en une ou plusieurs associations de dimension moindre sans perte de sémantique.

Soit le modèle suivant



Il faut remarquer les dépendances fonctionnelles sont valables entre entités. Dans le modèle ci-dessus nous constatons que Eleve -> seconde. Nous allons couper la patte de l'entité à la relation choisir et créer une nouvelle relation entre Eleve et Seconde. Par conséquent le modèle devient :



#### La fonctionnalité d'une association binaire

Les différents types de fonctionnalités d'une association binaire A définie entre deux entités E1 et E2 sont les suivants :

1. un à un (1 – 1) caractérisé par la cardinalité maximum égale à 1 sur les deux segments « E1 – A » et « E2 – A »,
2. un à plusieurs (1 – n) caractérisé par la cardinalité maximum égale à 1 sur l'un des segments et la cardinalité maximum égale à n sur l'autre,
3. plusieurs à plusieurs (m-n) caractérisé par la cardinalité maximum égale à n sur les deux segments « E1 – A » et « E2 – A ».

#### Les notions de partialité et de totalité

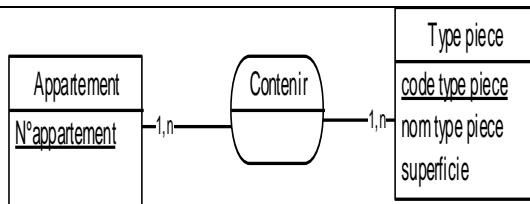
1. Une association binaire A définie entre les entités E1 et E2 est totale si elle est caractérisée par la cardinalité minimale égale à un sur les deux segments « E1 – A » et « E2 – A ».
2. Une association binaire A définie entre les entités E1 et E2 est partielle si elle est caractérisée par la cardinalité minimum égale à 0 sur l'un des segments.

#### L'association porteuse

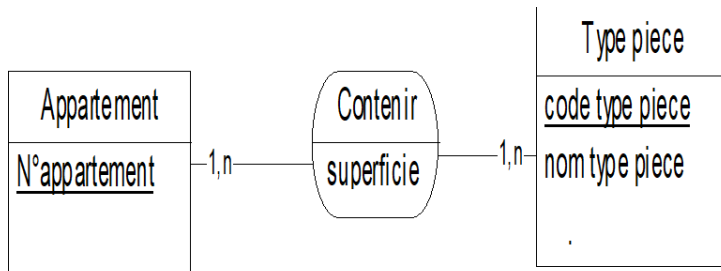
Les propriétés qui dépendent fonctionnellement de plusieurs identifiants d'entités sont portées par les associations entre ces entités. C'est une dépendance fonctionnelle multi-attributs au niveau de la source.

Le droit d'entrée d'une propriété P dans une association reliant n entités est donc soumis à l'existence de la dépendance fonctionnelle suivante : I1,... In -> P où I1, I2, .. représentent l'identifiant de chacune des entités qui participent à l'association.

Normaliser le modèle suivant :

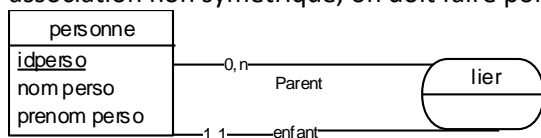


Chaque appartement selon le type de pièce possède des superficies différentes donc nous avons N°appartement, code\_type\_piece -> superficie. Nous obtenons alors



#### ■ Les associations réflexives

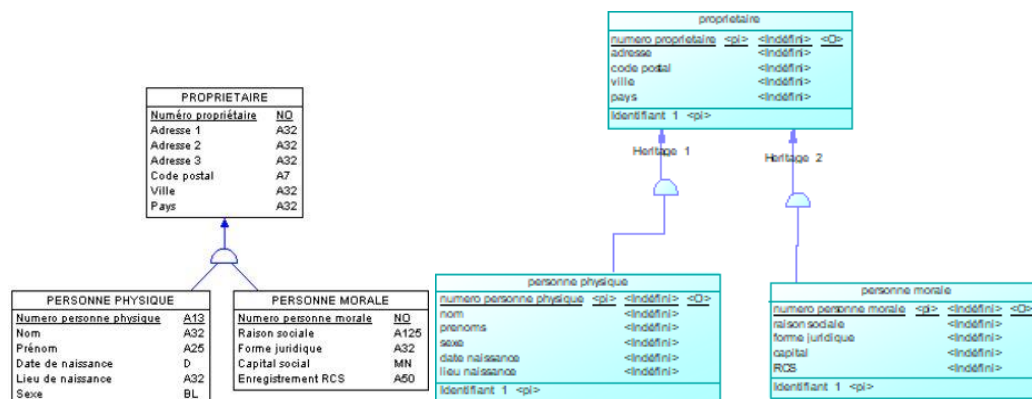
L'association réflexive est une association binaire qui relie une entité à elle-même. Une occurrence de l'association établit donc un lien entre une occurrence de l'entité et une autre occurrence de cette même entité. Dans le cas d'une association non symétrique, on doit faire porter le rôle sur chacun des segments.



#### ■ Généralisation (héritage)

Dans le schéma ci-dessous, les entités "Personne physique" (des êtres humains) et "Personne morales" (des sociétés, associations, collectivités, organisations...) sont généralisées dans l'entité "Propriétaires".

On dit aussi que l'entité "Propriétaire" est une entité parente et que les entités "Personne morale" et "Personne physique" sont des entités enfants, car il y a une notion d'héritage.



- situations d'apprentissages/activités locales

#### Exercice 1

1- une propriété est :

- une information élémentaire qui présente un intérêt pour le domaine d'étude
- une information composée qui présente un intérêt pour le domaine d'étude
- un élément matériel ou immatériel jouant un rôle dans le système
- un lien sémantique entre plusieurs entités

2- une entité est :

- a- une information élémentaire qui présente un intérêt pour le domaine d'étude
- b- une information composée qui présente un intérêt pour le domaine d'étude
- c- un élément matériel ou immatériel jouant un rôle dans le système
- d- un lien sémantique entre plusieurs entités

3- une relation est :

- a- une information élémentaire qui présente un intérêt pour le domaine d'étude
- b- une information composée qui présente un intérêt pour le domaine d'étude
- c- un élément matériel ou immatériel jouant un rôle dans le système
- d- un lien sémantique entre plusieurs entités

4- une occurrence de propriété est

- a- la valeur prise par cette propriété
- b- l'ensemble des valeurs prises par les propriétés qui sont en dépendance fonctionnelle avec l'identifiant.
- c- un lien particulier reliant les occurrences des entités qui participent au lien
- d- un lien binaire

5- une occurrence de l'entité est

- a- la valeur prise par cette propriété
- b- l'ensemble des valeurs prises par les propriétés qui sont en dépendance fonctionnelle avec l'identifiant.
- c- un lien particulier reliant les occurrences des entités qui participent au lien
- d- un lien binaire

6- une occurrence d'association est

- a- la valeur prise par cette propriété
- b- l'ensemble des valeurs prises par les propriétés qui sont dépendance fonctionnelle avec l'identifiant.
- c- un lien particulier reliant les occurrences des entités qui participent au lien
- d- un lien binaire

## Exercice 2

a-

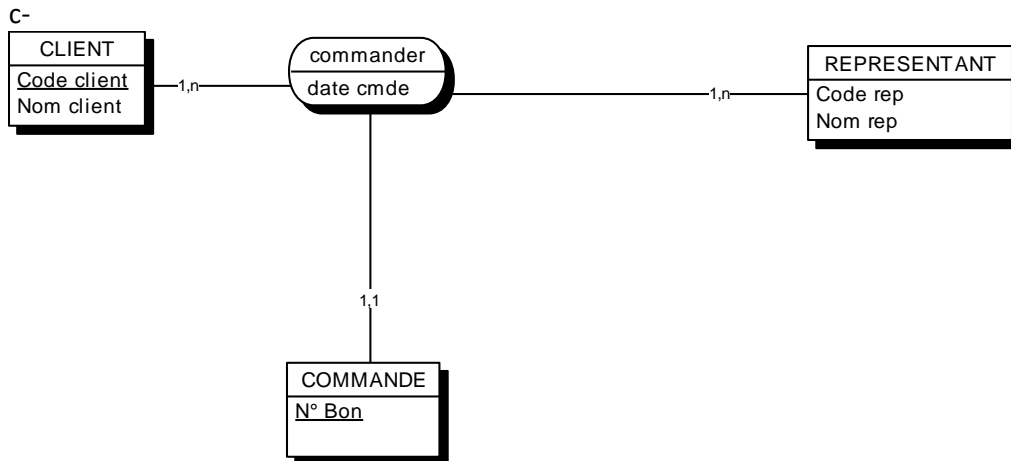
commande
<u>N°commande</u>
<u>referenceproduit</u>
designationproduit
prixunitaireprod
datecommande
numclient
nomclient
prenomclient
quantite

Soit le modèle ci-dessus, normaliser en s'appuyant sur les règles de dépendances fonctionnelles.

b-

EXEMPLAIRE
<u>Code ouvrage</u>
<u>N°exemplaire</u>
Date aquisition
titre
Code auteur
Nom auteur
Code collection
Nom collection

Même question



Même question

▪ Unité d'apprentissage 3 (OS3) : Règles de construction d'un M.C.D.

○ contenu

Pour la construction du modèle conceptuel, beaucoup de méthodes ont été mises en place mais aucune ne donne réellement satisfaction. On peut cependant les répartir en deux catégories :

1. modélisation directe
2. modélisation par analyse des dépendances. fonctionnelles

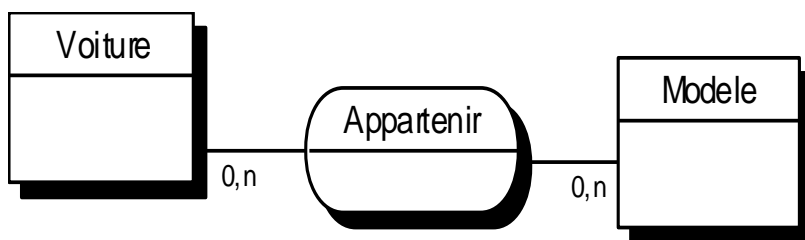
○ modélisation directe

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

1. les noms deviennent des entités
2. 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 petite taille.

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



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.

○ modélisation par analyse des dépendances. fonctionnelles

Cette méthode consiste à identifier en premier lieu toutes les propriétés du système d'information à analyser. Cette étape aboutit au dictionnaire des données épuré qui devra comporter ni synonyme, ni polysème, ni donnée calculée. Considérons le modèle suivant :

**Nom société : SOGECM**

**Le 12/09/2008**

**Tél : 50-30-12-56**

**BON DE COMMANDE N° : 1209200501**

**Doit :**

**Nom client : ZONGO Mahamoudou**

**Adresse : 01 BP 1245 Ouaga 01**

**Tél : 50-31-12-56**

**Pays : Burkina Faso**

**Détail de la commande**

Désignation	PU	Qtité	Montant
<b>Carton de sucre sosuco</b>	<b>25 000</b>	<b>5</b>	<b>125 000</b>
<b>Sacs de riz</b>	<b>14 000</b>	<b>10</b>	<b>140 000</b>
<b>Carton de spaghetti</b>	<b>3 000</b>	<b>10</b>	<b>30 000</b>
<b>Total HTVA</b>			<b>295 000</b>
<b>TVA 18%</b>			<b>53 100</b>
<b>Total TTC</b>			<b>348 100</b>

Travail à faire

- 1- Etablir le Dictionnaire données brutes
- 2- Etablir le dictionnaire de données épurées
- 3- Etablir la Matrice des dépendances fonctionnelles
- 4- Déduire le MCD

Pour faciliter la conception ultérieure des bases de données, il est recommandé de définir pour chaque donnée du dictionnaire son domaine. Le domaine d'une donnée est l'ensemble des valeurs que peut prendre cette donnée. Il peut être :

étendu: il correspond alors au type d'une donnée : Numérique, alphabétique, etc.

restreint: on l'exprime alors au moyen d'une liste ou d'un intervalle. Par exemple, pour la rubrique « Sexe », le domaine sera la liste de valeurs « F », « M ».

Notons les différents type :

A : Alphabétique

AN : Alpha-Numérique

N : Numérique

D : Date

B : Booléen

## 1- le dictionnaire brut de données

N°	Code	Désignation	Type	Taille	Règle de calcul
01	nom_soc	Nom de la société	AN	50	Néant
02	tel_soc	Téléphone de la société	N	15	Néant
03	date_com	Date de la commande	D		
04	num_bon	Numéro du bon	AN	20	Néant
05	nom_client	Nom du client	AN	50	Néant
06	pren_client	Prénoms du client	AN	50	Néant
07	Adres_cli	Adresse du client	AN	50	Code postal + Boîte postal + ville
08	tel_cli	Téléphone client	N	15	Néant
09	pay_cli	Pays du clieny	AN	50	Néant
10	design_prod	Désignation produit	AN	50	Néant
11	pu	Prix unitaire	N	14	Néant
12	qtte_cmd	Quantité commandée	N	14	Néant
13	mont_lign	Montant de la ligne de commande	N	14	Néant
14	tot_htva	Total hor TVA	N	14	$\sum$ mont_ign
15	tva	Taux sur la Valeur Ajoutée	N	2	Néant
16	mont_tva	Montant TVA	N	14	tot_htav x tva
17	Mont_ttc	Montant toutes taxes comprises	N	14	tot_htva + mont_tava

## 2- le dictionnaire ou dictionnaire épuré

N°	Code	Désignation	Type	Taille	Règle de calcul
01	nom_soc	Nom de la société	AN	50	Néant
02	tel_soc	Téléphone de la société	N	15	Néant
03	date_com	Date de la commande	D		
04	num_bon	Numéro du bon	AN	20	Néant
05	nom_client	Nom du client	AN	50	Néant
06	pren_client	Prénoms du client	AN	50	Néant
07	adres_cli	Adresse du client	AN	50	Code postal + Boîte postal + ville
08	tel_cli	Téléphone client	N	15	Néant
09	pay_cli	Pays du clieny	AN	50	Néant
10	design_prod	Désignation produit	AN	50	Néant
11	pu	Prix unitaire	N	14	Néant
12	qtte_cmd	Quantité commandée	N	14	Néant
13	mont_lign	Montant de la ligne de commande	N	14	Néant
14	tot_htva	Total hor TVA	N	14	$\sum$ mont_ign
15	tva	Taux sur la Valeur Ajoutée	N	2	Néant
16	mont_tva	Montant TVA	N	14	tot_htav x tva
17	Mont_ttc	Montant toutes taxes comprises	N	14	tot_htva + mont_tava

Nous supprimons toutes les propriétés calculées si les propriétés qui dans son calcul existes sinon elle est remplacée par ces dernières qui n'existent pas.

**Exemple :** tot\_htva est à supprimer par contre adres\_cli sera remplacée par code\_postal, boite\_postal, et ville.

Nous constatons que les propriétés ou la propriété pouvant identifiées le client et le produit n'existent pas. Il faut donc proposer les propriétés.

**Exemple** : pour client nous proposons code\_cli et pour produit ref\_prod  
Notre dictionnaire devient :

N°	Code	Désignation	Type	Taille	Règle de calcul
01	nom_soc	Nom de la société	AN	50	Néant
02	tel_soc	Téléphone de la société	N	15	Néant
03	date_com	Date de la commande	D		
04	num_bon	Numéro du bon	AN	20	Néant
05	nom_client	Nom du client	AN	50	Néant
06	pren_client	Prénoms du client	AN	50	Néant
07	code_cli	Code client	AN	20	Néant
08	tel_cli	Téléphone client	N	15	Néant
09	pay_cli	Pays du clieny	AN	50	Néant
10	design_prod	Désignation produit	AN	50	Néant
11	pu	Prix unitaire	N	14	Néant
12	qtte_cmd	Quantité commandée	N	14	Néant
13	mont_lign	Montant de la ligne de commande	N	14	Néant
14	code_p	Code postal	N	2	Néant
15	tva	Taux sur la Valeur Ajoutée	N	2	Néant
16	boite_p	Boîte postale	N	4	néant
17	ville	Ville	AN	50	Néant
18	ref_prod	Référence produit	AN	20	Néant

La seconde étape réside dans la recherche des dépendances fonctionnelles entre les propriétés recensées à la première étape. Pour mener de façon méthodique ce travail, on construit une matrice des dépendances fonctionnelles admettant une ligne et une colonne par propriétés du dictionnaire.

Un « 1 » placé à l'intersection de la ligne i et de la colonne j indique la présence d'une dépendance fonctionnelle entre la propriété Pj et Pi ( $P_j \rightarrow P_i$ ).

N°	Code	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	
01	nom_soc	1			1															
02	tel_soc	1	1		1															
03	date_com			1																
04	num_bon				1															
05	nom_client				1	1														
06	pren_client				1		1													
07	code_cli				1			1												
08	tel_cli				1			1	1											
09	pay_cli				1			1		1										
10	design_prod										1								1	
11	pu											1							1	
12	qtte_cmd												1							
13	mont_lign													1						
14	code_p				1			1							1					
15	tva															1				
16	boite_p				1			1									1			
17	ville				1			1										1		
18	ref_prod																		1	

Nous allons maintenant dépouiller la matrice de dépendance par supprimer les dépendances fonctionnelles réflexives et transitives. Finalement nous obtenons la matrice de dépendances fonctionnelles finale.

N°	Code	01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	
01	nom_soc				1															
02	tel_soc	1																		
03	date_com																			
04	num_bon																			
05	nom_client																			
06	pren_client																			
07	code_cli				1															
08	tel_cli							1												
09	pay_cli							1												
10	design_prod																		1	
11	pu																		1	
12	qtte_cmd																			
13	mont_lign																			
14	code_p							1												
15	tva																			
16	boite_p							1												
17	ville							1												
18	ref_prod																			

Cette phase met en évidence deux types de propriétés :

1. les propriétés identifiantes repérées dans la matrice par des colonnes qui comportent au moins un « 1 ». Dans l'exemple ci-dessus les propriétés 01, 04, 07 et 18 sont sources de dépendances fonctionnelles et joueront donc le rôle d'identifiant d'entités dans le modèle conceptuel.
2. les propriétés qui ne sont destinées d'aucune dépendance fonctionnelle et qui ont donc, dans la matrice des dépendances fonctionnelles, leur ligne vide. Dans l'exemple ci-dessus, les propriétés tva et qtte\_cmd ont ces caractéristiques. Parmi ces propriétés il convient alors de distinguer :
  - a. les propriétés paramètres telles que le taux de TVA ;
  - b. les autres propriétés pour lesquelles on doit rechercher les dépendances fonctionnelles ayant des sources multi-attributs qui permettent de les atteindre.

La source de ces dépendances sera constituée d'un sous-ensemble des rubriques identifiantes repérées à l'étape précédente. Ainsi, la propriété qttecmd, est déterminée à partir d'un num\_bon et d'une ref\_prod ce qui revient à écrire la dépendance fonctionnelle suivante :

Ref\_prod, num\_bon -> qttecmd.

Si l'on désigne par F l'ensemble initial des dépendances fonctionnelles, l'ensemble obtenu, noté F<sup>+</sup>, après élimination des dépendances parasites est appelé couverture minimale de F et peut être obtenu par application d'un algorithme. C'est encore un procédé algorithmique qui permet d'aboutir, à partir de la matrice dépouillée de toute dépendance inutile, au modèle conceptuel des données. Ainsi toute propriété identifiante donne naissance à une entité dont le contenu sera formé des propriétés avec lesquelles elle est en dépendance.

client			
code_cli	<pi>	<Indéfini>	<O>
nom_client		<Indéfini>	
pren_client		<Indéfini>	
tel_cli		<Indéfini>	
pays_cli		<Indéfini>	
code_p		<Indéfini>	
boite_p		<Indéfini>	
ville		<Indéfini>	
Identifiant_1	<pi>		

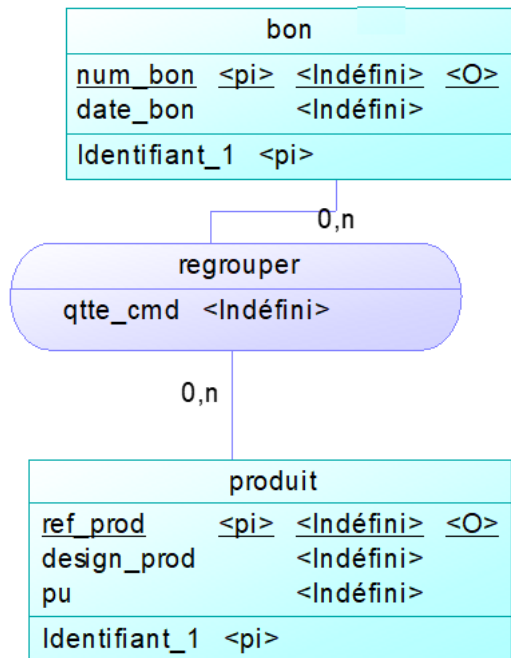
bon			
num_bon	<pi>	<Indéfini>	<O>
date_bon		<Indéfini>	
Identifiant_1	<pi>		

societe			
nom_soc	<pi>	<Indéfini>	<O>
tel_soc		<Indéfini>	
Identifiant_1	<pi>		

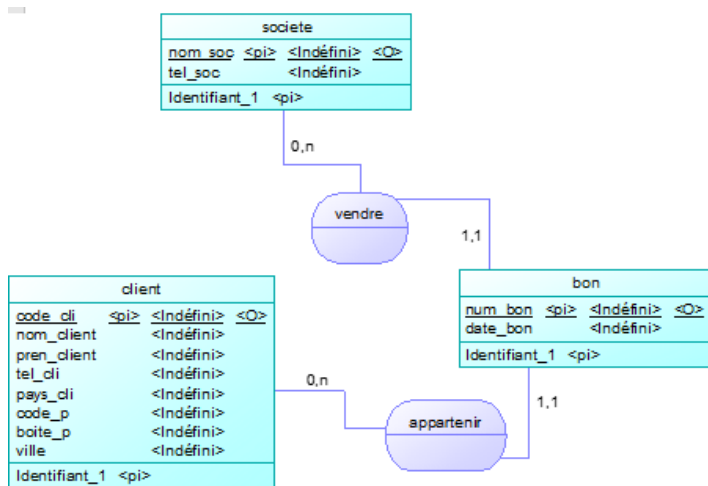
produit			
ref_prod	<pi>	<Indéfini>	<O>
design_prod		<Indéfini>	
pu		<Indéfini>	
Identifiant_1	<pi>		



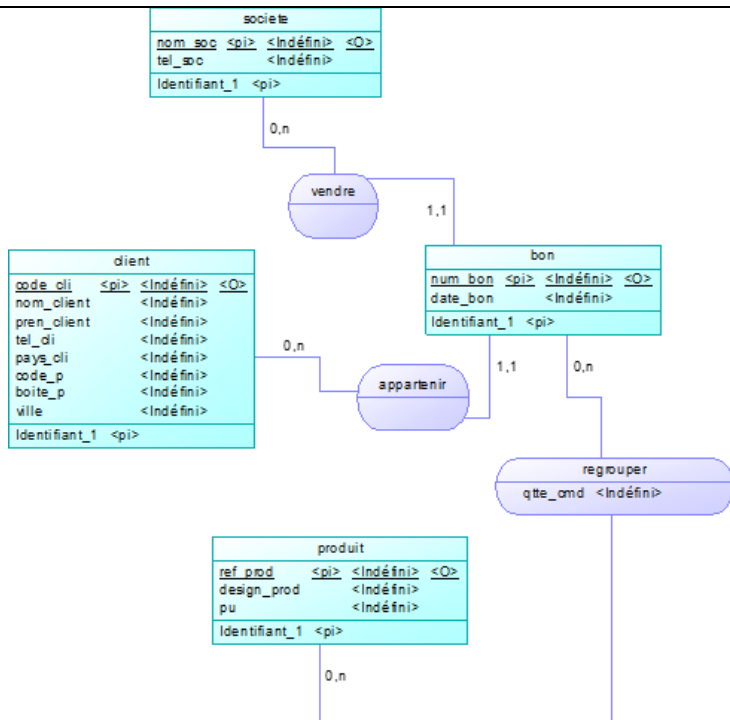
Les propriétés atteintes par des dépendances fonctionnelles multi-attributs seront intégrées à des associations porteuses (m-n) reliant les entités dont les identifiants sont spécifiés dans la source.



Enfin, les dépendances entre identifiants se matérialiseront par la présence d'une association (1-n). Le modèle ci-dessous résulte de l'application de ces différentes règles. Ce dernier devra alors être complété notamment au niveau des cardinalités minimales afin de prendre en compte toutes les règles de gestion.



Enfin



- Règles de vérification d'un MCD

#### A. Règles relatives aux entités

**Règle 1** : CHAQUE ENTITE DOIT POSSEDER UN IDENTIFIANT.

**Règle 2** : TOUTES LES PROPRIETES D'UNE ENTITE DOIVENT ETRE EN DEPENDANCE FONCTIONNELLE COMPLETE ET DIRECTE DEL'IDENTIFIANT.

**Rappel** :

- Dépendance fonctionnelle : une propriété B est dite en dépendance fonctionnelle d'une propriété A si pour toute valeur de A, il existe une et une seule valeur de B
- Dépendance complète : la dépendance fonctionnelle doit être de tout l'identifiant (toutes les propriétés qui le composent)
- Dépendance directe : chaque propriété doit dépendre directement de l'identifiant, et non pas par l'intermédiaire d'autre propriétés (par transitivité).

#### B. Règle relative aux associations

**Règle 3** : TOUTES LES PROPRIETES D'UNE ASSOCIATION DOIVENT DEPENDRE COMPLETEMENT DES IDENTIFIANTS DES ENTITES PARTICIPANT A L'ASSOCIATION

#### C. Règles générales relatives au MCD

**Règle 4** : UNE PROPRIETE NE PEUT APPARAITRE QU'UNE SEULE FOIS DANS UN MCD.

**Règle 5** : LES PROPRIETES RESULTANT D'UN CALCUL NE DOIVENT PAS APPARAITRE (sauf si elles sont indispensables à la compréhension)

**Règle 6** : LES PROPRIETES DOIVENT ETRE STABLES (leur valeur ne pas dépendre du temps : au lieu de l'âge, on conservera plutôt la date de naissance)

○ situations d'apprentissages/activités locales  
soit un système d'information suivant :

**Nom société : SOGECM**

***Le 12/09/2008***

**Tél : 50-30-12-56**

**BON DE COMMANDE N° : 1209200501**

**Doit :**

**Nom client : ZONGO Mahamoudou**

**Adresse : 01 BP 1245 Ouaga 01**

**Tél : 50-31-12-56**

**Pays : Burkina Faso**

**Détail du Bon de commande**

<b>Désignation</b>	<b>PU</b>	<b>Qtité</b>	<b>Montant</b>
<b>Carton de sucre sosuco</b>	<b>25 000</b>	<b>5</b>	<b>125 000</b>
<b>Sacs de riz</b>	<b>14 000</b>	<b>10</b>	<b>140 000</b>
<b>Carton de spaghetti</b>	<b>3 000</b>	<b>10</b>	<b>30 000</b>
<b>Total HTVA</b>			<b>295 000</b>
<b>TVA 18%</b>			<b>53 100</b>
<b>Total TTC</b>			<b>348 100</b>

**Nom société : SOGECM**

**Le 20/09/2008**

**Tél : 50-30-12-56**

**BORDEREAU DE LIVRAISON N° : 2209200501**

**Doit :**

**Nom client : ZONGO Mahamoudou**

**Adresse : 01 BP 1245 Ouaga 01**

**Tél : 50-31-12-56**

**Pays : Burkina Faso**

**Détail du Bon de livraison**

Désignation	Qtté cmdée	Qtté livrée
Carton de sucre sosuco	5	5
Sacs de riz	10	10
Carton de spaghetti	10	10

Date et signature du livreur

Date et Signature du client

**Règle de gestion**

**R1** : un bordereau de livraison appartient à une et une seule commande.

**R2** : une commande peut être livrée 0 ou 1 une fois.

**Travail à faire**

**1- Etablir le Dictionnaire données brutes**

**2- Etablir le dictionnaire de données épurées**

**3- Etablir la Matrice des dépendances fonctionnelles**

**4- Déduire le MCD**

- Unité d'apprentissage 4 (OS4) : générer le MLD à partir du MCD

- Contenu

Après avoir conçu le Modèle Conceptuel de Donnée (MCD), il est maintenant temps de le transposer en Modèle Logique de Données Relationnelles (MLDR). Ce MLDR est en fait le dernier pas vers le Modèle Physique de donnée(MPD), c'est à dire la description de la base qui va être créée

- Une entité se transforme en une relation (table)

Toute entité du MCD devient une relation du MLDR, et donc une table de la Base de Donnée. Chaque propriété de l'entité devient un attribut de cette relation, et dont une colonne de la table correspondante. L'identifiant de l'entité devient la Clé Primaire de la relation (elle est donc soulignée), et donc la Clé Primaire de la table correspondante.

CLIENT
<u>num_cli</u>
nom_cli
tel_cli

Devient

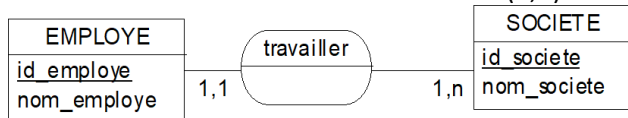
CLIENT(num\_cli, nom\_cli, tel\_cli)

Ou

CLIENT
<u>num_cli</u>
nom_cli
tel_cli

- Relation binaire aux cardinalités (X,1) - (X,n), X=0 ou X=1

La Clé Primaire de la table à la cardinalité (X,n) devient une Clé Etrangère dans la table à la cardinalité (X,1) :

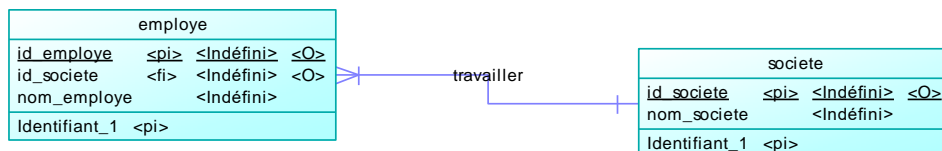


Modèle Logique de Donnée Relationnelle (MLDR) :

EMPLOYEE (id\_Employe, Nom\_Employe, #id\_Societe)

SOCIETE (id\_Societe, Nom\_Societe)

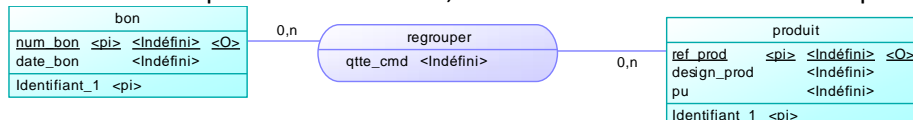
ou



- Relation binaire aux cardinalités (X,n) - (X,n), X=0 ou X=1

Il y a création d'une table supplémentaire ayant comme Clé Primaire une clé composée des identifiants des 2 entités. On dit que la Clé Primaire de la nouvelle table est la concaténation des Clés Primaires des deux autres tables.

Si la relation est porteuse de donnée, celles ci deviennent des attributs pour la nouvelle table.



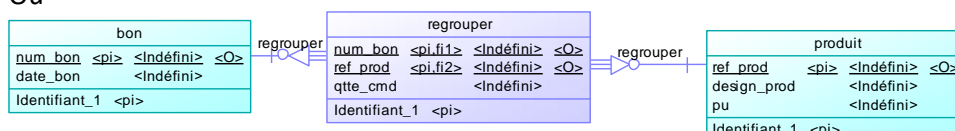
MLDR :

bon (num\_bon, date\_bon)

PRODUIT (ref\_prod, design\_prod)

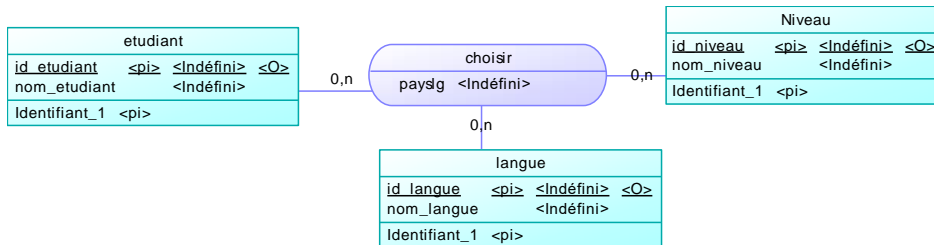
regrouper (#num\_bon, #ref\_prod, qtte\_cmd)

Ou



- Relation n-aire (quelles que soient les cardinalités).

Il y a création d'une table supplémentaire ayant comme Clé Primaire la concaténation des identifiants des entités participant à la relation. Si la relation est porteuse de propriétés, celles ci deviennent des attributs pour la nouvelle table.



MLDR :

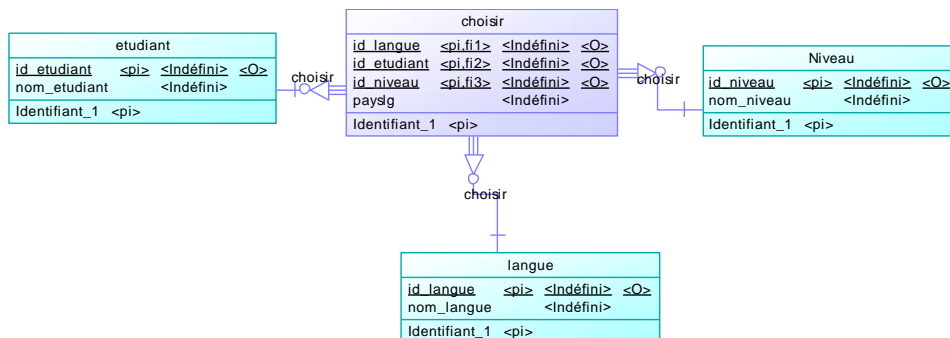
ETUDIANT (id\_Etudiant, Nom\_Etudiant)

NIVEAU (id\_Niveau, Nom\_Niveau)

LANGUE (id\_Langue, Nom\_Langue)

PARLE (#id\_Etudiant, #id\_Niveau, #id\_Langue, payslg)

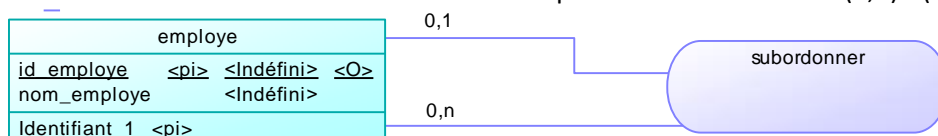
Ou



- Association Réflexive

- Premier cas : cardinalité (X,1) - (X,n), avec X=0 ou X=1.

La Clé Primaire de l'entité se dédouble et devient une Clé Etrangère dans la relation ou nouvelle table. Exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,1) - (X,n) (Cf la relation binaire).

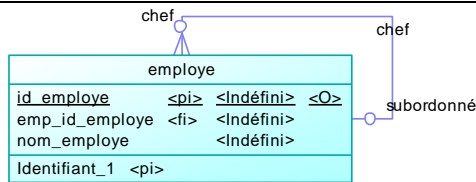


MLDR :

EMPLOYE (id\_Employe, Nom\_Employe, #emp\_id\_employe)

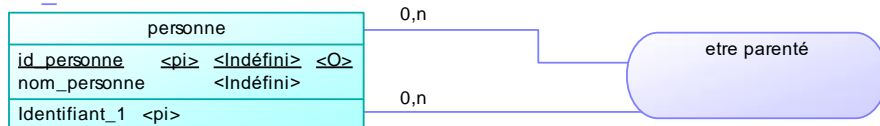
#emp\_id\_employe est l'identifiant (id\_Employe) du supérieur hiérarchique direct de l'employé considéré

Ou



Deuxième cas : cardinalité (X,n) - (X,n), avec X=0 ou X=1.

De même, tout se passe exactement comme si l'entité se dédoublait et était reliée par une relation binaire (X,n) - (X,n) . Il y a donc création d'une nouvelle table.



MLDR :

PERSONNE (id\_Personne, Nom\_Personne)

PARENTE (#id\_Parent, #id\_Enfant)

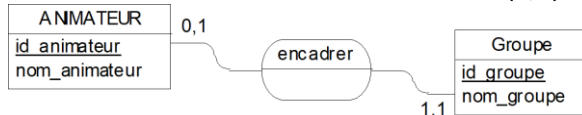
#id\_Parent est l'identifiant (id\_Personne) d'un ascendant direct de la personne. #id\_Enfant est l'identifiant (id\_Personne) d'un descendant direct de la personne.

La table PARENTE sera en fait l'ensemble des couples (parents-enfants) présent dans cette famille

ou

- Relation binaire aux cardinalités (0,1) - (1,1).

La Clé Primaire de la table à la cardinalité (0,1) devient une Clé Etrangère dans la table à la cardinalité (1,1) :

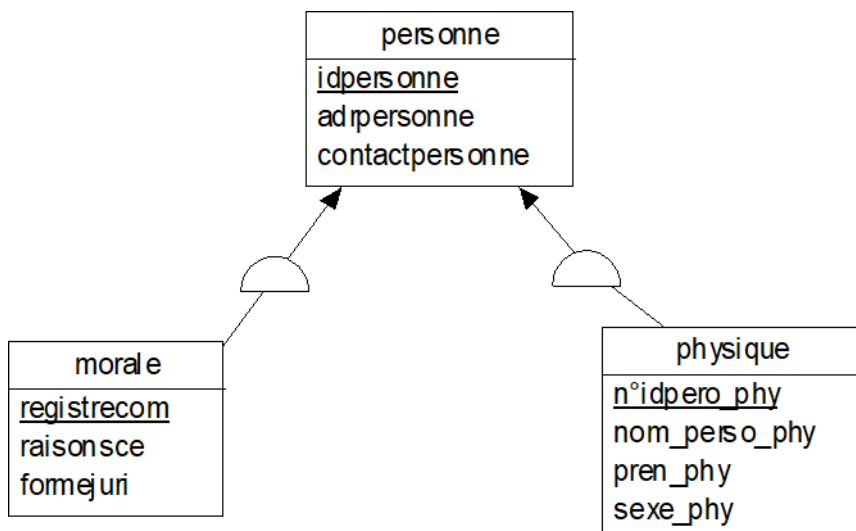


MLDR :

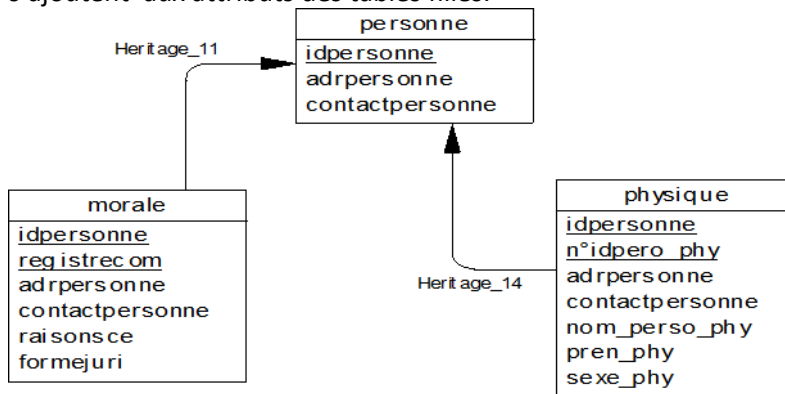
ANIMATEUR (id\_Animateur, Nom\_Animateur)

GROUPE (id\_Groupe, Nom\_Groupe, #id\_animateur)

- Héritage



L'identifiant de la table mère devient (ou s'ajoute à) la clé primaire des tables filles puis les propriétés de la table mère s'ajoutent aux attributs des tables filles.



- situations d'apprentissages/activités locales

#### ■ Unité d'apprentissage 5 (OS5) : Appliquer les notions de SQL pour générer une Base de données dans un SGBD

- Contenu

Cette partie permettra à l'apprenant de maîtriser et appliquer les différents langages appliquer à un SGBD.

- Rappel des types

Les types de données manipulés par les LMD sont généralement :

INT, SMALLINT entier, entier court

DEC(m,n) -- nombre décimal de m chiffres dont n après la virgule

FLOAT(n) -- nombre flottant

CHAR(n) -- chaîne de longueur n

VARCHAR(n) -- chaîne variable

DATE 'yyyymmdd'

TIME 'hhmmss'

TIMESTAMP 'yyyymmddhhmmssnnnnnn'

- Langage de définition des données (LDD)
  - Création d'une base de données

#### Syntaxe

create database <nom\_base\_donnees>;

#### Exemple

create database bdcommande;

- Création d'une table

#### Syntaxe :

Création d'une table : create table <nom\_table>

( <nom\_chp1> <type\_donnee1> ,

<nom\_chp2> <type\_donnee2> ,

...

Primary key (<champ\_identifiant>));

#### Exemple : create table personne(

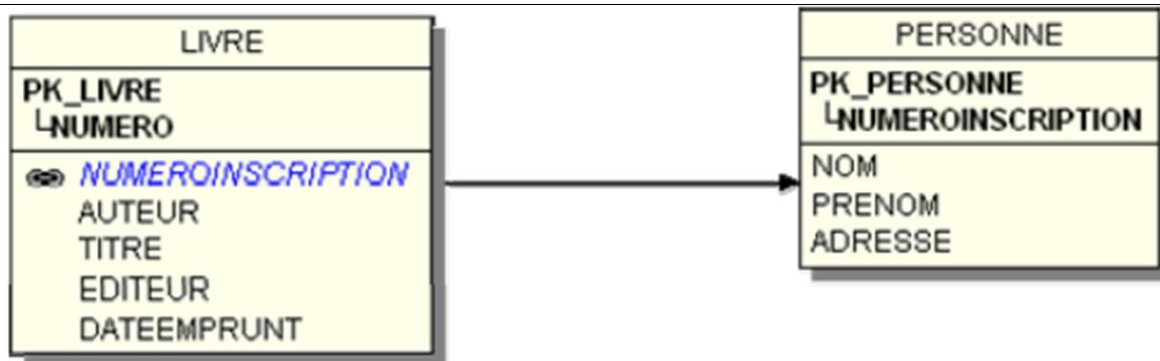
idpers int(11),

nom\_pers varchar(50),

age int(4),

Primary key (idpers));





#### Création de la table Livre avec clés primaire et étrangère

```

CREATE TABLE LIVRE
(
  NUMERO NUMBER(4) NOT NULL,
  NUMEROINSCRIPTION CHAR(32) NULL,
  AUTEUR CHAR(32) NULL,
  TITRE CHAR(32) NULL,
  EDITEUR CHAR(32) NULL,
  DATEEMPRUNT Date NULL,
  PRIMARY KEY (NUMERO),
  FOREIGN KEY (NUMEROINSCRIPTION) REFERENCES PERSONNE (NUMEROINSCRIPTION),
);
  
```

#### Modification de la structure d'une table

ALTER TABLE T\_CLIENT ADD CLI\_PRENOM VARCHAR(25) ; // ajout d'un champ

ALTER TABLE T\_CLIENT DROP RAISON\_SOC; // suppression d'un champ

ALTER TABLE messages MODIFY contenu LONGTEXT NULL; //modification du type d'un champ (type taille ....)

##### Langage de manipulation des données(LMD)

Ajouter un enregistrement : insert into <nom\_table>(<chp1>, <chp2>,...) values (<val1>, <val2>,...);

Exple : **insert into personne (idpers, nompers) values (1, "KINDO Alice");**

Modifier un enregistrement : update <nom\_table> set <chp1>=<val1>, <chp2>=<val2>, ... where condition;

Exple update personne set nompers="KINDO Pinda Alice" where idpers=1;

Supprimer un ou des enregistrement : delete from <nom\_table> where condition;

Exple : delete from personne where idpers=1;

##### Langage d'Interrogation des données (LID)

Projection (sélectionner les champs à afficher) : select <chp1>, <chp2>,... from <nom\_table>

Exple : lister le nom des personnes : select nompers from personne;

Projection avec calcul : select <chp1>, <chp2><op><val>/<chpqcq>,...from <nom\_table>

Où <op>= +, -, \*, /

Exple : select nompers, age+5 from personne;

Nb : select \* liste toutes les rubriques

Sélection permet de sélectionner des données respectant certaines conditions. Select <liste\_champ> from <nom\_table> where <condition>

Lister les personnes dont l'âge est compris entre 20 à 35 : select \* from personne where age between 20 and 35;

Lister la personne la plus âgée : select \* from personne where age=(select max(age) from personne);

Produit cartésien :  $A \times B = \{x \in A, y \in B \text{ et } (x,y) \in A \times B\}$  card(AxB)=cardA x cardB : la jointure sans condition : select \* from <table1>, <table2>...

Pas sens.

Jointure : produit cartésien avec condition : select \* from <table1>, <table2>, ... where

<table1>.<chpcm12>=<table2>.<chpcm12> and <table2>.<chpcm23>=...;

Exemple : lister les enfants suivis de leur père : `select * from personne as enfant, personne as pere where enfant.idperspere=pere.idpers;`

Exple : lister les personnes âgées de 12 ans : `select * from personne where age=12`

Lister les personnes dont l'âge est compris entre 20 à 35 : `select * from personne where age between 20 and 35;`

Utilisation d'une expression sql comme scalaire

Lister la personne la plus âgée : `select * from personne where age=(select max(age) from personne);`

Quelques fonctions

-AVG() : moyenne;

-max(): maximum;

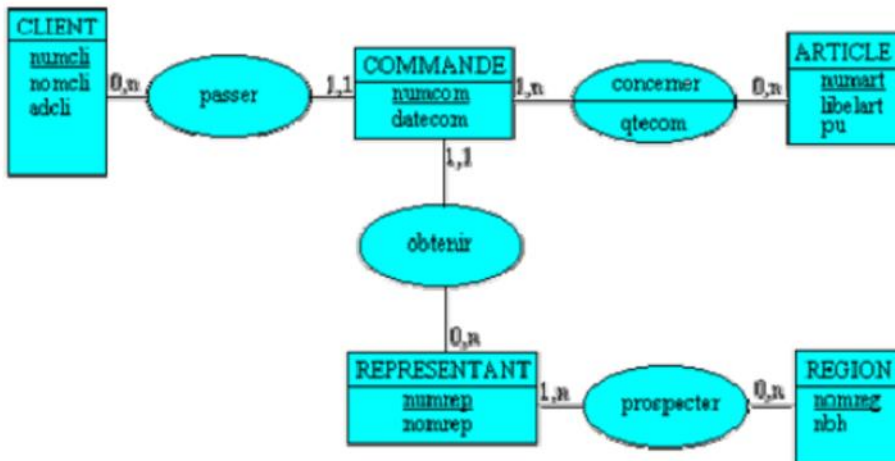
-min() : minimum;

-count() : compte le nombre d'enregistrement;

- situations d'apprentissages/activités locales

1- Transformer ce modèle en MLD

2- Créer les tables correspondantes



3- SQL

a- Lister les clients de la base.

b- lister les clients à l'adresse "Rue Kwamé Kruma Ouaga"

c- lister N°, Nom des clients à l'adresse "Rue 09 Ouagadougou"

d- lister les commandes de la date 25/01/2016

e- lister les produits donc le prix unitaire est supérieur 2000

f- liste les produits dont le prix unitaire est compris entre 1000 et 4000.

e- Quel est le produit le plus cher ?

g- lister les lignes (numart,libart,pu,qtecom,pu\*qtecom) de la commande dont le numcom=1

i- quelle est la recette engendrée par le représentant 1 en 2015

h- quels sont les représentants de la région du centre

k- quelle est la region la plus peuplées ?

l- quels sont les representants du client "KABORE Ali"?

client

Numcli	nomcli	adcli
1	KABORE Ali	Rue Kwamé N'Kruma

2	SANOU Ali	Rue Kwamé N’Kruma
3	BARRO Siaka	Rue Kwamé N’Kruma
4	BARRA Fati	Rue Kwamé N’Kruma
5	BAZIE Priska	Rue 09 Ouagadougou
6	KONE Madou	Rue 09 Ouagadougou
7	PALM Sié	Rue 09 Ouagadougou
8	DAO Bakary	Rue 06 Bobo

commande

Numcom	Datecom	numcli	numrep
1	25/01/2016	1	1
2	25/01/2016	1	2
3	01/04/2015	2	1
4	10/05/2015	3	2
5	14/04/2015	6	3
6	16/06/2015	7	1

Article

Numart	Libart	pu
1	Boite de sardine	350
2	Boite de lait non sucré	950

3	Ketchup	2100
4	Boite Mayonnaise calvé	2600
5	Paquet biscuit chat	1300
6	Bidon d’huile 5l AYA	4250

produitcommande

Numart	numcom	qtecom
1	1	4
2	1	5
5	1	6
2	3	4
5	3	8
6	3	9
1	5	4
3	5	5
4	5	8
5	6	4
6	6	3

Representant

Numrep	nomrep
--------	--------

1	DRABO Ali
2	TRAORE Karim
3	SANA Fati
4	CONGO Ariane

Region

Nomreg	nbh
Centre	1000500
Cascade	300657
Houet	896000
Est	207908

regionrepresente

Nomreg	numrep
Centre	1
Cascade	2
Houet	3
Est	1

○ Références

Guézélou, P. (2006). Modélisation des données : Approche pour la conception des bases des données.  
Hernandez, M. J. & Viescas, J. L. (2001). Introduction aux requêtes SQL. Eyrolles.  
Kauman, J., Matsik, B. & Spencer, K. (2001). Maîtrisez SQL (Wrox Press, Ed.). CampusPress.  
Labbé, C. (2002). Modéliser les données. Pratikio.  
Petrucci, L. (2006). Base de données. Présentation projetée et travaux dirigés. (IUT GTR Villetaneuse)  
Saglio, J.-M. (2002). Dominante informatique : Module bases de données.  
SQL Anywhere Studio. (2005a). ASA - Guide de programmation : Programmation avec Embedded  
SQL Anywhere Studio. (2005b). ASA - Guide de programmation : Utilisation de SQL dans les applications.

Szulman, S. (2005). Base de données et SGBD. Présentation projetée.