

Modélisation des données

Table des matières

Description générale	3
Définition 1	3
Définition 2	3
MERISE	4
Étapes de la conception	5
Réel	6
Modèle conceptuel	6
Modèle organisationnel	6
Modèle logique	6
Modèle physique	6
Recueil des informations	7
La modélisation conceptuelle et organisationnelle des données	9
Objet (ou entité) :(relation) association et propriétés	9
Référence	10
Code_dépôt	10
Concept de propriété:	10
Identifiant	11
Code_service	12
Code_élève	12
Code_classe	12
Code_prof	12
Référence	13
Num_Clt	13
Date	13
Num_pers	13
Règles de vérification et de normalisation	16
Méthodologie	17
Modèle conceptuel des traitements	25
I) règles de logique:.....	25
II) Traitement.....	30
Passage du formalisme entités/relations au formalisme Enregistrement/ensemble	33
Passage du modèle conceptuel des données au modèle relationnel	34
SGDB relationnel	35
Passage du modèle conceptuel des données au modèle relationnel	35

Description générale

Il est difficile de donner une définition exacte de la notion de base de données. Une définition très générale pourrait être :

Définition 1

-Base de données- Un ensemble organisé d'informations avec un objectif commun.

Peu importe le support utilisé pour rassembler et stocker les données (papier, fichiers, etc.), dès lors que des données sont rassemblées et stockées d'une manière organisée dans un but spécifique, on parle de base de données.

Plus précisément, on appelle base de données un ensemble structuré et organisé permettant le stockage de grandes quantités d'informations afin d'en faciliter l'exploitation (ajout, mise à jour, recherche de données). Bien entendu, dans le cadre de ce cours, nous nous intéressons aux bases de données informatisées.

Définition 2

-**Base de données informatisée**- Une base de données informatisée est un ensemble structuré de données enregistrées sur des supports accessibles par l'ordinateur, représentant des informations du monde réel et pouvant être interrogées et mises à jour par une communauté d'utilisateurs.

Le résultat de la conception d'une base de données informatisée est une description des données. Par description on entend *définir les propriétés d'ensembles d'objets modélisés* dans la base de données et non pas d'objets particuliers. Les objets particuliers sont créés par des programmes d'applications ou des langages de manipulation lors des insertions et des mises à jour des données.

Cette description des données est réalisée en utilisant un modèle de données. Ce dernier est un outil formel utilisé pour comprendre l'organisation logique des données.

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le **Système de gestion de base de données (SGBD)**. Un SGBD est caractérisé par le modèle de description des données qu'il supporte (hiérarchique, réseau, relationnel, objet : cf. section 1.1.2). Les données sont décrites sous la forme de ce modèle, grâce à un Langage de Description des Données (LDD). Cette description est appelée schéma.

Une fois la base de données spécifiée, on peut y insérer des données, les récupérer, les modifier et les détruire. C'est ce qu'on appelle manipuler les données. Les données peuvent être manipulées non seulement par un Langage spécifique de Manipulation des Données (LMD) mais aussi par des langages de programmation classiques.

Ref:

<http://merise.developpez.com/faq/?page=MCD>
<http://philippe.guezlou.free.fr/mcd/mcd.htm>

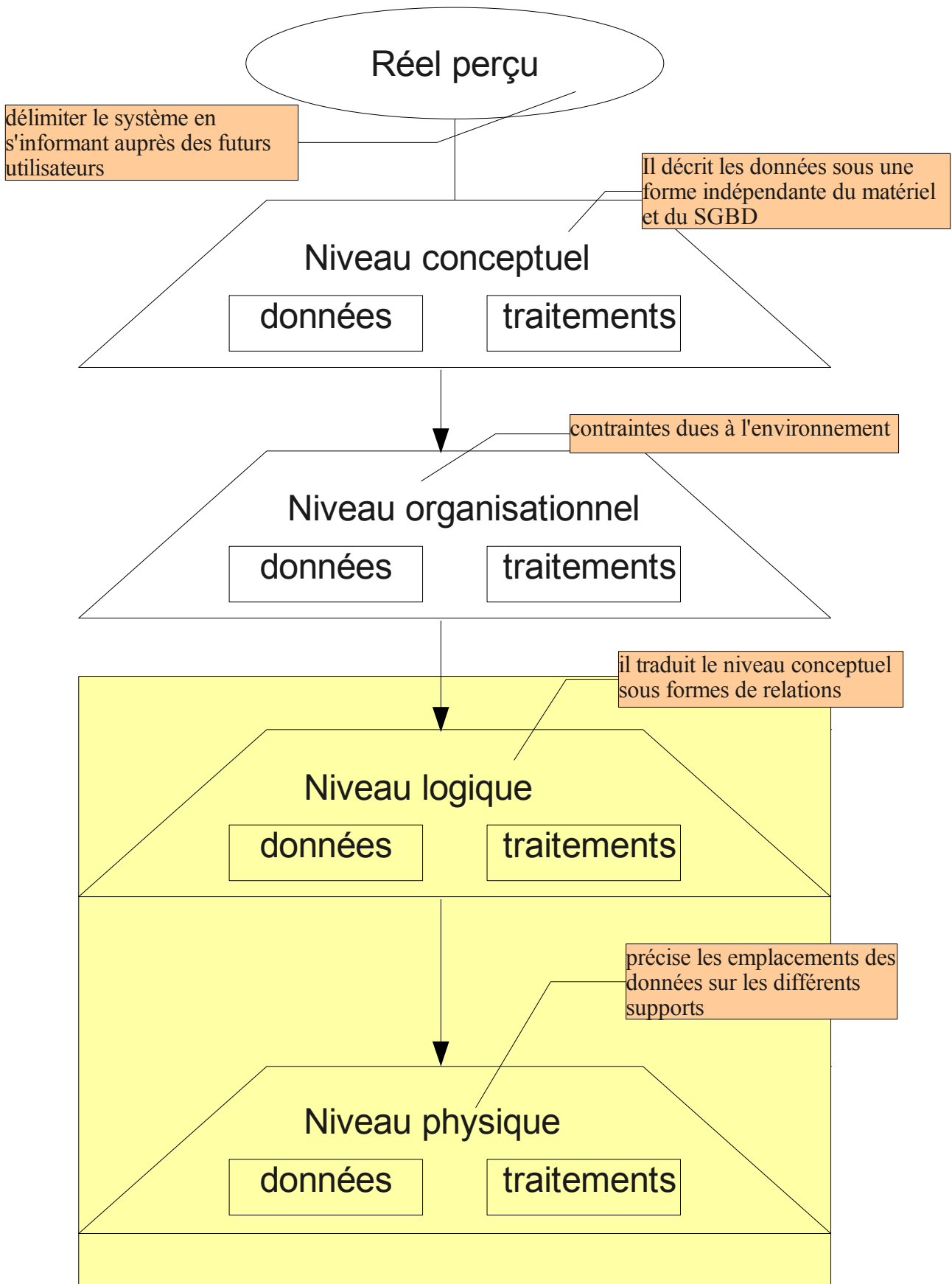
MERISE

MERISE (**M**éthode d'**É**tude et de **R**éalisation Informatique pour les **S**ystèmes d'**E**ntreprise) est certainement le langage de spécification le plus répandu dans la communauté de l'informatique des systèmes d'information, et plus particulièrement dans le domaine des bases de données.

Une représentation Merise permet de valider des choix par rapport aux objectifs, de quantifier les solutions retenues, de mettre en œuvre des techniques d'optimisation et enfin de guider jusqu'à l'implémentation. Reconnu comme standard, Merise devient un outil de communication. En effet, Merise réussit le compromis difficile entre le souci d'une modélisation précise et formelle, et la capacité d'offrir un outil et un moyen de communication accessible aux non-informaticiens.

Merise propose une démarche, dite **par niveaux**, dans laquelle il s'agit de hiérarchiser les préoccupations de modélisation : la conception, l'organisation et la technique. En effet, pour aborder la modélisation d'un système, *il convient de l'analyser en premier lieu de façon globale et de se concentrer sur sa fonction : c'est-à-dire de s'interroger sur ce qu'il fait avant de définir comment il le fait.*

Étapes de la conception



Réel

L'expression des besoins est une étape consistant à définir ce que l'on attend du système d'information automatisé, il faut pour cela :

- faire l'inventaire des éléments nécessaires au système d'information
- délimiter le système en s'informant auprès des futurs utilisateurs
- définir les flux d'informations à prendre en compte.

Modèle conceptuel

L'étape suivante consiste à mettre au point le **MCD** (Modèle conceptuel des données) et le **MCT** (Modèle conceptuel des traitements) décrivant les règles et les contraintes à prendre en compte.

Modèle organisationnel

Le modèle organisationnel consiste à définir le **MOT** (Modèle organisationnel des traitements) décrivant les contraintes dues à l'environnement (organisationnel, spatial et temporel).

Modèle logique

Le modèle logique représente un choix logiciel pour le système d'information.

Modèle physique

Le modèle physique reflète un choix matériel pour le système d'information.

Recueil des informations

La grille d'analyse est un document conçu pour recenser les différentes informations de différentes origines (chaque utilisateur a une vision personnelle du système d'information) et de différentes formes (écrite, orale ..) du système d'information. On collecte ces informations suite à des interviews, d'analyse de documents existants ou en projet ...

Ces informations doivent être analysées pour juger de leur pertinence pour l'organisation.

La grille d'analyse comprend :

- 1) La désignation des informations en précisant éventuellement le ou les documents comportant l'information.
- 2) Les règles de gestion concernant les informations calculées
- 3) Les informations déduites des informations recensées et des règles de gestion ce qui constitue le dictionnaire de données.

Exemple : Grille d'analyse

INFORMATIONS COLLECTEES	DOCUMENTS		REGLES DE GESTION	DICTIONNAIRE DE DONNEES	
	D1	D2		Définition	Code
Code facture	x		A1	Code facture	FactCode
Date facture	x			Date facture	FactDate
Code client	x	x		Code client	CliCode
Mtht produit	x				
Pu produit	x			Pu produit	ProdPu

Il faut obtenir toutes les données du système d'information en essayant de faire face aux 3 difficultés suivantes :

1. Recenser toutes les données utiles
2. Associer à chaque donnée un nom significatif (un libellé)
3. Analyser et résoudre les questions posées par les données composées (conserver une donnée composée est possible ; c'est un choix de gestion), les synonymes et les polysèmes, les données calculées, paramètres.

DONNEE COMPOSEE : Si on a une donnée composée de plusieurs données alors on éclate en plusieurs données élémentaires.

Exemple : L'identité du client composé du nom du client et du prénom du client donne lieu à 2 données : Nom du client, Prénom du client

SYNONYME : 2 données sont dites synonymes quand elles ont la même signification et des noms différents dans plusieurs documents, il n'y a donc qu'une donnée avec deux libellés.

Exemple : Numéro représentant et Code représentant

Taux de commission et pourcentage de commission

POLYSEME : 2 données sont dites polysèmes quand elles ont le même nom mais pas la même signification, deux libellés sont nécessaires.

Exemple: Sur un bon de commande on a une quantité de produit
Sur un état de stock on a quantité de produit qui n'a pas la même signification, dans ce cas cela correspond à la quantité en stock.

DONNEES CALCULEES : Ce sont des données obtenues par calcul à partir de données élémentaires. Il est important de retrouver la règle de gestion permettant l'obtention de la donnée calculée afin de s'assurer que l'on n'oublie pas de données élémentaires. En général les données calculées ne figurent pas dans le dictionnaire de données mais on peut aussi choisir de conserver une donnée calculée qui prendrait trop de temps à être calculé, exemple : SomDébit.

PARAMETRES : Les paramètres sont des propriétés dont la valeur peut être constante ou prévisible et dont l'utilité est ponctuelle comme par exemple la date du jour, le taux tva ...

La modélisation conceptuelle et organisationnelle des données.

Objet (ou entité) : (relation) association et propriétés.

Exemple:

- Un client Durant a passé la commande C1 contenant les produits P1 et P2.
- Le même Durant a passé la commande C2 contenant les produits P2 et P3.
- Le client Dupont a passé la commande C3 contenant les produits P1 et P2.
- Le même a passé la commande C4 contenant les produits P2 et P3

Présentons cela dans un tableau:

client	commande	produits	facture
Durant	C1	P1, P2,P3	F1
Durant	C2	P2, P3	F2
Dupont	C3	P1, P2	F3
Dupont	C4	P2, P3	F4

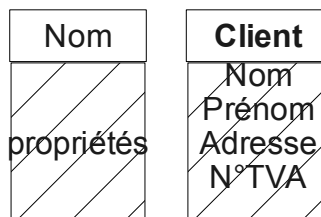
On voit 4 ensembles:

clients (2 constituants), commande (4 constituants) , produits (3) et factures (4)
Ces 4 ensembles seront appelés objets.

Un objet (ou entité)

*est pourvu d'une **existence propre** et conforme au choix de gestion de l'entreprise.*

Formalisme :rectangle contenant le nom et la liste des propriétés.



L'objet « client » désigne un client en général et, dans notre exemple, possède deux occurrences (2 clients)

Une entité peut être :

1. Une personne ou un individu : ELEVE, CLIENT...
2. Un objet concret : PRODUIT, OUTIL, MACHINE..
3. Un objet abstrait : COMPTE, ENSEIGNEMENT..
4. Un lieu : DEPÔT, ATELIER..
5. **Un objet documentaire : FACTURE, CONTRAT..**

Concept de relation – association

Il existe plusieurs manières d'associer les objets:

- Un client passe une commande
- Une commande contient plusieurs produits
- Une facture contient plusieurs produits et est liée à une commande

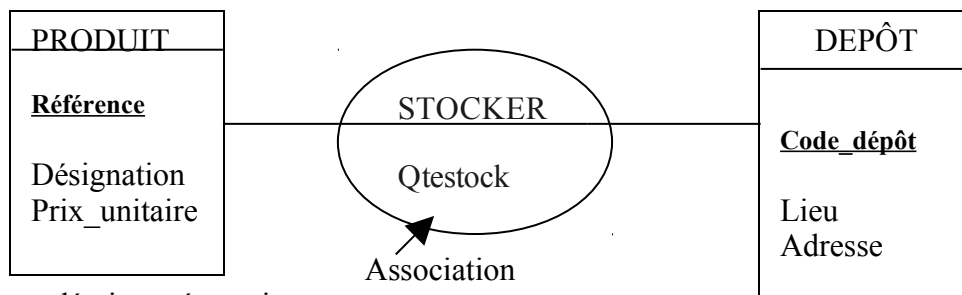
Une relation (Association) entre objets ou entités

est une association perçue dans le réel entre deux ou plusieurs objets. Une relation est dépourvue d'existence propre.

Formalisme



Exemple : Un produit est stocké dans un dépôt pour une certaine quantité :Qtestock.



Une association est décrite, nécessairement par :

1. Les entités qui jouent un rôle dans cette association,
2. Eventuellement par un ensemble de propriétés.

Concept de propriété:

Exemple: une personne emprunte des livres

L'objet **personne** possède pour chaque occurrence un

- numérode personne
- Nom
- Prénom
- Adresse

L'objet **livre** possède pour chaque occurrence :

- N° de livre
- titre
- éditeur

Pour la **relation emprunter**, on peut considérer la

- date d'emprunt

Entité CLIENT peut être caractérisée par :

Code_client
Raison_sociale
Rue
Ville
Code_postal.

Propriété

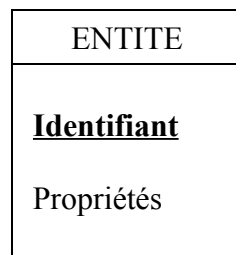
Une propriété est une donnée élémentaire appartenant à un objet ou une relation.
Formalisme: Le nom de la propriété est inscrit à l'intérieur de l'objet.

Identifiant

C'est une propriété particulière de l'objet telle qu'à chaque valeur de la propriété corresponde une et une seule occurrence de l'objet

L'identifiant d'une relation est obtenu par concaténation des identifiants des objets participant à la relation.

Formalisme :



Occurrence d'un objet

une occurrence est un élément individualisé appartenant à l'objet
exemple le client DUPONT

Occurrence d'une relation

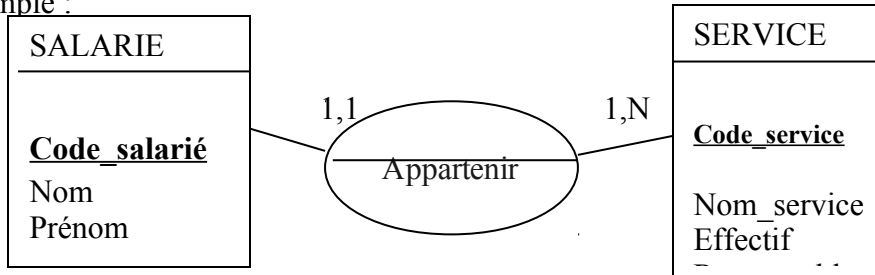
relation individualisée reliant une seule occurrence des objets participant à cette relation.

Cardinalité:

la cardinalité d'un objet s'exprime par rapport à une relation.

C'est le nombre minimum (maximum) d'occurrences de cet **objet** qui participent aux occurrence de la relation.

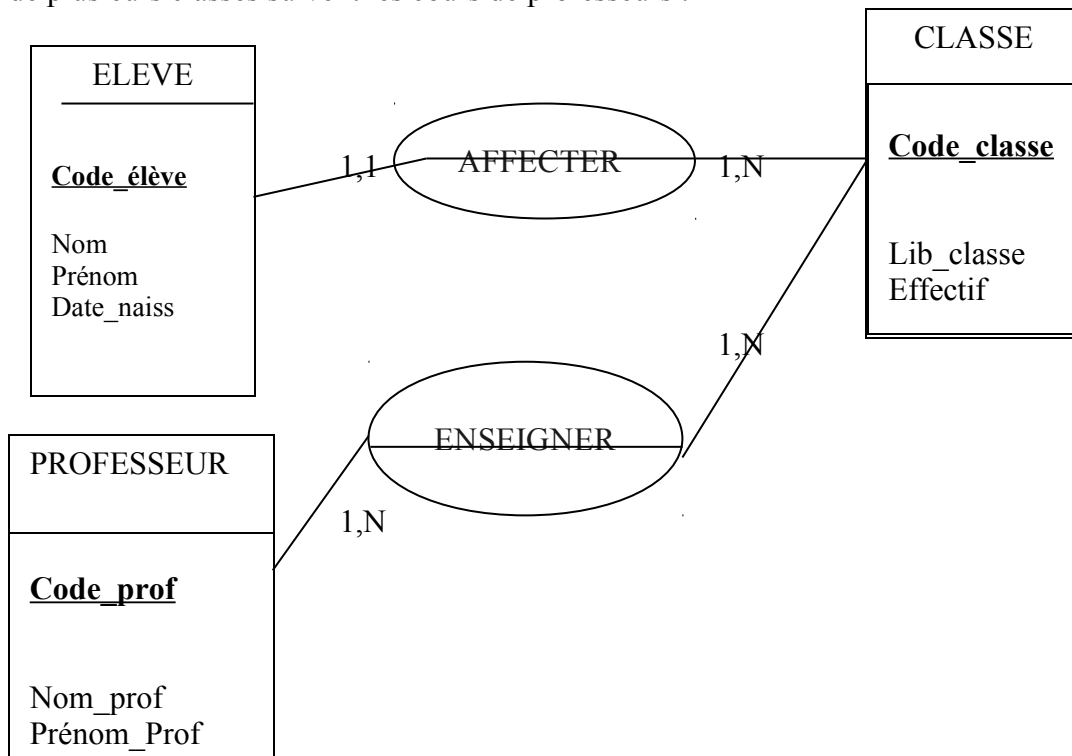
Exemple :



À partir de ces valeurs minimum et maximum possibles, il existe quatre types de cardinalité :

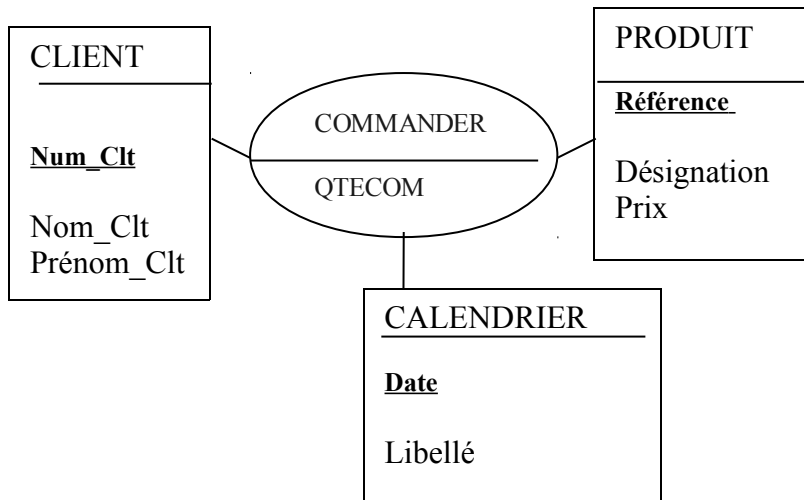
- (0,1) : une occurrence de l'objet n'est jamais liée plus d'une fois à l'association.
- (1,1) : une occurrence de l'objet est toujours liée une et une seule fois à l'association.
- (1,n) : une occurrence de l'objet est toujours liée au moins une fois à l'association.
- (0,n) : aucune précision donnée.

Les élèves de plusieurs classes suivent les cours de professeurs :

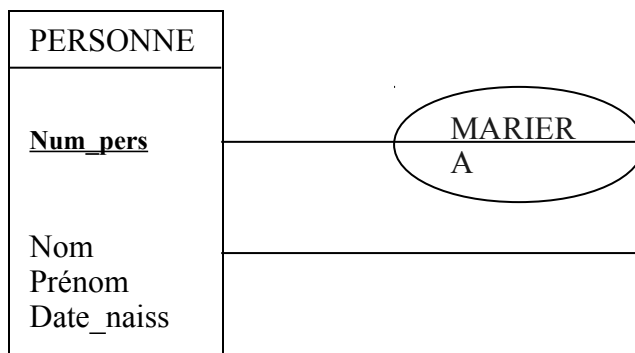


On appelle dimension de l'association, le nombre d'entités participant à celle-ci.

Exemple : Association de dimension 3 (Ternaire).



Une association peut être réflexive lorsqu'elle associe une entité à elle-même.



Règles de validation du MCD Entité-Association.

4. Toute propriété doit à terme prendre une valeur.
5. Une propriété ne doit pas figurer dans deux entités ou deux associations. De ce fait, on ne peut pas retrouver le même nom de propriété plusieurs fois dans un modèle de données.
6. Il n'est pas interdit qu'une propriété n'appartienne ni à une entité, ni à une association. Il s'agit alors d'une propriété paramètre.
7. Toute entité doit avoir un identifiant qui doit prendre une valeur unique pour toute occurrence.
8. Toute association possède un identifiant composé des identifiants des entités qui participent à cette association.

Application:

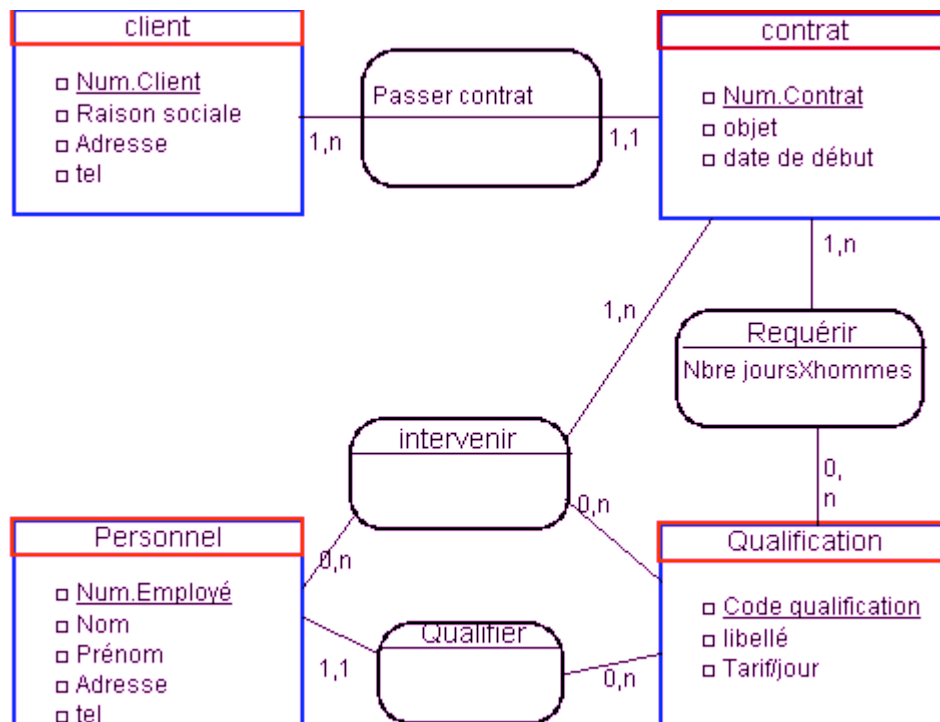
Une entreprise de petits travaux (plomberie, électricité, nettoyage...) est chargée de mettre des personnes à disposition de clients.

Chaque intervention donne lieu à un contrat avec le client.

Les principales informations du contrat sont:

- description succincte de l'intervention
- Date de début
- Qualification précise de chaque intervenant (une par intervenant et une vingtaine possibles)
- Le nombre de jours X hommes prévus.
- A chaque qualification correspond un tarif journalier.
- Chaque personne possède au moins une qualification.
- La qualification d'intervention est dans l'ensemble des qualifications standard.
- A chaque intervention, il est possible de réajuster la qualification d'intervention par rapport à la qualification de base.

Construire le modèle conceptuel de données.



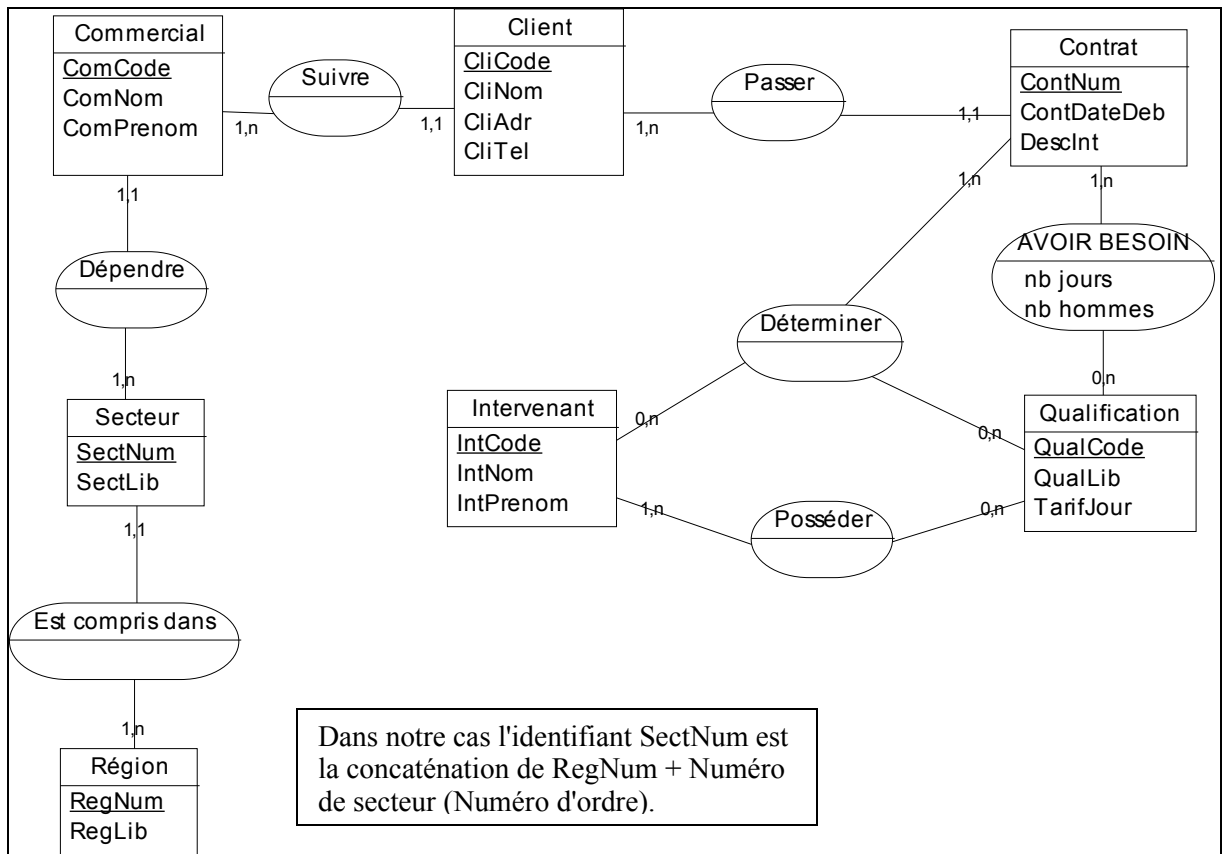
Variante du précédent :

Soit une PME spécialisée dans la mise à disposition de personnes pour le compte de ses clients. Chaque intervention donne lieu à un contrat avec le client passé par le commercial qui suit ce client. Ce commercial dépend d'un secteur d'une région. Le numéro de secteur est relatif au numéro de région : on a le numéro de secteur 1 pour la région Pays de Loire, on a aussi un numéro de secteur 1 pour la région Bretagne.

Les principales informations du contrat sont :

- la description succincte de l'intervention,
- la date de début de l'intervention,
- le nombre de jours * hommes prévus dans chaque qualification
- les personnes intervenant dans ce contrat.
- Chaque intervenant possède une à plusieurs qualifications
- A chaque qualification correspond un tarif journalier

La PME détermine en interne la qualification retenue pour l'intervenant pour un contrat donné.



Règles de vérification et de normalisation

Règles concernant les objets

- 1) Existence d'un identifiant pour chaque objet.
- 2) Toutes les propriétés autres que l'identifiant doivent être en dépendance fonctionnelle directe de l'identifiant.

Règles concernant les relations.

- 3) toute propriété d'une relation doit dépendre complètement de l'identifiant de la relation. De plus, chaque propriété doit dépendre complètement de l'identifiant et non d'une partie de celui-ci.
- 4) Une propriété ne peut apparaître qu'une seule fois dans un MCD.
- 5) Les propriétés qui résultent de calculs ne doivent pas figurer dans le MCD sauf pour faciliter la compréhension.
- 6) Normalisation des relations (importante) : les attributs des associations doivent dépendre des identifiants de toutes les entités en association.
Par exemple, la quantité commandée dépend à la fois du numéro de client et du numéro d'article, par contre la date de commande non.

Méthodologie

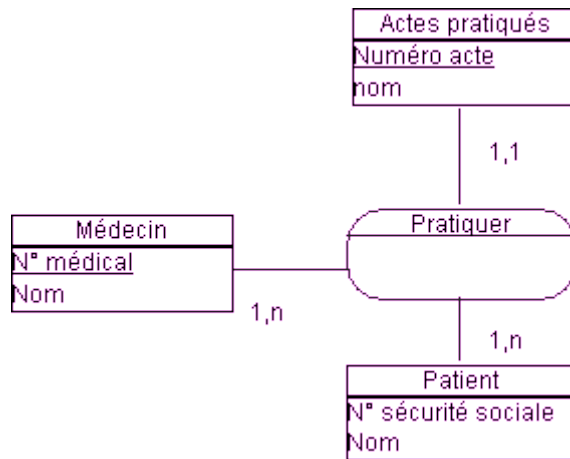
Face à un problème bien formulé (même si cela n'existe pas) procéder ainsi :

- identifier les entités en présence ;
- lister leurs attributs ;
- ajouter les identifiants ;
- établir les associations entre les entités ;
- lister leurs attributs ;
- éliminer les synonymes (plusieurs signifiants pour un signifié) et les polysèmes (plusieurs signifiés pour un signifiant) ;
- calculer les cardinalités ;
- vérifier la troisième forme normale et la normalisation des relations (surtout pour les associations non binaires) ;
- effectuer les corrections nécessaires.

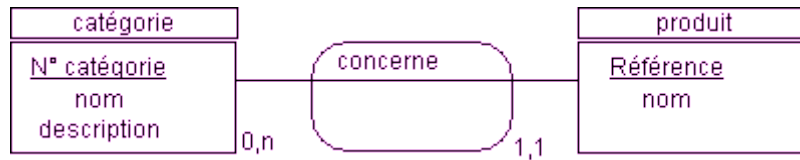
Il faut garder également à l'esprit que le modèle doit être exhaustif (c'est-à-dire contenir toutes les informations nécessaires) et éviter toute redondance (le prix unitaire d'un article n'a pas besoin de figurer dans l'entité commandes, c'est la troisième forme normale).

Quelques exercices:

Dans une clinique, les médecins spécialisés pratiquent des actes techniques sur des patients.



Relation entre catégorie de produit et produit.

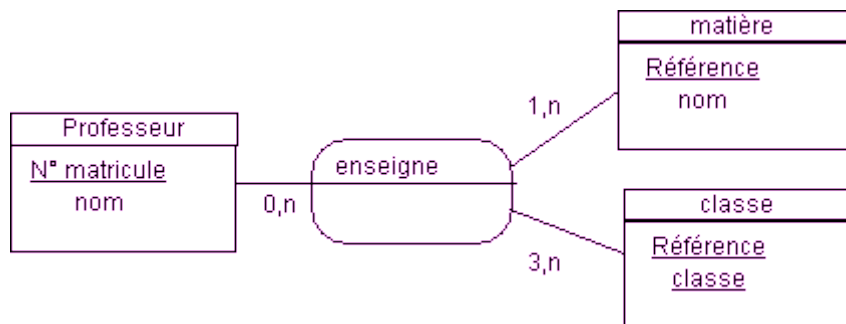


MCD d'une école:

Tout professeur enseigne une matière au moins mais peut est en "disponibilité par défaut d'emploi".

Toute matière reprise dans la liste est enseignée au moins une fois dans une classe.

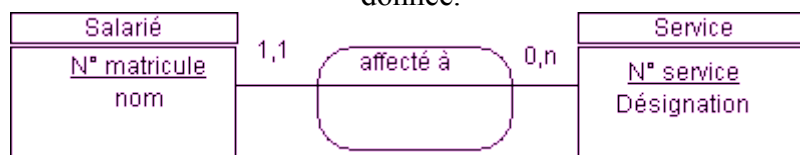
Toute classe a au moins 3 enseignants.



Un homme est le fils d'une femme.



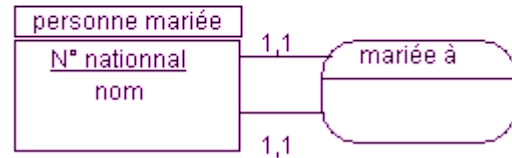
Un salarié (N° matricule, nom) est affecté à un service (N° service, intitulé) à partir d'une date donnée.



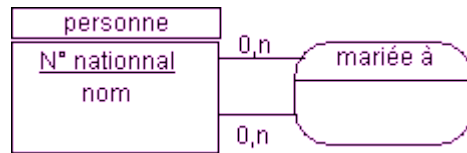
Un livre est écrit par un auteur (au moins)



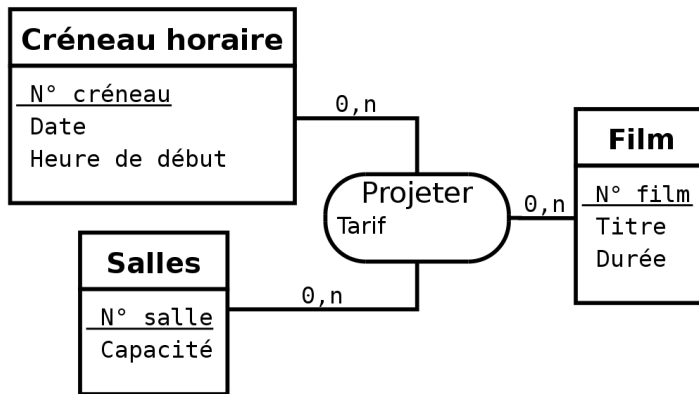
Relation entre deux personnes de l'ensemble des personnes mariées actuellement.



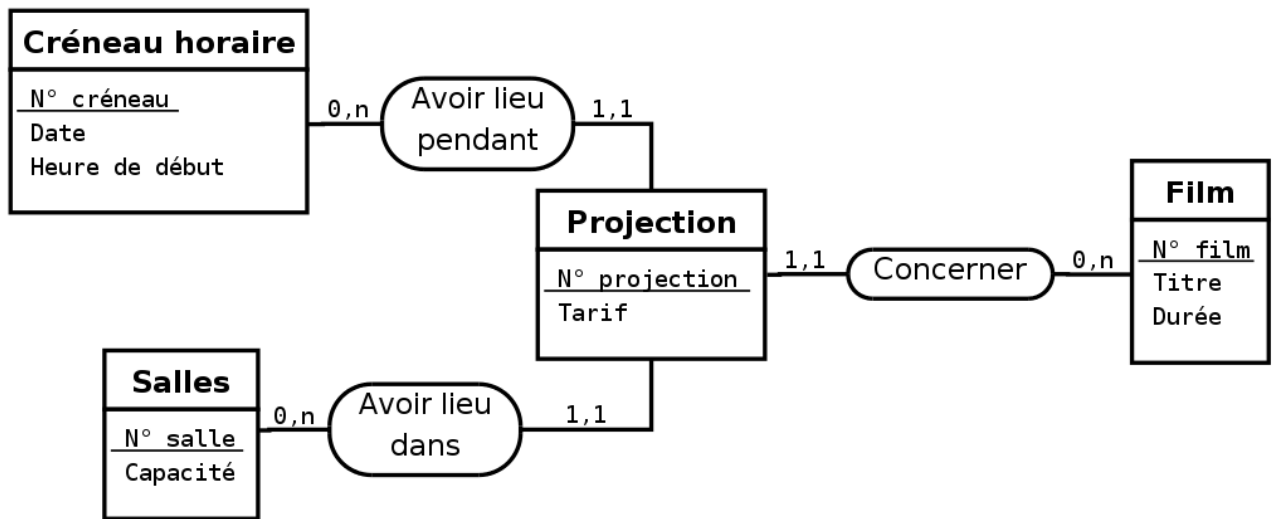
Ensemble des personnes et la relation de mariage



Décomposition d'une association n-aire



Exemple de type association ternaire entre des type-entités *Créneau horaire*, *Salle* et *Film*.

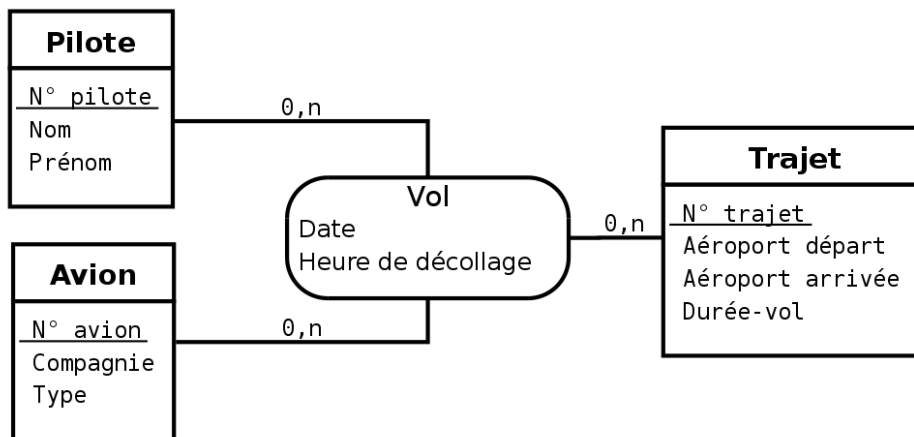


Transformation du type-association ternaire en un type-entité et trois type-associations binaires.

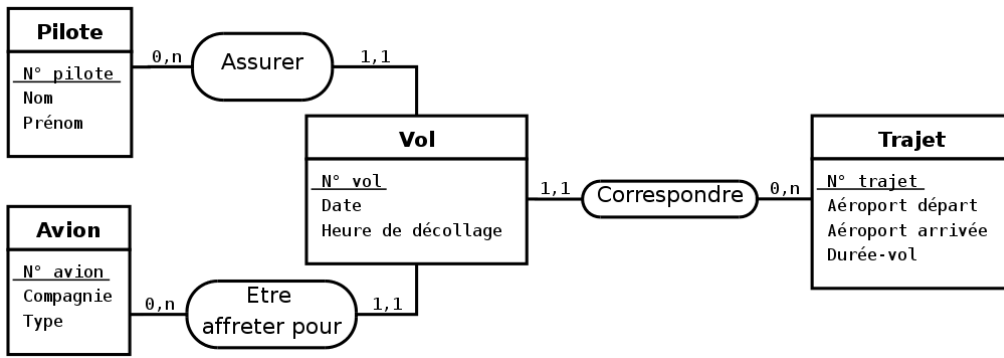
Il est toujours possible de s'affranchir d'un type-association n-aire ($n > 2$) en se ramenant à des type-associations binaires de la manière suivante :

- On remplace le type-association n-aire par un type-entité et on lui attribue un identifiant.
- On crée des type-associations binaire entre le nouveau type-entité et tous les type-entités de la collection de l'ancien type-association n-aire.
- La cardinalité de chacun des type-associations binaires créés est 1,1 du côté du type-entité créé (celui qui remplace le type-association n-aire), et 0,n ou 1,n du côté des type-entités de la collection de l'ancien type-association n-aire.

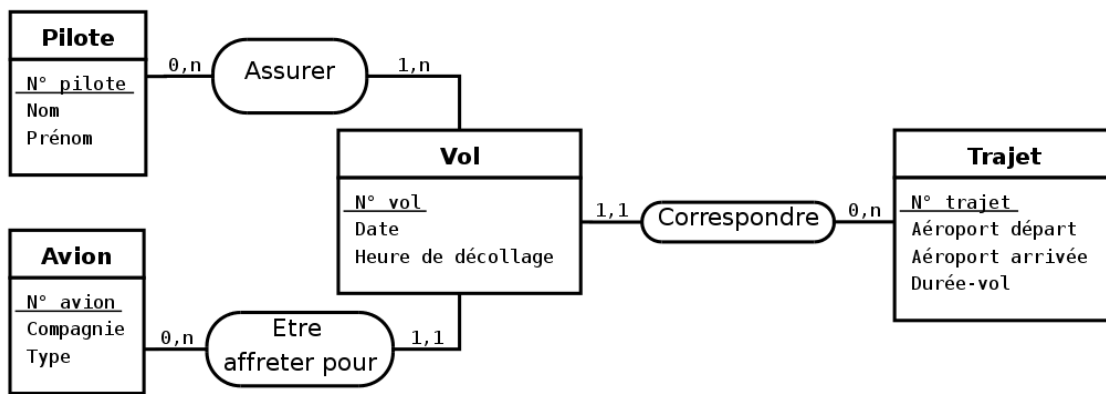
Détection d'une erreur de modélisation par décomposition d'une association n-aire



Modèle représentant un type-association ternaire *Vol* liant trois type-entités *Avion*, *Trajet* et *Pilote*.



Transformation du type-association ternaire en un type-entité et trois type-associations binaires.



:Modèle corrigé au niveau des cardinalités.

Contraintes d'intégrité fonctionnelle (CIF)

Définition:

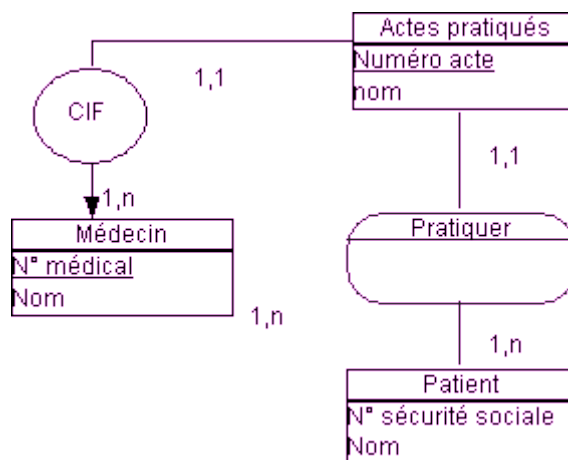
Une CIF exprime que l'un des objets est totalement déterminé par la connaissance d'un ou plusieurs autres.

L'intérêt d'une CIF dans une relation de dimension supérieure à deux réside dans le fait qu'on peut diminuer la dimension de la relation de 1.

Exemple:

Dans une clinique, les médecins spécialisés pratiquent des actes techniques sur des patients.

On voit dans cet exemple qu'un acte est nécessairement pratiqué par un médecin et qu'un médecin peut pratiquer plusieurs actes.



Globalement, on obtient deux relations d'ordre 2 au lieu d'une d'ordre 3.

Dépendance fonctionnelle (DF).

Si la dépendance n'est pas permanente dans le temps, il s'agira d'une dépendance fonctionnelle (DF)

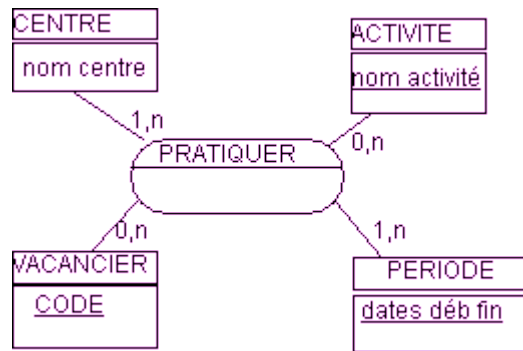
Exemple:

Un groupe sportif comprend différents centres d'activités. Les différentes activités organisées sont réparties dans les différents centres et la répartition varie selon les périodes de l'année.

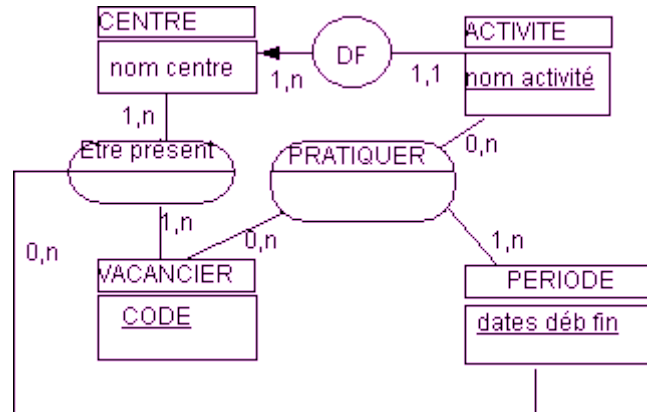
Les principaux traitements sont :

Enregistrement des activités pratiquées par les vacanciers.

Elaboration des statistiques des activités pratiquées par centre et par vacanciers.



Une même activité n'existe que dans un seul centre mais pas de façon permanente
On a une dépendance fonctionnelle.

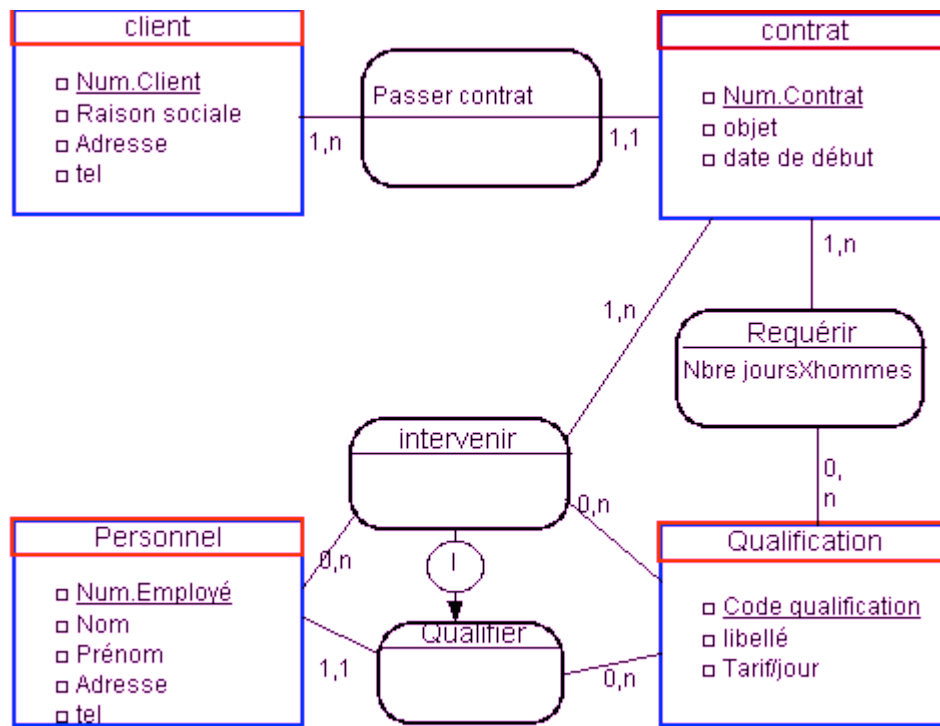


Contraintes d'extension sur les relations entre objets.

- Contraintes d'inclusion : (I)
- Contraintes d'exclusion : (X)
- Contraintes de totalité : (T)
- Les contraintes de ou exclusif (+)
- Les contraintes d'égalité (=)

Inclusion:

Reprenons l'exemple de la société de services mais supposons qu'un employé possède plusieurs qualifications (une seule est utilisée à la fois).



Chaque couple d'occurrence Personnel-Qualification de la relation INTERVENIR est inclus dans la relation QUALIFIER.

Modèle conceptuel des traitements

Le **traitement** constitue la partie dynamique du système d'information.
Il décrit les actions à organiser afin d'obtenir les résultats demandé.

I) règles de logique:

Objet de la logique.

Il est difficile de donner une définition complète de la logique sans la pratiquer mais tous les logiciens, d'Aristote à nos jours, s'accordent à dire que la logique étudie les raisonnements .
Un raisonnement est un discours dans lequel on passe d'un certain nombre de phrases appelées prémisses pour aboutir à une autre phrase appelée conclusion et introduite par une particule telle que "**donc**".

Dans un raisonnement, il importe de distinguer le FOND et la FORME. La logique étudie les FORMES qui définissent un raisonnement comme correcte mais ne peut rien dire sur le fond. Pour mener une étude efficace, on utilise des symboles particuliers. C'est la raison pour laquelle cette logique est appelée: LOGIQUE SYMBOLIQUE.

A) Validité d'un raisonnement.

Dans la vie courante, on entend souvent dire :
Cette affirmation n'est pas logique.

ex: *Les truites sont des haricots*. Cette affirmation est absurde du point de vue du FOND mais a une FORME parfaitement valide. Un logicien s'intéresse au caractère **valide** ou **invalide** d'un raisonnement.

Lorsqu'on dit "Ce type raisonne comme un pied" on affirme simplement que le raisonnement est invalide.

Un raisonnement valide est un raisonnement qui obéit aux règles de la logique. La validité d'un raisonnement est indépendante de la **valeur de vérité** (vérité ou fausseté) de ses arguments.

Le but est donc de définir ces règles.

B) Proposition:

DEFINITION

Une proposition est un énoncé VRAI ou FAUX. Une proposition sera représentée par une lettre :
p,q,...

La notion de vrai ou de faux représente la
VALEUR DE LA PROPOSITION.

Cette valeur peut être représentée par un système binaire:

$$\text{VRAI} = 1 \quad \text{FAUX} = 0$$

1) Une proposition est vraie ou fausse; c'est le **principe du tiers exclus**

2) Elle n'est jamais à la fois vraie et fausse; c'est le **principe de non contradiction**

Exemples:

"Ma mère est un homme" est une proposition.

"6 est plus petit que 7" est une proposition.

"Va au diable !" n'est **pas une proposition**.

C) Fonctions et foncteurs propositionnels.

Considérons la proposition " Jean est en sciences et Anne est en langues ". Il est clair que cette proposition complexe est constituée de deux propositions simples reliées par **et**.

Chaque proposition est nommée **argument**.

L'expression qui relie n arguments (ex: **et, ou**) est un **foncteur de vérité à n arguments**.

La proposition complexe constituée par l'ensemble des propositions simples et appelée **fonction de vérité à n arguments**.

D) Notion de condition:

Les propositions

un rectangle est un parallélogramme,

un carré est un parallélogramme,

un losange est un parallélogramme,

un trapèze est un parallélogramme,

ont toutes la forme "x est un parallélogramme." que nous noterons aussi $p(x)$

E) Opérations sur les propositions.

1) NEGATION

La négation d'une proposition est l'opération qui change la valeur de la proposition.

La négation de p est non p

· (\bar{p})

Si p est VRAI alors non p est FAUX et inversement.

table de vérité:

\bar{p}
p p
1 0

0 1

Exemples: p: le ciel est bleu non p: le ciel n'est pas bleu

\bar{p} : $x < 5$ p: $x \geq 5$

Soit la condition $p(x)$: "Le résultat x du jet d'un dé est inférieur à 3".

domaine $D = \{ 1,2,3,4,5,6 \}$

validité $V = \{ 1,2 \}$

\bar{p} s'exprime par "Le résultat x du jet d'un dé est plus grand ou égal à 3" dont l'ensemble de vérité sera $\{ 3,4,5,6 \}$ qui est le complémentaire de V dans D .

2) CONJONCTION ET \cap

Exemple:

p : Demain, il y aura congé.

q : Demain, nous serons mercredi.

Composons les deux propositions par la conjonction de coordination "et":

"Demain, il y aura congé et demain nous serons mercredi." Cette proposition est appelée conjonction

Notation $p \cap q$. (p et q)

La conjonction est l'opération qui compose deux propositions au moyen de la conjonction de coordination "et". Le résultat est une proposition qui n'est vraie que si les propositions composantes sont vraies.

Rem: En français, le ET peut avoir un autre sens qui est celui de l'addition. Ex.: Il est cinq heures et quart.

Table de vérité:

p q $p \cap q$

1 1 1

1 0 0

0 1 0

0 0 0

3) DISJONCTION \cup

Exemple:

p : Il étudie.

q : Il échoue.

Au moyen de la conjonction de coordination "ou" nous obtenons une nouvelle proposition composée, appelée disjonction de p et q .

"Il étudie ou il échoue."

Notation: $p \cup q$ se lit p ou q

La disjonction est l'opération qui compose deux propositions au moyen de la conjonction de coordination "ou" dans le sens inclusif.

Le résultat est une proposition qui est vraie si au moins une des propositions composantes est vraie.

Les deux propositions peuvent être vraies en même temps.

Ex.: Il est technicien ou peintre. Rien n'interdit d'être les deux.

Dans le premier exemple on peut hélas, étudier **et** échouer mais pas nécessairement.

Table de vérité

P	Q	$p \cup q$
1	1	1
1	0	1
0	1	1
0	0	0

4) OU EXCLUSIF

Je rentre ou je sors.

Table de vérité

P	Q	$p \text{ xor } q$
1	1	0
1	0	1
0	1	1
0	0	0

5) Implication

L'implication est l'opération qui combine deux propositions au moyen d'une conjonction de subordination "Si ... Alors...".

Le résultat est une proposition qui n'est **fausse** que si la première proposition est vraie et la deuxième fausse

P	Q	$p \Rightarrow q$
1	1	1
1	0	0
0	1	1
0	0	1

6) Equivalence

L'équivalence est une opération qui combine deux propositions au moyen de la conjonction "Si ... et seulement si".

Le résultat est une proposition qui n'est vraie que si les deux propositions ont la même valeur de vérité.

P	Q	$p \Leftrightarrow q$
1	1	1
1	0	0
0	1	0
0	0	1

F) Lois de DE MORGAN

$$\overline{\overline{p \cap q}} \Leftrightarrow p \cup q$$

$$\overline{p \cap q} \Leftrightarrow \overline{p} \cup \overline{q}$$

$$p \cap q \Leftrightarrow \overline{\overline{p \cup q}}$$

La conjonction de deux propositions est équivalente à l'inverse de la disjonction des propositions inverses.

II) Traitement

Le traitement constitue la partie dynamique du système d'information.
Il décrit les actions à organiser afin d'obtenir les résultats demandé.

Exemple:

Une commande ne sera exécutée que si la quantité en stock est supérieure à la quantité demandée.

On peut donc décomposer le traitement en

lire la quantité demandée Q

lire la quantité en stock S

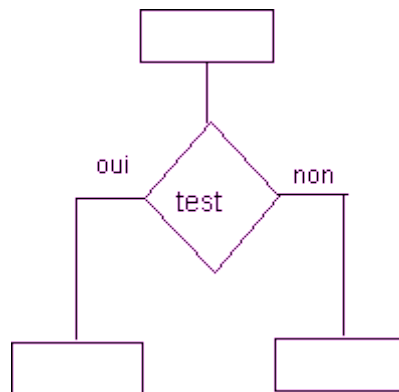
Si la quantité $Q < S$ alors

on exécute

sinon

on ne fait rien.

On voit une structure bien particulière dite structure alternative.



Dans ce graphique, une instruction est représentée par un rectangle et on se déplace du haut vers le bas.

Le losange représente un test (branchement conditionnel) qui a deux réponses possibles oui ou non. On effectue un test et selon le résultat, on part vers l'instruction de droite ou de gauche.

Structures:

Tout (langage) traitement évolué possède trois types de structures:

La séquence

est une suite d'instructions (d'opérations) telle que la fin de l'exécution d'une entraîne l'exécution de la suivante. Quitter une séquence pour aller vers une autre c'est effectuer **un branchement** et si le branchement est conditionné par un choix, on l'appelle **branchement conditionnel**.

L'alternative

donne le choix entre différentes séquences d'instructions. Ce choix est le résultat d'un test.

La répétitive

permet d'exécuter indéfiniment la même séquence d'instructions **tant qu'une condition** est (n'est pas) satisfaite ou **jusqu'à ce qu'une condition** soit (ne soit pas) satisfaite.

Un (programme) traitement contient généralement ces trois structures et chaque structure peut contenir les deux autres.

Certains programmes utilisent une autre façon de procéder qui est la rupture de séquence avec branchement obligatoire sur une autre séquence (ALLER A ou GOTO).

On démontre que cette façon de procéder peut être évitée dans tous les cas.

Les travaux de BOEHM et JACOPINI ont abouti au théorème suivant:

Tout algorithme peut-être conçu en utilisant les trois éléments cités plus haut, c'est à dire des répétitives, des alternatives et la séquence.

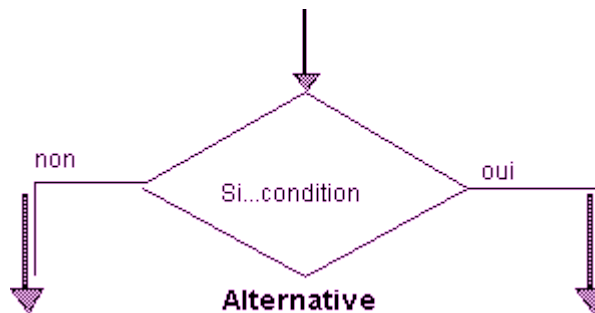
Les alternatives et les répétitives sont les deux piliers d'un programme structuré.

L'alternative a la forme:

SI... CONDITION ALORS SINON

IF ... THEN ELSE

ORGANIGRAMME:

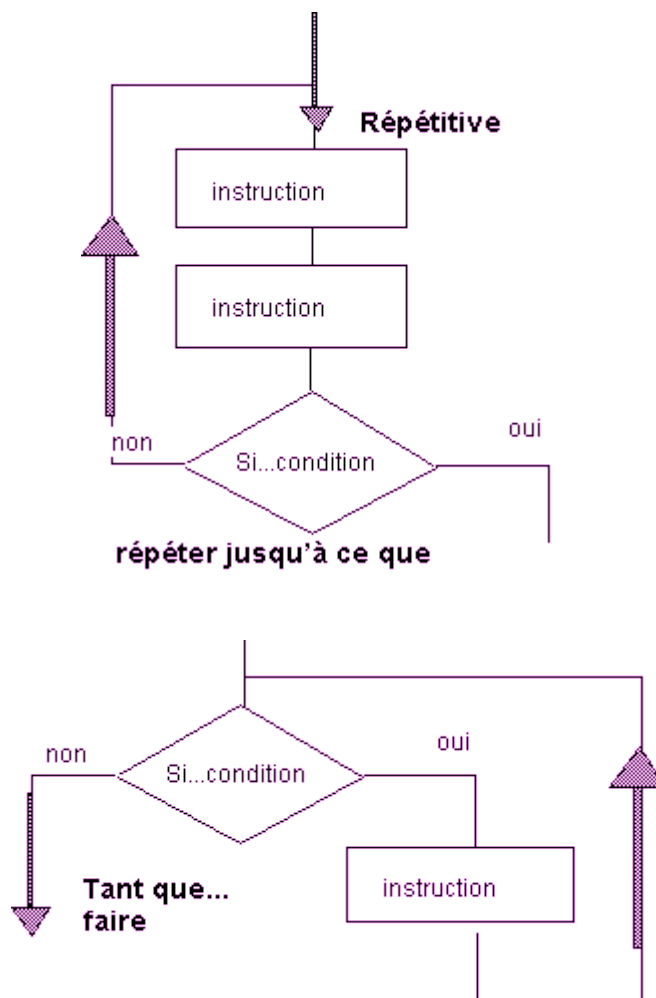


La répétitive peut s'écrire sous deux formes:

REPETER séquence . JUSQU'A CE QUE CONDITION VERIFIEE
REPEAT UNTIL

TANT QUE CONDITION VERIFIEE
séquence à répéter .
FIN REPETITION
WHILE END

La différence entre les deux est que le test est effectué avant ou après exécution de la séquence.



Passage du formalisme entités/rerelations au formalisme Enregistrement/ensemble

Cas du modèle relationnel.

On ne raisonne plus en termes d'enregistrements mais en termes de **relation entre différentes propriétés**

Exemple:

Une table clients qui contient les champs

code, Nom, rue, numero code_postal et localité constitue une relation entre code, nom, rue, numero, code_postal et localité

Il ne faut pas confondre les relations du modèle relationnel avec les relations du modèle Entités/rerelations à partir duquel nous avons construit nos M.C.D.

Dans le modèle Entités/rerelations, une relation exprime une association entre entités.

Dans le modèle relationnel, il s'agit d'associations de propriétés.

La notion de relation au sens relationnel est équivalente à celle d'enregistrement à propos des fichiers.

Relation-type = ensemble des propriétés - type.

exemple:

code, Nom, rue, numero code_postal et localité constitue une relation-type

Occurrence d'une relation=ensemble de valeurs des propriétés types appartenant à une relation-type.

Exemple:

A001, Duchmol, rue des deux églises, 69, 1000, Bruxelles

Le domaine

Chaque propriété prend ses valeurs sur un domaine. Le domaine est l'ensemble des valeurs d'une propriété.

Le degré

Le degré d'une relation est le nombre de constituants qui participent à cette relation; dans l'exemple : 6 (code, Nom, rue, numero code_postal et localité)

on dit parfois aussi 6-uple

La cardinalité

La cardinalité d'une relation est son nombre d'occurrences.

tableau -table

Une relation peut se représenter par un **tableau** (une table)

exemple :

relation type code élève, nom , résultat du cours, examen, total pondéré

code élève	nom	résulta du cours	examen	total pondéré
1	GŽnie	18	17	17,5
2	Benet	10	12	11
3	Lagaffe	15	5	10
4	Fayot	18	2	10

5-uple avec 4 occurrences. Le domaine des propriétés nom est Génie Benet Lagaffe Fayot

La clé

La clé d'une relation est la propriété différente pour chaque occurrence.

Normalisation

Chaque propriété doit dépendre de l'ensemble de propriétés qui participent à la relation mais d'aucun sous-ensemble.

Exemple, dans la base de données Musique, (voir exercices) on voit que la caractéristique d'une catégorie dépend du numéro de la catégorie uniquement. On aurait intérêt à découper cette relation en deux et à établir une jointure (voir plus loin).

Passage du modèle conceptuel des données au modèle relationnel.

Règle 1

Toute **entité devient une relation** ayant pour **clé primaire son identifiant**. Chaque propriété se transforme en attribut.

Règle 2

Toute association hiérarchique (de type **[1, n]**) se traduit par une **clé étrangère**. La clé primaire correspondant à l'entité père (côté n) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1).

Règle 3

Toute **association non hiérarchique (de type [n, n] ou de dimension > 2)** devient une **relation**. La clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

Exception à la règle 1

Les entités n'ayant que leur identifiant comme attribut ne deviennent pas des relations, mais des **attributs dans les autres relations liées**.

SGDB relationnel.

Un **système de gestion de base de données** est un logiciel permettant de décrire, manipuler et traiter les données d'une base.

Il utilise le concept de relation et assure une totale indépendance des programmes et des données. le modèle physique n'est pas connu du programmeur. Celui-ci ne voit que les relations.

Il possède un puissant langage de manipulation des données utilisant l'algèbre relationnel ou le calcul des prédicats.

La définition des données se fait en décrivant le modèle relationnel; cette définition assure la création automatique du modèle physique.

table	clé		
client	code_cli		
commande	N°Bon	code_cli	date
ligne de commande	N°Bon	N _i Ref	quantité
produits	N°Ref		prix unitaire
TVA	N°Catégorie	taux	

Passage du modèle conceptuel des données au modèle relationnel.

Règle 1

Toute entité devient une relation ayant pour clé primaire son identifiant. Chaque propriété se transforme en attribut.

Règle 2

Toute association hiérarchique (de type [1, n]) se traduit par une clé étrangère. La clé primaire correspondant à l'entité père (côté n) migre comme clé étrangère dans la relation correspondant à l'entité fils (côté 1).

Règle 3

Toute association non hiérarchique (de type [n, n] ou de dimension > 2) devient une relation. La clé primaire est formée par la concaténation (juxtaposition) l'ensemble des identifiants des entités reliées. Toutes les propriétés éventuelles deviennent des attributs qui ne peuvent pas faire partie de la clé.

Exception à la règle 1

Les entités n'ayant que leur identifiant comme attribut ne deviennent pas des relations, mais des attributs dans les autres relations liées.

Une relation R du MCD de type



disparaît dans le MLD, l'identifiant de A étant incorporé à la relation B

Une relation R du MCD de type



devient une relation au sens relationnel, sa clé étant obtenue en concaténant les identifiants qui participent à cette relation.

Si R est porteuse de propriétés, celle-ci deviennent les constituants de la table R .

Exemple:

Un client passe commande de produits affectés de taux de TVA différents.

MCD

