

Cours Bases de données
3^{ème} Année Systèmes d'Information

Chapitre 01

Conception des Bases de Données Le Modèle Entité/Association

Fouad DAHAK

Enseignant-Chercheur

Chargé de cours Bases de données

Ecole Nationale Supérieure d'Informatique (ESI)

(f_dahak@esi.dz – <http://dahak.esi.dz>)

Table des matières

I. Concepts de base	4
I.1. Note	4
I.2. Présentation	4
I.3. Éléments du modèle	5
I.3.1. Entité	5
I.3.2. Type - Entité	5
I.3.3. Attribut, valeur	6
I.3.4. Identifiant ou clé	6
I.3.5. Association ou relation	7
I.3.6. Type-Association	7
I.3.7. Représentation Graphique	8
I.4. Les Cardinalités	8
I.4.1. Concept de cardinalité	8
I.4.2. Interprétation des relations	9
I.5. Concepts Supplémentaires	12
I.5.1. Entité faible	12
I.5.2. Structures Hiérarchiques	13
I.5.3. Associations plurielles	13
I.5.4. Associations réflexives	13
I.5.5. Les domaines de valeurs	14
I.5.6. Conserver l'Histoire	14
I.5.7. Agrégation	15
I.5.8. Généralisation / spécialisation	16
II. Les Contraintes d'Intégrité	17
II.1. Objectif	17
II.2. Définition :	17
II.3. Effet	17
II.4. Formulation des contraintes	18
II.4.1. Un formalisme inspiré de la logique du premier ordre	18
II.4.2. Représentation graphique	18
II.5. Types de Contraintes d'intégrité	18
II.5.1. Contraintes d'intégrité sur les attributs	18
II.5.2. Contraintes d'intégrité sur les cardinalités	19
II.5.3. Contraintes sur les entités / association	19
III. Le Schéma Conceptuel	22

III.2. Règles de conception	22
III.2.1. Règle de représentation par un type d'entité	22
III.2.2. Règle de représentation par un type association	22
III.2.3. Règle de représentation par un attribut	22
III.3. Règles d'optimisation	23
III.3.1. Règles portant sur les noms	23
III.3.2. Règles de normalisation des attributs.....	23
III.3.3. Règles de fusion/suppression d'entités/associations	23
III.4. Vérification du schéma EA	23
III.4.1. Vérification "syntaxique"	23
III.4.2. Par jeu d'essai.....	23
III.4.3. Complétude par rapport aux traitements	24
III.4.4. Par les utilisateurs.....	24
III.5. Description d'un schéma EA	24
III.5.1. Entête	24
III.5.2. Description d'un TE	24
III.5.3. Description d'un TA	24
III.5.4. Description d'un Attribut	24
IV. Conclusion	25
IV.1. Limites du modèle E/A	25
IV.2. Inconvénients du modèle E/A	25
IV.3. Avantages du modèle E/A	25

I. Concepts de base

I.1. Note

Le niveau d'analyse conceptuel est un niveau d'analyse sémantique. C'est à dire qu'il n'est aucunement préoccupé par l'aspect physique de l'implantation des structures d'information, mais se préoccupe plutôt du sens des choses.

Quand on est au niveau conceptuel, on ne doit pas se préoccuper de l'aspect ni des fonctionnalités du système projeté en termes de traitements. Ce qui nous intéresse à ce niveau c'est la structuration des données de manière à ce que le modèle conçu soit le plus proche de la réalité perçue.

Il y a également un aspect très important à prendre en considération et qui représente un piège que beaucoup d'étudiants n'arrivent pas à éviter au moment de la conception des données. Il s'agit du facteur temps. En fait, la conception des données n'est pas liée au temps. On conçoit une base de données dont la structure reste valable, correcte et proche de la réalité et ceci indépendamment du temps.

I.2. Présentation

Une base de données est un ensemble de données structurées qui représentent une partie de la réalité perçue. Avant que cette base de données ne prenne sa forme finale (une forme utilisable par un SGBD) il faut passer une étape très importante qu'est la conception de cette dernière. Il s'agit donc de représenter les objets de la réalité et les interactions entre ces objets de manière à ce qu'ils soient facilement manipulables, pour cela on a besoin de modéliser et donc l'utilisation d'un modèle est nécessaire. Le modèle entités-associations constitue l'un des meilleurs modèles de conception des données et des plus courants.

Ce modèle, présenté par le Prof. Peter Pin-Shan Chen (1976), permet une description naturelle du monde réel à partir des concepts d'entité et d'association.

Basé sur la théorie des ensembles et des relations, ce modèle se veut universel et répond à l'objectif d'indépendance données-programmes.

Ce modèle, utilisé pour la phase de conception, s'inscrit notamment dans le cadre d'une méthode plus générale et très répandue : Merise.

L'idée fondamentale du modèle EA est de retenir comme concepts de base pour la représentation les mêmes concepts génériques que ceux qui guident le processus d'abstraction conduisant de l'observation d'une réalité à sa description.

On suppose que la perception d'une situation observée se fait naturellement sur la base d'une identification des objets présents (qu'ils soient réels, une personne, ou abstraits, une foule), de liens entre ces objets (une personne conduit une voiture) et de propriétés observables (taille, couleur ...).

Le modèle Entité / Association propose une description sur la base de ces mêmes trois concepts, la correspondance entre les trois concepts génériques et la terminologie du modèle EA est la suivante:

- Objet → entité
- Lien → association (relation)
- Propriété → attribut

I.3. Éléments du modèle

Un diagramme entités/association se compose de trois types d'objets:

- Les entités,
- Les relations et
- Les attributs (qu'on nomme aussi propriétés).

I.3.1. Entité

Définition 1: Une entité est un objet, une chose concrète ou abstraite qui peut être reconnue distinctement et qui est caractérisée par son unicité.

Dès qu'on rencontre des objets identifiable de manière unique (sans confusion ni ambiguïté) qui interagissent et qui font ou subissent des actions dans notre système on parle d'entités. Par exemple, si je dis qu'Ali lit le livre « Bases de données » qu'il a acheté chez Amar. Je distinguerai nettement trois entités : Ali, le livre « Bases de données » et Amar. Ali fait deux actions, celles de lire et d'acheter, le livre « Bases de données » subit ces mêmes actions alors que Amar participe à l'action d'achat de Ali qui est interprétée de son coté comme étant une vente.

I.3.2. Type - Entité

Définition 2: Un type-entité désigne un ensemble d'entités qui possèdent une sémantique et des propriétés communes.

On parle de type entité quand on essaie de représenter de manière générale un ensemble d'entités qui ont une même description. C'est analogue au concept d'ensemble dans les mathématiques en considérant les entités comme les éléments de cet ensemble.

Dans l'exemple précédent on a présenté une situation bien définie celle de Ali qui lit le livre « Bases de données » qu'il a acheté chez Amar. Mais si on désire modéliser cette situation de manière générale il faut faire en sorte que notre modèle puisse représenter cette situation quel que soit le lecteur, l'acheteur, le livre ou le vendeur. Pour cela on ne devrait plus parler d'Ali, le livre « Bases de données » et Amar précisément mais de ce qu'ils représentent dans cette situation c'est-à-dire l'acheteur, le livre et le vendeur. Donc on pourrait dire finalement qu'Ali et Amar sont des entités du type entité Personne car tous deux ont les mêmes caractéristiques et le livre « Bases de données » est une entité du type-entité Livre.

Au niveau du modèle, ce sont les types-entités qui sont représentés et non pas les entités.

Note : Une entité est souvent nommée occurrence ou instance de son type-entité.

Remarque: Par abus de langage, on utilise souvent le mot entité au lieu du mot type-entité, il faut cependant prendre garde à ne pas confondre les deux concepts.

Personne

I.3.3. Attribut, valeur

Définition 3: Un attribut (propriété) est une caractéristique associée à un type-entité ou à un type-association.

Dans l'exemple précédent, Ali et Amar sont des prénoms des deux entités qu'ils désignent et « Bases de données » est le titre du livre acheté par Ali. En fait « Ali » est une valeur pour la propriété prénom, c'est donc le prénom qui est un attribut et non pas Ali, idem pour le titre du livre.

On peut dire finalement, qu'une entité est une réalisation ou concrétisation des propriétés d'un type-entité ce qui signifie qu'on donne une valeur à chaque attribut du type-entité pour obtenir une entité de ce type-entité.

Définition 4: Chaque attribut possède un domaine qui définit l'ensemble des valeurs possibles qui peuvent être choisies pour lui (entier, chaîne de caractères, booléen, . . .).

Chaque attribut possède une valeur compatible avec son domaine.

Règle 1 : Un attribut ne peut en aucun cas être partagé par plusieurs type-entités ou type-associations.

Règle 2 : Un attribut est une donnée élémentaire, ce qui exclut des données calculées ou dérivées.

Règle 3: Un type-entité et ses attributs doivent être cohérents entre eux (i.e. ne traiter que d'un seul sujet).

I.3.4. Identifiant ou clé

Définition 5: L'identifiant l'ensemble minimum d'attributs qui désignent de manière unique une entité.

Le numéro de sécurité sociale par exemple peut être utilisé comme identifiant d'une personne comme le numéro de châssis pour une voiture et le code ISBN pour un livre.

Règle 4: Chaque type-entité possède au moins un identifiant, éventuellement formé de plusieurs attributs.

Ainsi, chaque type-entité possède au moins un attribut que, s'il est seul, est donc forcément l'identifiant.

Personne

- Nss
- Nom
- Prénom

I.3.5. Association ou relation

Définition 6: Une association (relation) est un lien sémantique entre plusieurs entités.

Dans l'exemple précédent on distingue un lien de lecture et d'achat entre Ali et le livre de bases de données et un lien de vente entre ce dernier et Amar.

I.3.6. Type-Association

Définition 7: Un type-association (type-relation) désigne un ensemble de relations qui possèdent les mêmes caractéristiques.

Le type-association décrit un lien entre plusieurs type-entités. Les associations de ce type-association lient des entités de ces type-entités.

Règle 5: Un attribut peut être placé dans un type-association uniquement lorsqu'il dépend de toutes les entités liées par le type-association.

Un type-association peut ne pas posséder d'attributs explicites et cela est relativement fréquent, mais il possède au moins des attributs implicites.

Une association est souvent nommée occurrence ou instance de son type-association.

Remarque:

Par abus de langage, on utilise souvent le mot association en lieu et place du mot type-association, il faut cependant prendre garde à ne pas confondre les deux concepts.

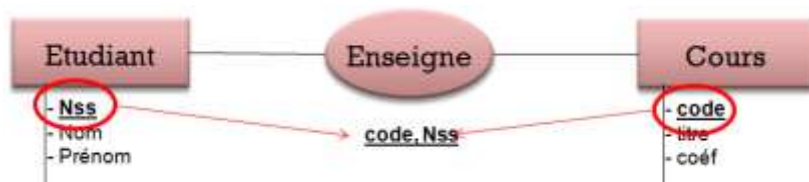
Définition 8: Les type-entités intervenant dans un type-association sont appelés les participants de ce type-association.

Définition 9: L'ensemble des participants d'un type-association est appelé la collection de ce type-association.

Cette collection comporte au moins un type-entité, mais elle peut en contenir plus, on parle alors de type-association n-aire.

Définition 10: La dimension, ou l'arité d'un type-association est le nombre de type-entités contenu dans la collection.

Règle 6: La concaténation des identifiants des type-entités liés à un type-association constitue un identifiant de ce type-association et cet identifiant n'est pas mentionné sur le modèle (il est implicite).



I.3.7. Représentation Graphique

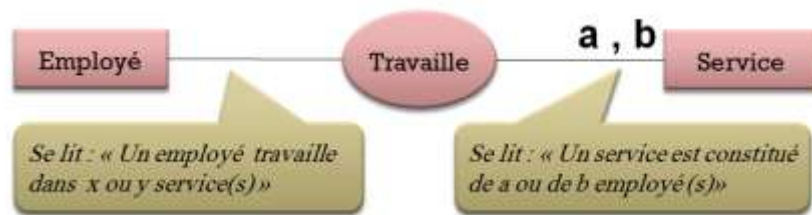
Dans le modèle entité association, le type-entité et le type-association ont plusieurs représentations graphiques assez similaires dans l'ensemble. Dans la suite de notre cours nous utiliserons la convention suivante :

Un type-entité est représenté avec un rectangle et le type-association avec une ellipse. Les pattes du type-association représentent le rôle que joue l'entité correspondante dans la relation.

I.4. Les Cardinalités

I.4.1. Concept de cardinalité

Définition 12: La cardinalité d'une patte reliant un type-association et un type-entité précise le nombre de fois minimal et maximal d'interventions d'une entité du type-entité dans une association du type-association.



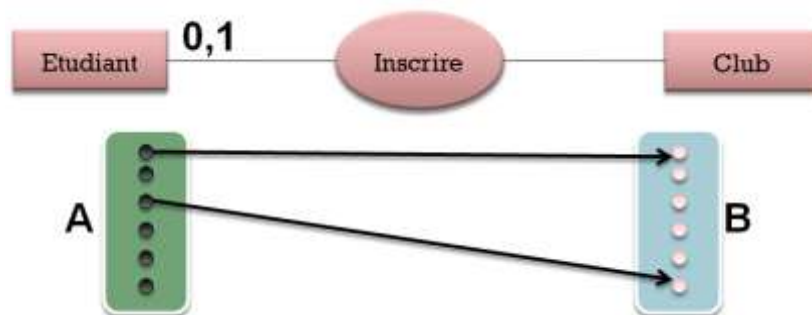
Règle 7: La cardinalité minimale doit être inférieure ou égale à la cardinalité maximale.

Règle 8: L'expression de la cardinalité est obligatoire pour chaque patte d'un type-association.

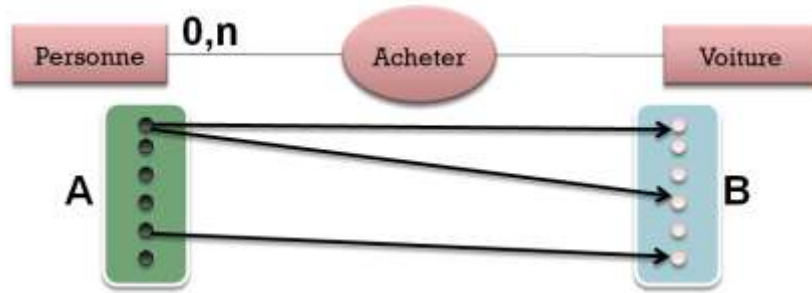
Règle 9: Une cardinalité minimal est toujours 0 ou 1 et une cardinalité maximale est toujours 1 ou n.

Les seuls cardinalités admises sont donc : 0,1 - 0,n - 1,1 - 1,n

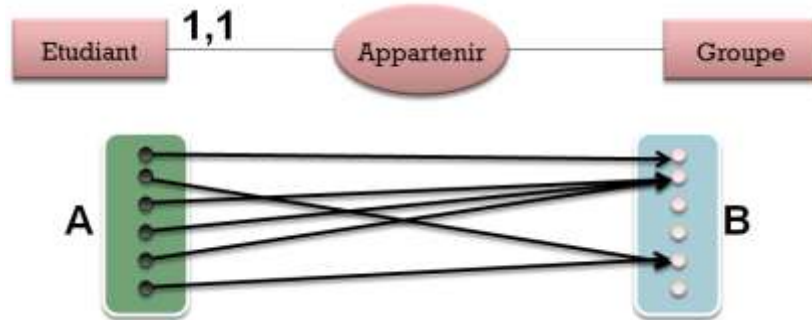
0,1 : Une occurrence du type-entité peut exister tout en n'étant impliquée dans aucune association et peut être impliquée dans au maximum une association.



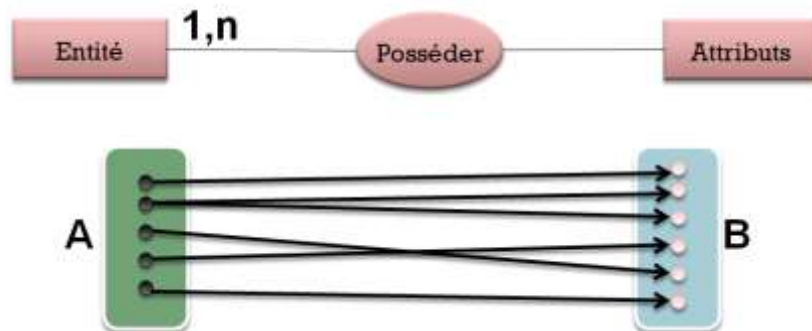
0,n : c'est la cardinalité la plus ouverte ; une occurrence du type-entité peut exister tout en étant impliquée dans aucune association et peut être impliquée, sans limitation, dans plusieurs associations.



1,1 : Une occurrence du type-entité ne peut exister que si elle est impliquée dans exactement (au moins et au plus) une association.



1,n : une occurrence du type-entité ne peut exister que si elle est impliquée dans au moins une association.



I.4.2. Interprétation des relations

I.4.2.1. Interprétation d'une association binaire



On interprète cette association comme suit :

Un employé exerce une fonction ou bien une fonction est exercée par un employé. Les pattes de l'association représentent le rôle que joue le type-entité correspondant à l'association.

Pour trouver les bonne cardinalité il faut répondre aux questions suivantes :

a : Un employé quelconque, peut-il n'être associé à aucune fonction par l'association exerce ?

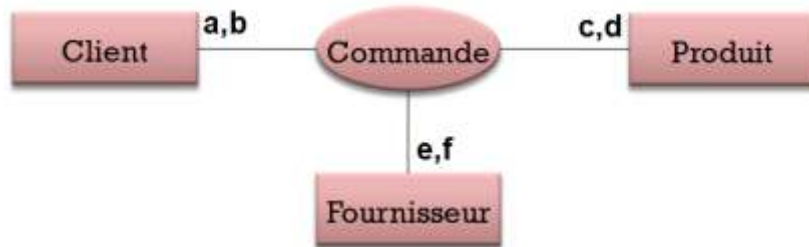
SI oui Alors $a=0$ SINON $a=1$

b : Un employé quelconque, peut-il être associé à plus d'une fonction, par l'association exerce?

SI oui Alors $b=1$ SINON $b=n$

1.4.2.2. Interprétation d'une association ternaire

On reprend le même exemple que précédemment mais cette fois-ci en y ajoutant un historique qui est représenté par le type-entité Date. Je m'intéresse donc aux fonctions qu'a exercées un employé dans le temps.



L'interprétation des associations ternaires n'est plus la même que pour les binaires. A ce niveau on interprète le rôle d'une entité par rapport au couple correspondant formé des deux autres entités participantes à l'association. On dit donc qu'un employé exerce des fonctions à des dates données. En quelque sorte on est en train de dire qu'un employé est lié à un couple (date, fonction) avec la relation exerce.

Pour trouver les différentes cardinalités, on répond aux questions suivantes :

a : Un employé quelconque, peut-il n'être associé à aucun couple (date, fonction) par l'association exerce ?

SI oui Alors $a=0$ SINON $a=1$

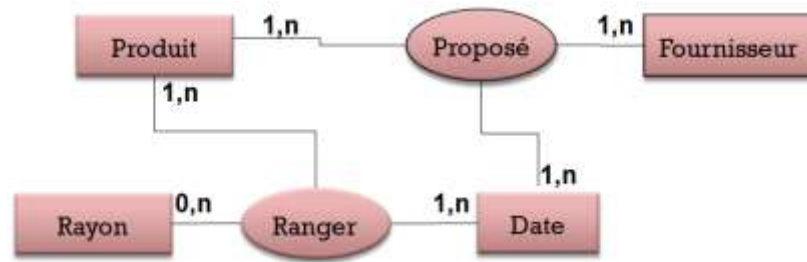
b : Un employé quelconque, peut-il être associé à plus d'un couple (date, fonction) par l'association exerce?

SI oui Alors $b=1$ SINON $b=n$

1.4.2.3. Remarques sur les cardinalités

- Une cardinalité minimale de 1 doit se justifier par le fait que les entités du type-entité en questions ont besoin de l'association pour exister. Dans tous les autres cas, la cardinalité minimale vaut 0.
- Une cardinalité minimale de 0 signifie qu'une entité du type-entité correspondant peut exister tout en étant impliquée dans aucune association.
- Si une entité **e** pour exister doit participer à l'association **a** on dit que la participation du type-entité **TE** au type-association **TA** est totale. Sinon, on parle de participation partielle.

1.4.2.4. Problème de la cardinalité minimale 1



Le fait de mettre 1 comme cardinalité minimale impose la participation de l'entité concernée à l'association pour exister. Dans l'exemple ci-dessus, la date ne peut exister que si elle participe à la fois à l'association « Proposer » et « Ranger ». Or la date de proposition d'un produit par un fournisseur n'existe pas forcément comme date de rangement et vis-versa. Ce qui fait que dans un cas pareil, la cardinalité minimale doit impérativement être 0 vu que les deux associations « Proposer » et « Ranger » sont sémantiquement indépendantes.

Règle 10: Pour trouver les cardinalités d'une patte d'une association il faut :

1. Prendre en considération toutes les pattes,
2. Pour chaque participant, prendre en considération toutes ses participations.

Cette règle est surtout valable dans le cas de présence de la cardinalité minimale 1.

1.4.2.5. Problème de l'association (1,1) - (1,n)



Dans l'association « Achète » ci-dessus, la propriété Date Achat est modélisée à son niveau, ce qui signifie qu'elle dépend à la fois de Voiture et de Personne. Or, le fait que la cardinalité du côté de Voiture est 1,1 signifie qu'une voiture ne peut exister que si elle participe Une et Une fois à l'association Achat, ce qui fait que finalement, la date achat dépend uniquement de Voiture et non pas des deux entités ensemble. C'est pourquoi elle est modélisée au niveau de Voiture plutôt que dans l'association.

Alors que dans le cas d'une association avec une cardinalité 0,1, la date achat dépend des deux entités. Car si on la met au niveau de voiture et sachant qu'une voiture peut exister sans participer à l'association, cette date pourrait ne pas avoir de valeur, ce qui n'était pas le cas dans la cardinalité 1,1. Donc, dans le cas de la cardinalité 0,1, la date achat doit être placée au niveau de l'association.

Règle 11: l'association 1,1 - 1,N ne doit pas posséder de propriétés.

I.5. Concepts Supplémentaires

I.5.1. Entité faible

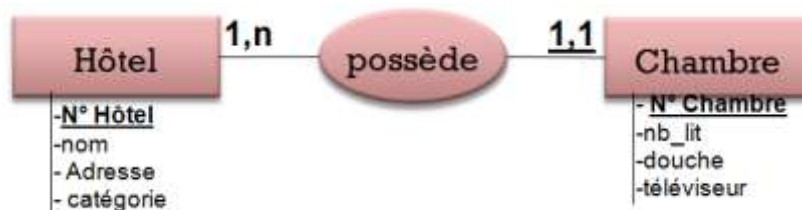
Jusqu'à présent nous avons considéré le cas d'entités indépendantes les unes des autres. Chaque entité, disposant de son propre identifiant, pouvait être considérée isolément.

Il existe des cas où une entité ne peut exister qu'en étroite association avec une autre entité, et est identifiée relativement à cette autre entité. On parle alors d'entité faible.

Définition 13: Une entité faible est une entité possédant un identifiant insuffisant de par lui-même pour identifier de manière unique chacune de ses occurrences. Sa caractéristique d'identifiant n'est valable qu'à l'intérieur du contexte spécifique de l'occurrence d'une entité principale.

Exemple : Un hôtel et ses chambres ou un cinéma et ses salles.

L'identifiant d'une chambre est constituée de deux parties : N° Hôtel et N° Chambre.



Remarque : L'introduction d'entités faibles n'est pas une nécessité absolue puisqu'on peut très bien utiliser une association classique. La principale différence est que, dans le cas d'une entité faible, on obtient une identification composée qui est souvent plus pratique à gérer, et peut également rendre plus faciles certaines requêtes. Alors que dans le second cas, on doit créer nous même un identifiant fictif qu'il faudra gérer par la suite.

Remarque : La cardinalité du côté de l'entité faible est toujours 1,1. La cardinalité maximale 1 se justifie par le fait d'identification composée (Chaque entité du type-entité faible n'a qu'une seule entité de l'autre type-entité par rapport à laquelle elle s'identifie) et la cardinalité minimale ne peut pas être 0 car cela signifierait qu'une entité du type-entité faible pourrait exister sans participer à l'association ce qui est contradictoire avec la définition de l'entité faible.

Remarque : Il est possible d'avoir des entités faibles en cascade, l'identifiant des entités est obtenu dans ce cas en effectuant la migration des identifiants des pères vers les fils, en cascade également.

Exemple :

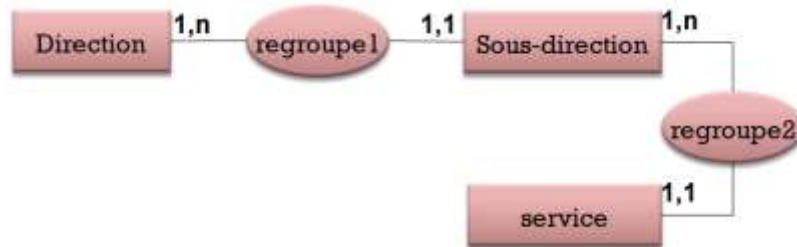
Dans ce cas, les entités du type-entité jours sont : Samedi, Dimanche, lundi, etc. celles de mois sont : Janvier, Février, Mars, etc. et pour année ce sont les années.

Le jour n'est identifié que dans le cadre d'une semaine précise, la semaine dans le cadre du mois et le mois dans le cadre d'une année

donnée. Un jour est donc identifié comme suit : Samedi de la première semaine du mois de janvier de l'année 2008.

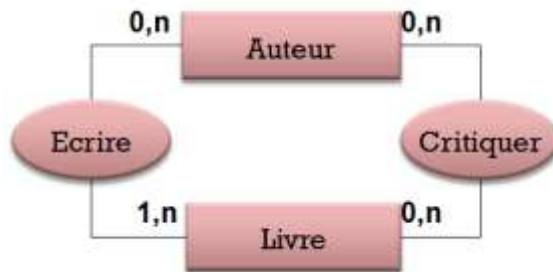
I.5.2. Structures Hiérarchiques

Définition 14: Une structure hiérarchique représente une décomposition de concepts allant du général au particulier. Il s'agit d'une structure où un parent peut avoir plusieurs enfants, mais où chaque enfant ne peut avoir qu'un seul parent.



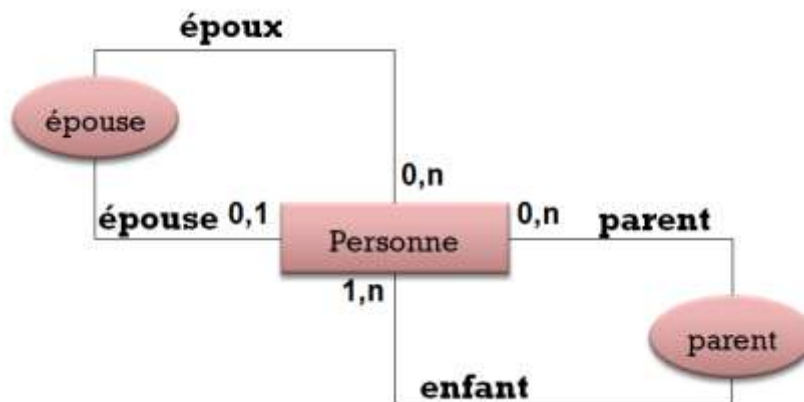
I.5.3. Associations plurielles

Les associations plurielles expriment le fait que deux objets puissent avoir plusieurs liens sémantiquement distincts.



I.5.4. Associations réflexives

Définition 15: Un type-association est qualifié de réflexif quand il matérialise une relation entre un type-entité et lui-même.



Note : Dans le cas d'une association réflexive, les rôles doivent être mentionnés clairement sur le diagramme.

Remarque : Deux entités ne peuvent participer à la même association avec le même rôle. Dans le cas d'un type-association réflexif, comme dans l'exemple ci-dessus, deux personnes ne peuvent participer à une association marier avec le même rôle.

I.5.5. Les domaines de valeurs

Certaines informations ne peuvent accepter qu'un ensemble déterminé et limité de valeurs. Elles sont donc restreintes à un domaine de valeurs (les valeurs permises pour le sexe).

Il y a deux façons de modéliser ces domaines de valeurs : par un attribut ou par une entité.

Par un attribut: Ajouter un attribut au sein de l'entité concernée et indiquer, dans la documentation du modèle, les valeurs permises.

Cette méthode est bien adaptée aux cas où l'utilisateur n'a aucun contrôle sur le domaine de valeurs en question (on ne peut ni ajouter ni retirer des valeurs permises).

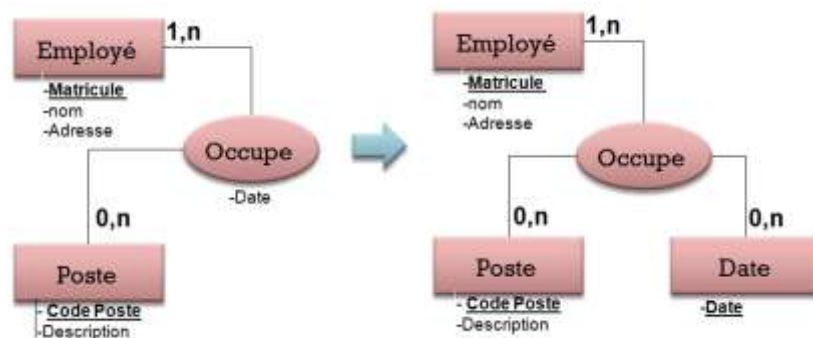
Par une entité : Ajouter une entité au modèle pour représenter le domaine de valeurs.

Cette méthode doit être utilisée lorsque l'utilisateur a plein contrôle sur le domaine de valeurs.

I.5.6. Conserver l'Histoire

Pour garder l'historique d'une relation on fait sortir l'attribut Date comme entité et l'ajouter à la collection de cette association.

Dans le premier cas la date n'est qu'une propriété de l'association « Occupe ». Donc si je prends le cas d'un employé E1 qui a occupé les deux postes P1 et P2 j'aurai les occurrences suivantes de l'association « Occupe » : (E1,P1,20/02/2000) et (E1,P2,05/04/2002) sachant que l'identifiant de l'association « occupe » est celui de employé concaténé avec celui de poste. Mais imaginons maintenant que l'employé E1 retourne à son poste d'origine qui est P1 donc j'aurai une autre occurrence de ce type (E1,P1,06/10/2003). Ici, il y a problème car une occurrence avec cet identifiant existe déjà ce qui fait que l'ancienne occurrence est remplacée avec la nouvelle. Donc je perds l'information que cet employé a déjà occupé ce poste à une date ultérieure. Le seul moyen de garder les deux occurrences c'est de faire en sorte que la date fasse partie de l'identifiant de l'association « occupe ». Pour cela il faut que ça soit une entité qui participe à cette association.



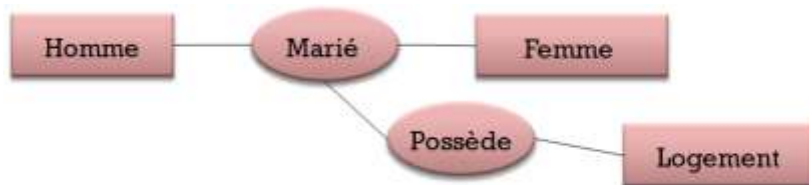
I.5.7. Agrégation

Définition 16: Agréger c'est réunir de manière à constituer un tout.

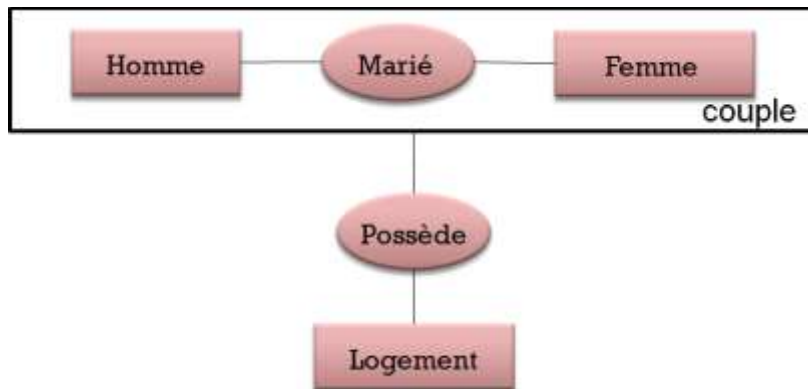
L'agrégation est un concept qui est rajouté au modèle entité association pour permettre d'exprimer certaines situations où nous avons un lien entre un objet et un ensemble d'objets liés avec une association.

Supposons qu'on veuille modéliser le fait qu'un couple de personnes possède un appartement. Ce ne sont pas les deux personnes individuellement qui possèdent l'appartement mais les deux ensemble. Un couple est représenté par une association réflexive « Marier ». Ce sont donc les deux personnes qui sont liées par cette association qui peuvent posséder un appartement.

Pour modéliser cette situation il faudrait relier deux associations entre elles, ce qui est interdit dans le modèle entités – associations de base mais que le modèle étendu permet. Il existe cependant deux manières de représenter l'agrégation : la première consiste à relier deux associations entre elles,



Et la seconde, celle que nous utiliserons dans la suite de notre cours, consiste à délimiter le bloc, le nommer ensuite le lier avec une autre association comme si c'était une nouvelle entité.



Remarque : Une agrégation de plusieurs entités n'a de sens que si ces entités sont liées entre elles avec une association et une seule.

Quand on veut faire plusieurs agrégations on utilise les regroupements en cascade.

Remarque : Quand on a une relation avec une cardinalité 1,1 d'un côté, l'agrégation de cette association peut être facilement évitée. Car l'existence du type-entité fils (du côté 1,1) est conditionné par l'existence du type-entité père et de l'association peut être remplacée par ce type-entité dans l'agrégation.

Exemple :

Dans cet exemple l'agrégation ne peut être utilisée et ne signifie absolument rien, vu qu'un employé n'est recruté que par une seule entreprise et ne peut exister que s'il est recruté. Ce qui fait que l'existence d'un employé implique l'existence de l'association Recrute et par conséquent de l'association correspondante.

L'association « occupe » est, en réalité, liée à l'employé et non pas à l'association « Recrute ».

Règle : Quand l'une des pattes de l'association qu'on veut agréger a la cardinalité 1.1, on enlève l'agrégation et toutes les associations auxquelles elle participe seront liées à l'entité se trouvant du côté de la cardinalité 1.1 (le fils).

Remarque : Quand la cardinalité minimale du côté de l'agrégation est un 1 signifiant que l'association agrégée ne peut exister que si l'association à laquelle participe l'agrégation existe. L'agrégation n'a pas lieu d'être car ceci signifie que l'entité liée à l'agrégation n'est en réalité qu'un participant de l'association agrégée et que les deux associations (agrégée et l'externe) ne forment qu'une seule association.

Exemple :

Dans l'exemple ci-dessus, le mariage ne peut exister que si le couple possède un appartement ce qui revient à dire que les deux associations « possède » et « marier » ne forment qu'une seule association. On obtient donc le modèle suivant :

Comme un appartement n'est possédé que par un seul couple, ceci revient à dire que l'appartement est possédé par deux personnes qui sont via cette possession mariées ensemble. L'association « marier » disparaît donc au profit de l'association possède puisqu'elle y est incluse. La solution devient donc :

Règle : Une agrégation n'est justifiée que si sa cardinalité minimale dans l'association à laquelle elle participe est un 0.

Règle générale : Une agrégation n'est justifiée que si aucune des pattes de l'association agrégée n'a une cardinalité 1.1 et la cardinalité minimale de l'association à laquelle elle participe est un 0.

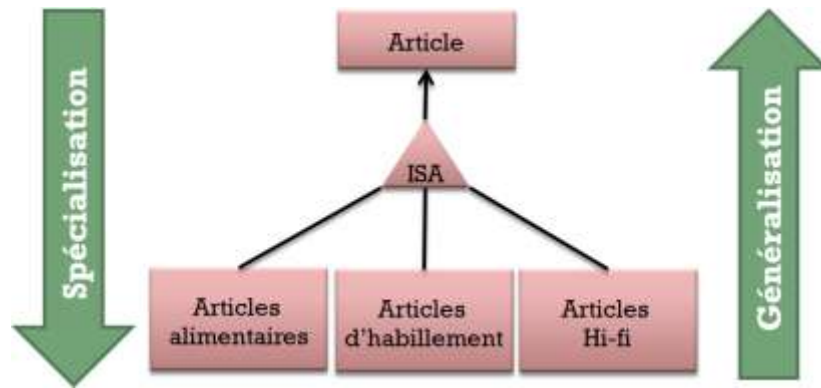
I.5.8. Généralisation / spécialisation

Exemple: dans un hypermarché, certains traitements s'appliquent à tous les articles: inventaire, recherche des caractéristiques, ...

Pour d'autres usages (ventes promotionnelles), on peut vouloir séparer les articles en plusieurs classes (alimentation, habillement, ...). Chaque classe peut avoir des caractéristiques qui lui sont propres, par exemple: date limite de vente (alimentation), taille et couleur (habillement),

Dans le modèle E-A correspondant, on sera donc amené à décrire, en plus du TE générique **Article**, des TE plus spécialisés, représentant les sous-classes "intéressantes".

1. Article alimentaire,
2. Article d'habillement
3. Article Hi-Fi.



Définition 16 : Généralisation : regrouper les différents types d'entité en faisant abstraction de leurs différences. Ce qui nous donne un type générique (mise en facteur des attributs communs).

Définition 17 : Spécialisation : pour un type donné, on définit des sous-types en mettant en évidence leurs particularités.

Définition 18 : Processus de Généralisation : Processus d'abstraction consistant à généraliser les entités, et les ensembles d'entités, en un seul ensemble ascendant.

Définition 19 : Processus de Spécialisation : Les sous-ensembles d'entités dans une hiérarchie de généralisation résultent du processus de spécialisation.

Note : La généralisation/Spécialisation est utilisée dans deux cas uniquement : quand un sous ensemble d'entités d'un type entité ont des caractéristiques en plus où participent à des traitements (donc des relations) différentes.

II. Les Contraintes d'Intégrité

II.1. Objectif

Spécifier des propriétés sémantiques du réel perçu qui ne sont pas exprimables avec le modèle E.A.

II.2. Définition :

“Une contrainte d'intégrité (C.I.) est une propriété non représentée par les concepts de base du modèle E.A. que doivent satisfaire les données appartenant à la base de données”.

Une BD est cohérente si toutes les CI définies sont respectées par les valeurs de la BD.

II.3. Effet

Limiter les occurrences possibles des structures d'information.

II.4. Formulation des contraintes

Les contraintes d'intégrité doivent être décrites explicitement (avec un langage approprié tel que la logique du premier ordre) si elles ne peuvent pas être décrites avec les concepts du modèle de données.

II.4.1. Un formalisme inspiré de la logique du premier ordre

$\forall x, y \in \text{Personne}, \langle x, y \rangle \in \text{Mariage} \Rightarrow x.\text{état-civil} = \text{"marié"} \wedge y.\text{état-civil} = \text{"marié"}$

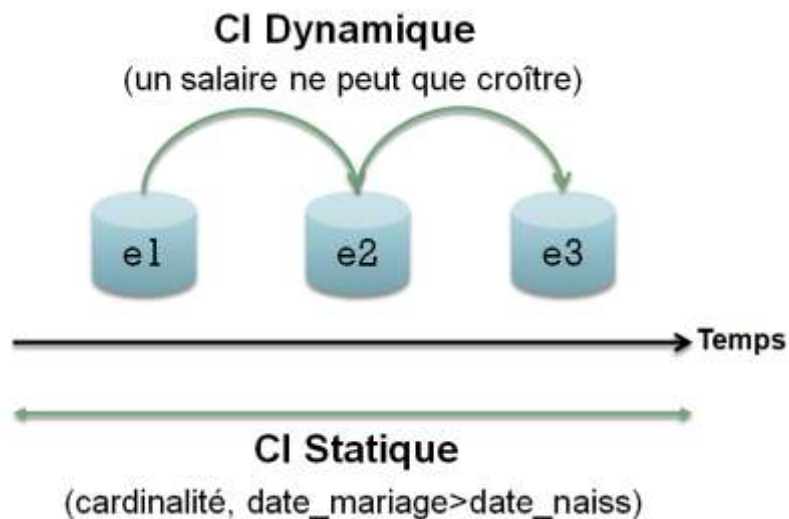
$\forall x, y \in \text{Personne}, \langle \text{époux}:x, \text{épouse}:y \rangle \in \text{Mariage} \Rightarrow x.\text{sexe} = \text{"M"} \text{ et } y.\text{sexe} = \text{"F"}$

$\forall x \in \text{Personne}, \forall t_1, t_2 \in \text{Temps}, t_2 > t_1 \wedge x(t_1).\text{état-civil} = \text{"marié"} \Rightarrow x(t_2).\text{état-civil} \neq \text{"célibataire"}$

II.4.2. Représentation graphique

II.5. Types de Contraintes d'intégrité

On distingue deux types de contraintes au point de vue de temps, c'est-à-dire du moment où elles doivent être vérifiées : Des contraintes statiques ou dynamiques.



Une contrainte statique est une propriété qui doit être vérifiée à tout moment alors qu'une contrainte dynamique est une propriété que doit respecter tout changement d'état de la BDD.

Du point de vue des parties du modèle sur lesquelles les contraintes peuvent être définies, on distingue :

II.5.1. Contraintes d'intégrité sur les attributs

La première contrainte qui peut être définie sur les attributs est le domaine de définition de chaque attribut. Ensuite, si l'attribut est obligatoire ou facultatif en fin on peut faire des restrictions sur les valeurs que peut prendre un attribut grâce à une formule à respecter. La comparaison peut être définie par rapport à une constante ou bien

par rapport à la valeur d'un autre attribut dans la même entité ou d'une autre entité différente. Exemple : âge entre 1 et 100 ou date de naissance < date de mariage

II.5.2. Contraintes d'intégrité sur les cardinalités

Les cardinalités elles-mêmes sont considérées comme des contraintes très importantes dans le schéma conceptuel. Mais il peut également exister des contraintes entre les cardinalités définies pour les différents rôles que joue une entité dans une ou plusieurs associations distinctes, par exemple : Nombre d'enfants d'un parent = nombre d'occurrences de «est parent de» qui lient ce Parent.

II.5.3. Contraintes sur les entités / association

Des contraintes qui peuvent être définies pour les entités comme pour les associations. Car les deux concepts peuvent être définis comme étant des ensembles d'éléments de même nature. On en distingue quatre contraintes.

II.5.3.1. Notions de base

II.5.3.1.1. Couverture

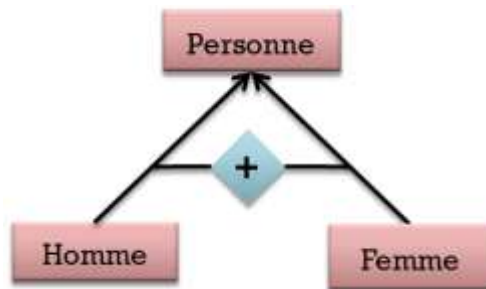
Définition 20 : Ensemble recouvrant tous les éléments.

II.5.3.1.2. Disjonction

Définition 21 : Pas d'éléments communs à deux sous-ensembles.

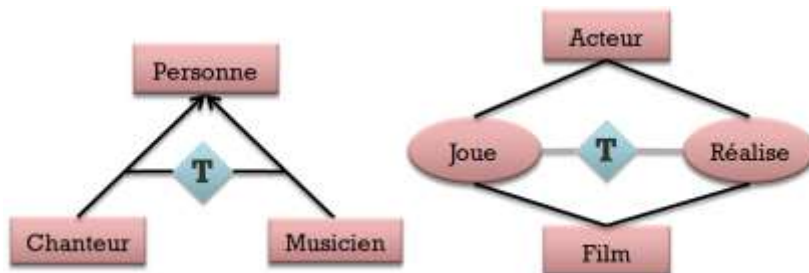
II.5.3.2. Partition

Définition 22 : disjonction + couverture (l'un ou l'autre mais pas les deux et pas autres choses)



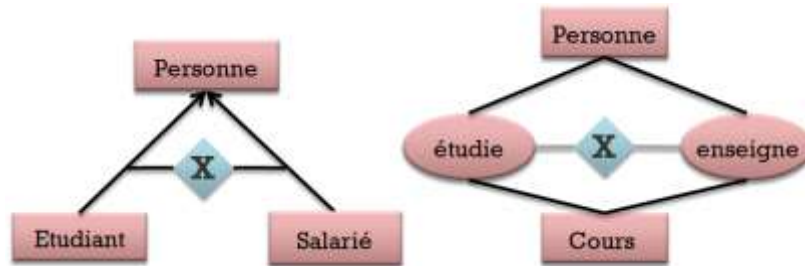
II.5.3.3. Totalité

Définition 23 : ¬disjonction + couverture (l'un ou l'autre ou les deux et pas autres choses)



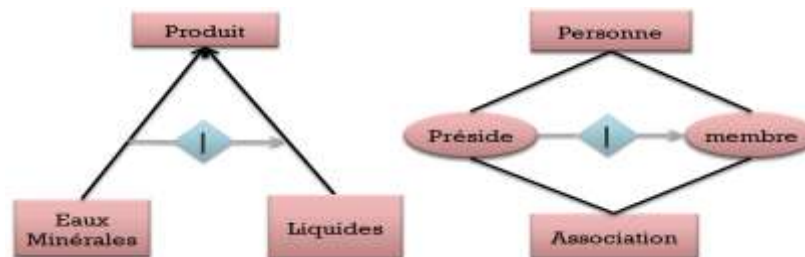
II.5.3.4. Exclusion

Définition 24 : disjonction + \neg couverture (l'un ou l'autre mais pas les deux et autres choses)



II.5.3.5. Inclusion

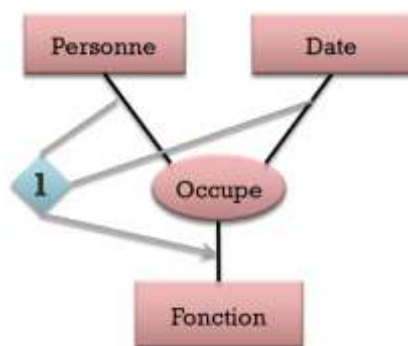
Définition 25 : Tous les éléments du premier ensemble sont inclus dans le second ensemble.



II.5.3.6. Unicité

Définition 26 : c'est ce qu'on appelle la dépendance fonctionnelle. Elle exprime le fait qu'on connaissant la valeur d'un objet je suis en mesure de déduire une et une seule valeur de l'objet dépendant du premier.

Dans l'exemple ci-après : Une personne à une date donnée ne peut occuper qu'une seule fonction donc si je connais la personne et la date je serai en mesure de déterminer avec exactitude une et une seule fonction. Ce qui signifie qu'il y a dépendance entre personne et date vers fonction.



II.5.3.7. Participants à une contrainte

Les contraintes entre entités ou entre associations ne sont possibles que si les deux entités ou les deux associations sont de même nature. Ce qui signifie que pour les entités, le seul cas où les contraintes sont possibles c'est dans le cas de la généralisation/spécialisation. Dans le cas des associations, il faut

vérifier que les deux associations ont le même type d'occurrences. Ceci est possible dans le cas des associations plurielles. Mais quand on veut mettre une contrainte entre deux associations qui n'ont pas les mêmes types d'occurrences mais juste quelques éléments en commun, ces derniers doivent être mentionnés avec des pointillés.

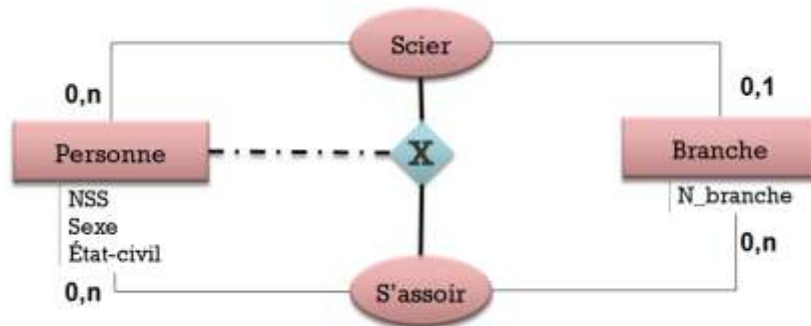
L'exemple suivant montre clairement l'importance et l'influence de l'utilisation des pointillés sur le sens de la contrainte :

Contrainte : On ne s'assoie pas sur une branche qu'on est en train de scier.

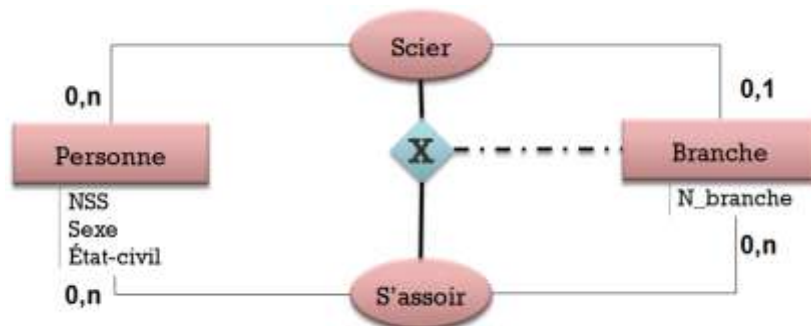
Cette contrainte exprime une situation où des personnes scient des branches avec une restriction qui dit qu'on ne scie par une branche sur laquelle on est assis. Il est clair qu'il existe deux associations entre personne et branche qui sont scier et s'asseoir. Et pour exprimer la contrainte mentionnée ci-dessus on doit utiliser une exclusion entre les éléments de l'association s'asseoir et ceux de l'association scier. Comme ce sont des associations plurielles, elles ont le même type d'occurrences. C'est-à-dire que chaque occurrence se compose du couple (personne, branche). La question qu'on se pose ici :

Veut-on faire une exclusion sur personne, branche les deux en même temps où les deux séparément ?

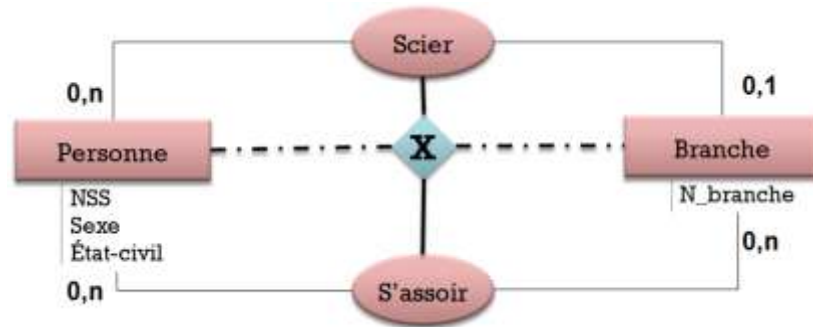
Signification 1 : On ne scie pas assis



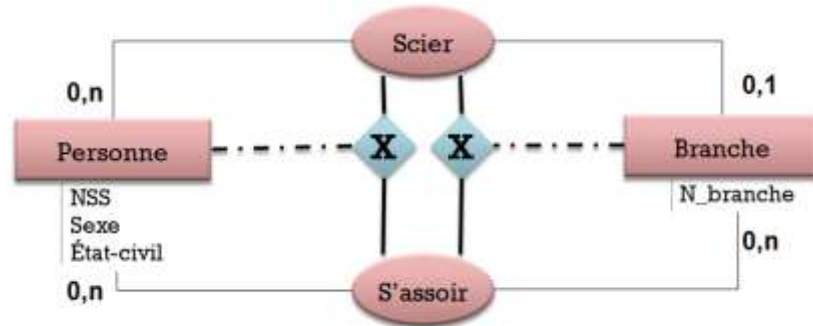
Signification 2 : On ne scie pas une branche sur laquelle on est assis



Signification 3 : On ne scie pas une branche sur laquelle on est soi-même assis



Signification 4 : On ne scie pas assis et on ne scie pas une branche sur laquelle on est assis



Exemple :

III. Le Schéma Conceptuel

Définition 26 : Un schéma conceptuel entité association est un ensemble de descriptions de types d'entité et de types d'association (avec leurs attributs et les liens de généralisation entre types d'entité), et des contraintes d'intégrité (CI) associées: **SEA = ({TE}, {TA}, {CI})**

III.2. Règles de conception

III.2.1. Règle de représentation par un type d'entité

Est représenté par un TE tout ensemble d'objets similaires qui a un intérêt en soi pour au moins un traitement de l'application.

III.2.2. Règle de représentation par un type association

Est représenté par un TA tout ensemble de liens similaires et de même sémantique, d'intérêt pour l'application, entre deux ou plusieurs objets représentés par des entités.

III.2.3. Règle de représentation par un attribut

Est représenté par un attribut toute information intéressante qui participe à la description d'un objet ou d'un lien et qui ne fait l'objet de traitement qu'en tant que partie de cet objet ou lien.

III.3. Règles d'optimisation

III.3.1. Règles portant sur les noms

Règle : Dans un modèle entités-associations, le nom d'un type-entité, d'un type-association ou d'un attribut doit être unique.

Règle : Il faut remplacer un attribut multiple en un type-association et un type-entité supplémentaires.

III.3.2. Règles de normalisation des attributs

Règle : Il ne faut jamais ajouter un attribut dérivé d'autres attributs, que ces autres attributs se trouvent dans le même type-entité ou pas.

Règle : Un attribut correspondant à un type énuméré est généralement avantageusement remplacé par un type-entité.

III.3.3. Règles de fusion/suppression d'entités/associations

Règle : Il faut factoriser les TE quand c'est possible.

Règle : Il faut factoriser les TA quand c'est possible.

Règle : Un TE remplaçable par un TA doit être remplacé.

Règle : Lorsque les cardinalités d'un TA sont toutes 1, 1 c'est que le TA n'a pas lieu d'être.

Règle : Il faut éviter les TA redondants. S'il existe deux chemins pour se rendre d'un TE à un autre, il faut supprimer le chemin le plus court puisqu'il est déductible des autres chemins.

Règle : Lorsque l'une des cardinalités d'une ternaire est (1,1) ou (0,1), celle-ci est éclatée en deux autres associations à partir de cette patte.

Règle : toute TA n-aire peut être remplacé par un TE en lui attribuant un identifiant et en créant des TA binaires entre le nouveau TE et tous les TE de la collection de l'ancien TA n-aire. La cardinalité de chacun des TA binaires créés est 1, 1 du côté du TE créé et 0, n ou 1, n du côté des TE de la collection de l'ancien TA n-aire.

III.4. Vérification du schéma EA

III.4.1. Vérification "syntaxique"

Il s'agit de vérifier que les règles du modèle entité association sont respectées

III.4.2. Par jeu d'essai

Le concepteur vérifie grâce à une mini base de données que le diagramme permet effectivement de stocker les informations nécessaires à l'entreprise.

III.4.3. Complétude par rapport aux traitements

Le concepteur vérifie que le diagramme contient tous les types d'information nécessaires à l'exécution des traitements prévus.

III.4.4. Par les utilisateurs

Le concepteur présente le diagramme accompagné des définitions aux personnes qui utiliseront la base de données et vérifie que les informations contenues correspondent bien aux besoins.

III.5. Description d'un schéma EA

III.5.1. Entête

Titre :

Description :

Date de création :

Date de mise à jour :

Version :

Concepteur(s) :

Propriétaire :

Corps .../...

III.5.2. Description d'un TE

Nom :

Nom du (ou des) TE sur-type de ce TE, s'il en existe :

Définition libre (commentaire) précisant la sémantique du TE :

Description des attributs du TE :

Composition des identifiants du TE :

Contraintes d'intégrité propres au TE :

III.5.3. Description d'un TA

Nom du type d'association

Définition libre (commentaire) précisant la sémantique du TA

Noms des TE participant au TA, avec le nom du rôle les associant au TA et pour chaque rôle sa cardinalité.

Description des attributs du TA, s'il en existe

Composition des identifiants du TA

Contraintes d'intégrité propres au TA

III.5.4. Description d'un Attribut

Nom de l'attribut

Définition libre de sa sémantique

Domaine de valeurs

Contrainte sur les valeurs de l'attribut

IV. Conclusion

IV.1. Limites du modèle E/A

Le modèle E/A est un modèle de données et non de traitement, qui ne permet pas de prendre en compte l'aspect dynamique des données, les modalités de déclenchement d'une opération ainsi que la façon dont la modification est réalisée.

IV.2. Inconvénients du modèle E/A

Le concepteur éprouve des difficultés à faire des choix parce qu'il ne discerne pas toujours les conséquences des décisions prises. Il y a souvent des ambiguïtés entre entité et attribut, ce qui pousse parfois à devoir créer des types-entités artificiels. Enfin il manque d'un support formel pour contrôler la qualité du schéma.

IV.3. Avantages du modèle E/A

L'approche paraît plus naturelle : obtention directe de résultats de synthèse, simplicité du support graphique, outil de communication efficace, la représentation graphique est accessible à tous les intéressés, qu'il s'agisse des utilisateurs finaux, des partenaires du processus de conception ou des informaticiens responsables de la mise en œuvre technique.