Chapitre 3
Construction d'un schéma EA

I. Introduction

Afin de pouvoir construire le schéma entité-association répondant aux besoins d'une entreprise, le concepteur étudie les différents modèles existants, les dsifférents besoins et supports de l'entreprise en recensant les divers documents… Afin de pouvoir réunir les différentes connaissances de l'entreprise et lui permettre de gérer au mieux les employés et les différents services, le concepteur procède à différentes taches :

- Etablir la liste des entités ;
- Lister leurs attributs et déterminer les identifiants ;
- Identifier les différentes associations reliant ces entités en établissant les différentes cardinalités ;
- Lister leurs attributs ;
- Établir les règles d'intégrité ;
- Vérifier la cohérence et la pertinence du schéma obtenu.

Exemple :
Dans une agence bancaire, le directeur explique : « Nous gérons des comptes bancaires, mais aussi nous permettons aux clients d'établir une épargne et d'avoir une assurance. Les comptes bancaires, épargnes et assurances possèdent un numéro d'identification et d'une date de création. Le client possède lui aussi un numéro d'identification. Nous permettons à un client de réaliser diverses opérations sur ses comptes bancaires et épargne, et nous luis envoyons une fois par mois, à une date précise, le relevé de ses comptes. »

De l'exposé précédent on peut extraire les informations suivantes :

- *quatre TE : Client, Compte, Opération, Relevé.*
- *N° d'identification du client : attribut du TE Client*
- *N° d'identification, Type et Date de création du compte : attributs du TE Compte*
- *un TA : Ouvrir entre Client et Compte ; cette association a pour cardinalité (0,N) pour le TE Compte*

Les choix de représentation dépendent du point de vue du concepteur. C'est donc seulement en examinant l'utilisation des données qu'il est possible de déterminer la représentation adéquate.

II. Règles de modélisation

Règle de représentation par un type d'entité
Tout ensemble d'objets similaires utiles pour l'application est représenté par une entité.

Règle de représentation par un type d'association
Tout ensemble de liens similaires et de même sémantique, utiles pour qualifier l'application, entre deux ou plusieurs objets représentés par des entités est représenté par une association.

Règle de représentation par un attribut
Toute information utile et descriptive d'un objet ou d'un lien, ne faisant l'objet d'aucun traitement en dehors de cet objet ou de ce lien est représentée par un attribut.

Remarque importante :
Par rapport à la réalité, un schéma est une représentation :

- **incomplète** : il ne représente que les informations qui sont jugées utiles pour l'application ;
- **partiale** : il représente le point de vue du concepteur ;
- **infidèle** : il ne représente pas la réalité telle qu'elle est, mais telle qu'elle intéresse le concepteur.

III. Vérification du diagramme EA

Une fois le diagramme EA établi, plusieurs types de vérifications sont effectués :

- **vérification syntaxique** : il s'agit de vérifier que les règles du modèle EA sont respectées ;
- **par jeu d'essai** : vérifier grâce à une mini BD que le diagramme permet effectivement de stocker les informations nécessaires à l'entreprise ;
- **complétude par rapport aux traitements** : vérifier que le diagramme contient tous les types d'informations nécessaires à l'exécution des traitements prévus ;
- **par les utilisateurs** : présenter le diagramme accompagné des définitions aux futurs utilisateurs et vérifier que les informations contenues correspondent bien aux besoins.

Chaque oubli, erreur, modification, …, détecté lors des vérifications entraîne une mise à jour du diagramme et relance les différentes phases de vérification.

IV. Notion de dépendance

Pour un TE (ou un TA) donné, on dit qu'il y a **dépendance** d'un attribut, ou un ensemble d'attributs, *A* vers un attribut, ou un ensemble d'attributs, *B*, si toutes les occurrences qui ont même valeur pour *A* ont toujours même valeur pour *B* : on note cette dépendance $A \rightarrow B$, et on dit que *B* **dépend** de *A* ou que *A* **détermine** *B*. Par définition, l'identifiant d'un TE (TA) détermine tous les autres attributs du TE (TA).

Formellement, quelles que soient deux occurrences du TE (TA) de valeurs (*a*,*b*) et (*a*',*b*') pour (*A*,*B*) :

$$a = a' \Rightarrow b = b'$$

Soient E_1 et E_2 deux entités liées par une association A. On dit qu'il y a dépendance de E_1 vers E_2 (notée $E_1 \rightarrow E_2$), si pour chaque occurrence de E_1, l'association A lui associe toujours la même occurrence de E_2.

V. Règles de validation d'un diagramme EA

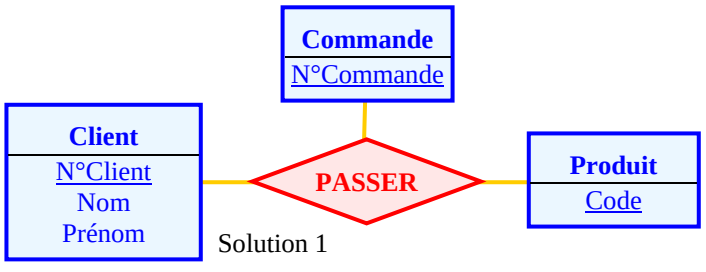
L'élaboration d'un schéma EA se fait en plusieurs étapes : une de celles-ci est celle qui consiste à vérifier le schéma en utilisant un certain nombre de règles dites de vérification et de normalisation. Ces règles permettent d'obtenir un schéma dans lequel les erreurs de mises à jour, d'insertion et de suppression, ainsi que les redondances logiques, sont les plus limitées possibles.

1. Règles concernant les entités

Règle n°1 : Existence d'un identifiant pour chaque entité.

Cette règle permet d'aider le concepteur quand celui-ci hésite entre modéliser un concept par une entité ou par une association.

Exemple :
On représente une situation où des clients commandent des produits. Si le concept *commande* inclut un *numéro de commande* unique pour chaque commande, alors il faut modéliser le concept *commande* par une entité – cf. solution 1. Sinon, on peut le modéliser par une association : la clé sera alors le couple clé (*Client*,*Produit*) – cf solution 2.



Solution 1

Solution 2

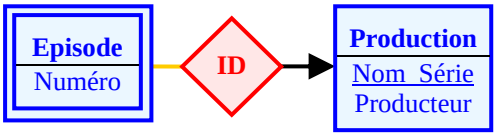
Règle n°2 : Tous les attributs autres que l'identifiant doivent être en dépendance fonctionnelle complète et directe de l'identifiant.

La notion de **dépendance fonctionnelle** est celle du paragraphe précédent : quelle que soit l'instance de l'entité considérée, la valeur de la clé est unique (deux instances différentes ne peuvent pas avoir la même valeur de clé) et pour tous les attributs non clé, il n'existe qu'une seule valeur (les attributs sont monovalués : la valeur d'un attribut est une valeur simple et non un ensemble de valeurs). Ceci correspond à la **première forme normale** qui sera vue dans le chapitre 8 (1NF). La notion de **dépendance complète** stipule que les valeurs des attributs non clé dépendent de la valeur de toute la clé et non d'une partie seulement.

Exemple :

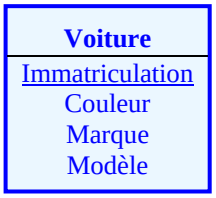
L'entité *Episode*, sur la figure ci-contre, a pour clé le couple d'attributs (*Nom_Série*,*Numéro*). Si on fait l'hypothèse que toute série n'a qu'un *producteur*, alors il y a une dépendance fonctionnelle *Nom_Série* → *Producteur*. Donc *Episode* ne suit pas la règle 2.

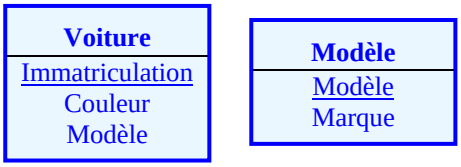
Pour suivre la règle 2, il faut transformer *Episode* grâce à la méthode suivante. On crée une nouvelle entité formée des attributs qui constituent la DF, et on enlève de *Episode* la partie droite de la DF (c'est à dire l'attribut *Producteur*). Comme, alors, il reste *Nom_Série* et *Numéro* dans Episode, et que *Nom_Série* est clé primaire de *Production*, *Episode* devient une entité faible, reliée à *Production* par une association d'identification. Cette notion de dépendance complète correspond à la **deuxième forme normale** qui sera vue dans le chapitre 8 (2NF). La notion de dépendance directe stipule que tout attribut dépend directement de la clé, c'est à dire qu'il n'y a pas de DF avec une partie gauche différente de la clé.



Exemple :

Dans l'entité *Voiture* sont présents les attributs *Marque* et *Modèle*. Or quand on connaît la valeur de *Modèle*, on connaît forcément la *Marque*. Il y a donc une DF *Modèle* → *Marque* avec comme partie gauche Modèle qui n'est pas la clé. Donc voiture ne suit pas la règle numéro 2.





Comme précédemment, on résout le problème en construisant une nouvelle entité dont les attributs sont les attributs de la DF et en ôtant la partie droite de la DF de l'entité d'origine. On obtient les deux entités à gauche.

Cette notion de dépendance directe correspond à la **troisième forme normale** qui sera vue dans le chapitre 8 (3NF). En résumé, appliquer la règle 2 revient à mettre le schéma EA en troisième forme normale.

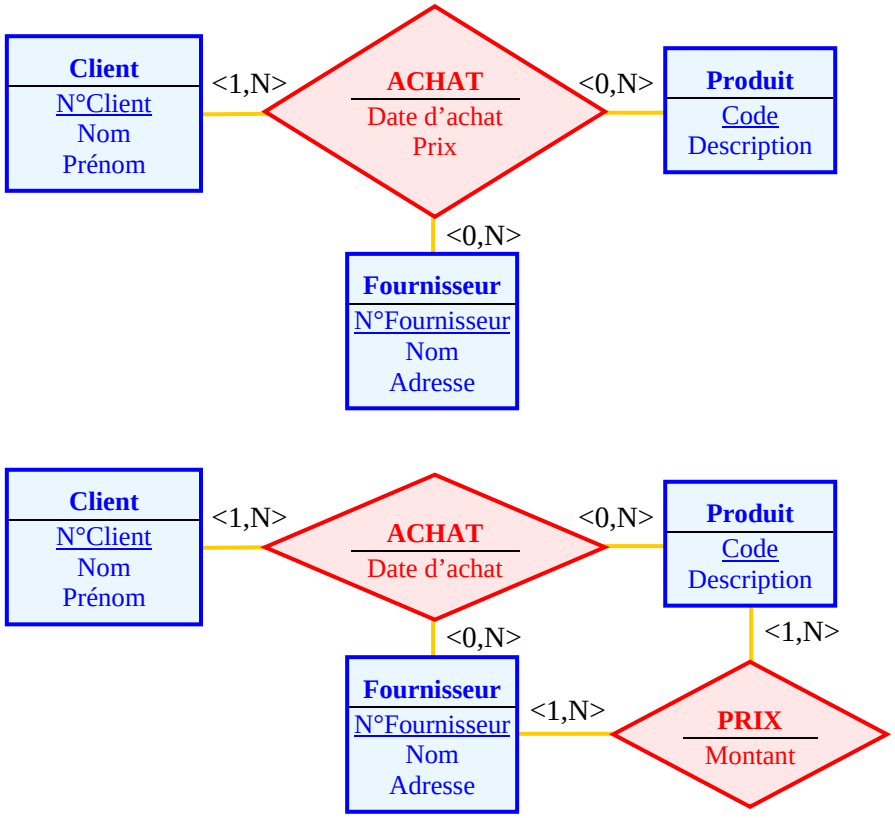
2. Règle concernant les associations :

Règle n°3 : Tous les attributs d'une association doivent dépendre complètement de la clé de cette association, c'est-à-dire de l'ensemble des clés de toutes les entités associées. Chaque attribut doit dépendre de toute la clé et non d'une partie de la clé seulement.

Exemple :

L'association *Achat* du premier schéma relie 3 entités : *Client*, *Produit* et *Fournisseur*. La clé de *Achat* est donc (*N°Client*, *Code*, *N°Fournisseur*). Deux attributs supplémentaires figurent dans *Achat* : *Date d'achat* et *Prix*. Il est clair que la *date d'achat* dépend à la fois du *client* qui achète, du *produit* acheté et du *fournisseur*. Par contre, le *prix* ne dépend que du *fournisseur* et du *produit* (on suppose que le *client* ne peut pas marchander). On a donc la dépendance fonctionnelle (*Code*,*N°Fournisseur*) → *Prix*. Ici, la partie gauche n'est pas toute la clé de *Achat*, mais seulement une partie. Donc *Achat* ne suit pas la règle 3. Pour résoudre le problème, on crée une nouvelle association à partir des attributs de la dépendance fonctionnelle et on enlève la partie droite de la dépendance fonctionnelle de l'association d'origine.

On obtient le deuxième schéma.

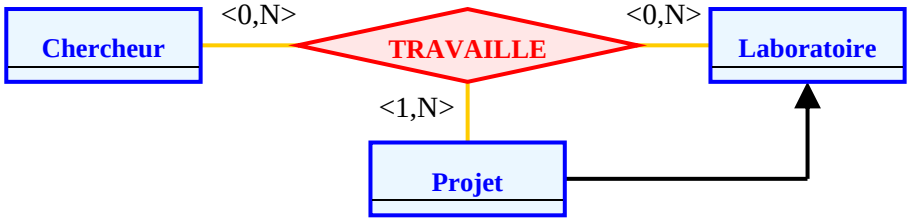


Remarque :
On a stocké ici la *date d'achat* comme attribut de *Achat*. Cela signifie que pour un *achat*, c'est-à-dire pour un triplet (*client*,*produit*,*fournisseur*), il n'y a qu'une seule *date* correspondante. Autrement dit on ne peut pas stocker plusieurs *achats* identiques à deux *dates* différentes. Si on voulait le faire, il faudrait faire une entité spéciale pour *Date* et relier cette entité à l'association *Achat* qui serait alors 4-aire.

3. Validation d'un TA (arité)

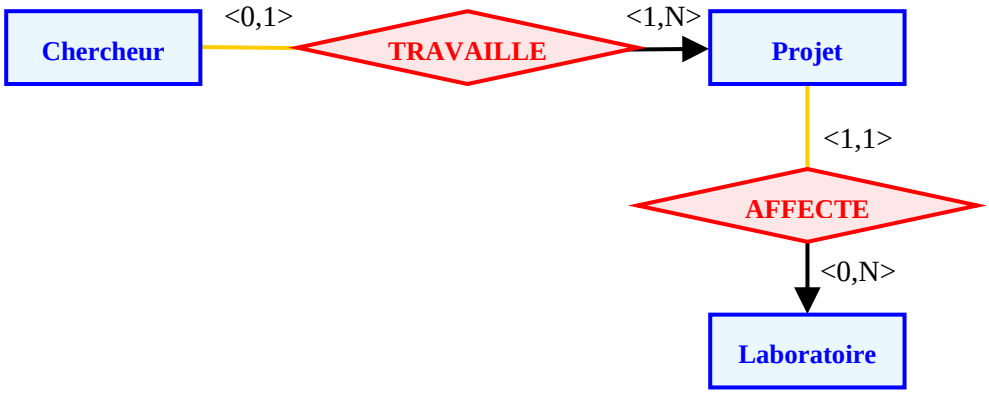
Règle 4 : Soit un TA bien construit, liant les TE $E_1,\ldots,E_n$; s'il y a une dépendance $(E_1,\ldots,E_i) \rightarrow E_{i+1}$, alors il y a la dépendance $(E_1,\ldots,E_i) \rightarrow (E_{i+1},\ldots,E_n)$.

Donc si un TA comporte l'une de ces dépendances sans les autres, il faut décomposer le TA, afin de matérialiser cette dépendance par un nouveau TA de cardinalité maximum 1 pour le rôle source de la dépendance.



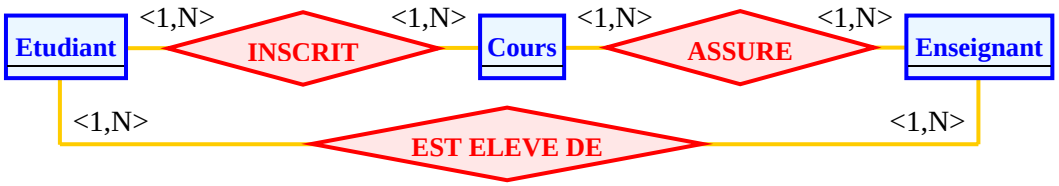
Dans l'exemple ci-dessus, cette dépendance traduit la règle d'entreprise : chaque projet est réalisé dans un et un seul laboratoire. Le diagramme précédent n'est pas correct. La dépendance doit être décrite par un TA binaire reliant les deux TE Projet et Labo, le TE restant sera lié par un TA binaire au TE source de la dépendance.

La correction du diagramme est donnée ci-après.



4. *Elimination des types d'association (TA) redondants*

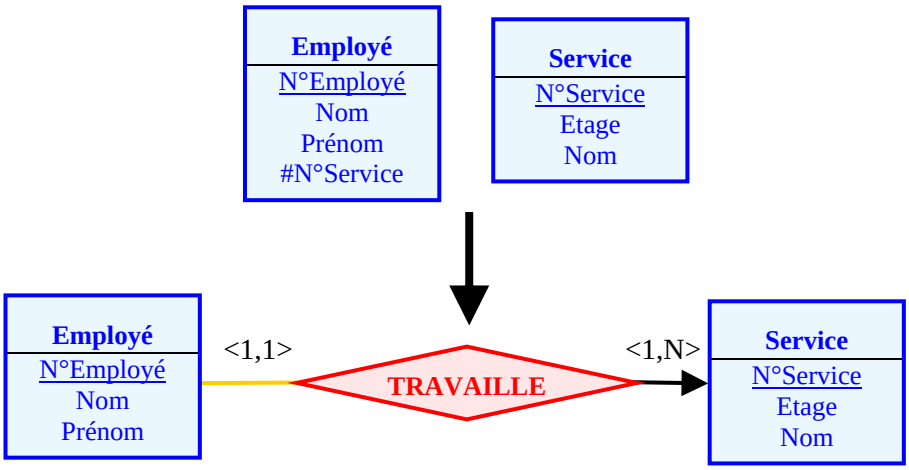
Un TA est redondant si les associations correspondantes peuvent être établies sans ambiguïté par composition des associations d'autres TA



Si pour l'application on veut conserver un TA redondant, il faut alors déclarer par une contrainte d'intégrité que ce TA est dérivé d'autres.

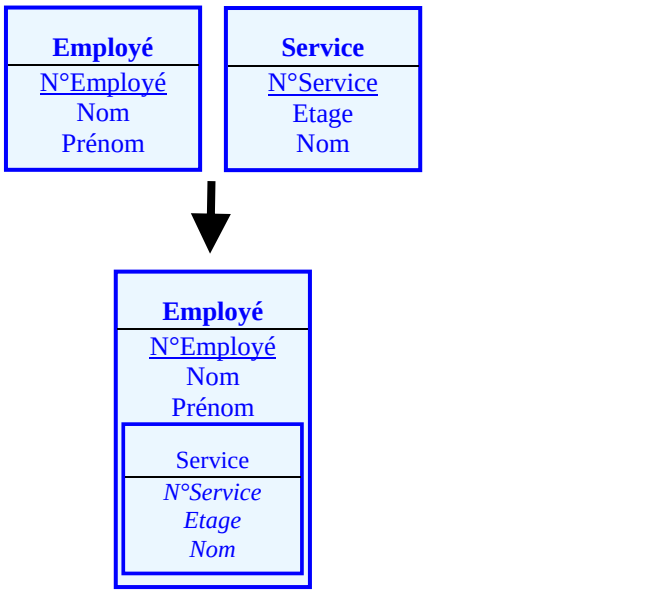
5. *Transformation des attributs référence*

Si l'on trouve dans un TE E_1 un attribut dont la valeur est égale à celle de l'identifiant d'un TE E_2, cet attribut exprime un lien entre E_1 et E_2. Ce lien doit être explicitement décrit comme une association entre les deux TE et l'attribut supprimé de E_1.

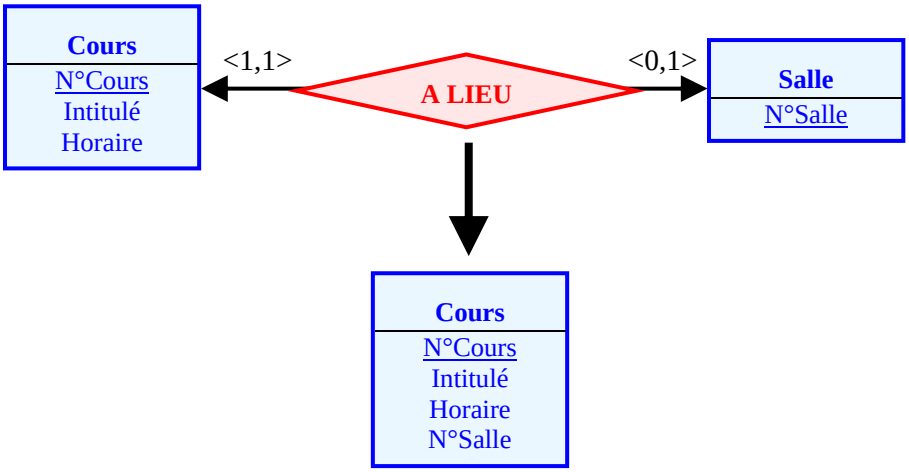


6. *TE ou attribut complexe ?*

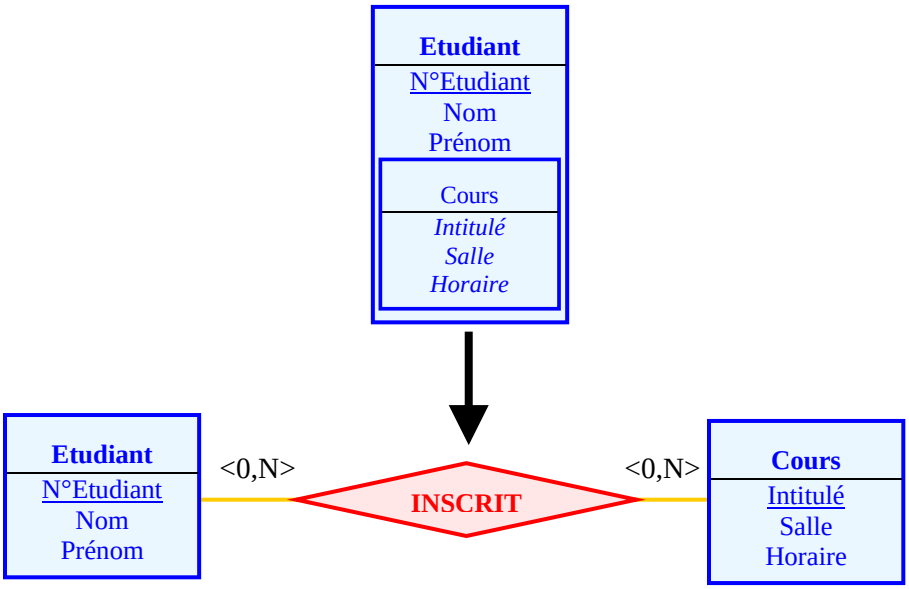
Si un TE retenu a priori se révèle ne présenter d'intérêt pour aucun traitement de l'application mais qu'il faut néanmoins conserver l'information correspondante, il convient de rattacher celle-ci, sous forme d'un attribut, au TE (TA) où elle est pertinente.



Un TE qui n'a qu'un seul attribut doit être transformé en attribut des TE auxquels il est relié.



Si un TE possède un attribut complexe qui fait l'objet de traitements indépendamment du TE en question, il faut le remplacer par un TE, plus un TA qui conserve le lien avec le TE d'origine.



VI. Description d'un schéma EA

1. Entités :

Une entité est décrite par :
- le **nom** du type d'entité ;
- une **définition** précisant la population exacte de l'entité ;
- la **description** des attributs de l'entité ;
- la composition des **identifiants** de l'entité, s'il en existe.

Exemple (TE *Employé* de la BD *hypermarché*) :
- nom : *Employé*
- définition : *« toute personne employée de l'entreprise en ce moment »*
- attributs : *nom, salaire (avec leur description)*
- identifiant : *nom*

Remarque :
Deux TE différents peuvent avoir le même nom. La définition libre est une partie importante de la description du TE et permet de définir exactement, de façon non ambiguë, la population du TE.

2. Associations :

Une association est décrite par :
- le **nom** du type d'association ;
- une **définition** précisant la population exacte du TA ;
- les noms des **entités participant** à l'association, avec le nom du rôle si nécessaire ;
- pour chaque rôle, ses **cardinalités** : $min \geq 0$, $max \geq 1$;
- la description des **attributs** du TA, s'il en existe ;
- la composition des **identifiants** du TA, s'il en existe.

Exemple (TA *Emploi* de la BD *hypermarché*) :
- nom : *Emploi*
- définition : *« lie un employé au rayon dans lequel celui-là travaille aujourd'hui »*
- TE participants : *Employé, Rayon*
- cardinalités : *Employé: 0:1 Rayon: 0:n*
- identifiants : *(Employé.nom + Rayon.nomR)*

3. Attributs :

Un attribut est décrit par les spécifications suivantes :
- le **nom** de l'attribut ;
- une **définition** (libellé clair) ;
- la **cardinalités** ;
- le **domaine** de valeurs si l'attribut n'est pas composé d'autres, la **description des attributs composants** sinon.

Exemple (attribut « date de naissance » d'un TE Personne) :
- nom : *date de naissance*
- définition : *« date de naissance de la personne »*
- cardinalités : *min = 1, max = 1*
- composition :
  - nom : *jour* ; définition : *« jour de naissance de la personne »* ; cardinalités : *min = 1, max = 1* ; domaine : *l'intervalle entier [1,31]*
  - nom : *mois* ; définition : *« mois de naissance de la personne »* ; cardinalités : *min = 1, max = 1* ; domaine : *l'intervalle entier [1,12]*
  - nom : *année* ; définition : *« année de naissance de la personne »* ; cardinalités : min = 1, max = 1 ; domaine : *l'intervalle entier [1870,2004]*