

Chapitre II : Modèle Entité/Association

Introduction : La conception d'une base de données, passe par un certain nombre d'étapes allant du cahier des charges au schéma de la base. Plusieurs formalismes peuvent être utilisés pour concevoir une base de données. Dans ce chapitre, nous allons utiliser le MCD (modèle entité/association) du formalisme Merise. Le modèle entité/association est un modèle de conception utilisé en mode graphique. Il est indépendant des possibilités logiques et physiques de SGBD sur lequel la base sera implémentée. Il repose essentiellement sur trois concepts de base qui sont : l'entité, l'association et les contraintes portant sur elles.

I. Concepts généraux

I. 1. La notion d'entité

Une entité est la représentation d'un objet matériel (concret) ou immatériel (abstrait) du monde réel. Elle permet de caractériser les objets les plus significatifs du système que l'on doit modéliser. Les entités de même type et qui sont caractérisées par les mêmes propriétés sont regroupées dans un ensemble d'entités appelé classe d'entités. Une classe d'entités est donc constituée de plusieurs entités de même type et décrites par les mêmes propriétés.

Dans la suite du cours, nous parlons d'entité à la place de classe d'entités.

Exemples :

Entités concrètes : un élève, une table, une maison, etc.

Entités abstraites : un cours, une cérémonie, un entretien, etc.

I. 1. 1. Attributs d'une entité

Les entités étant les objets les plus importants du système à modéliser, le concepteur se doit de les définir avec le plus de soins possibles. Pour ce faire chacune d'elles possède un certain nombre de propriétés (attributs) permettant de la caractériser. Chaque attribut donne une information concrète sur l'entité.

Exemples :

- Un arbre peut être défini par : son nom, sa famille, son âge, sa hauteur, etc.
- Un événement peut être défini par : sa date, son heure, son objet, sa durée, etc.

I. 1. 2. Domaine d'un attribut

Pour chaque attribut, il y a un ensemble de valeurs que les occurrences peuvent prendre. Cet ensemble est appelé domaine de l'attribut. Le domaine d'un attribut est un type de données (type de base) auquel s'ajoute éventuellement certaines contraintes supplémentaires (taille du champ, précision pour un numérique, contraintes de validité restreignant les valeurs possibles (intervalle, liste, format, etc.), présence ou non de la valeur NULL, etc.).

Exemples :

- La note d'un étudiant est un réel appartenant à l'intervalle $[0, 20]$;
- Le jour d'un entretien est une chaîne de caractères de taille maximum 8 et appartenant à la liste (Lundi, Mardi, Mercredi, Jeudi, Vendredi, Samedi) ;
- Les noms de famille des personnes sont des chaînes de 15 caractères au maximum ;
- L'âge d'une personne est un entier supérieur ou égal à zéro et inférieur à 150.

Remarque : Un attribut dont la valeur peut être obtenue à partir de valeur(s) d'un ou de plusieurs autres attributs est appelé attribut calculé. Ce type d'attribut est à éviter dans un modèle entité/association. Par exemple, la moyenne obtenue par un étudiant pour une matière donnée peut être calculée à partir de ses notes de contrôle continu et d'examen.

I. 1. 3. L'identifiant d'une entité

I. 1. 3. 1. Définition

Un sous-ensemble non vide de l'ensemble des attributs d'une entité permet de distinguer les différentes occurrences de cette entité. Dès que l'on connaît la valeur prise par ce sous-ensemble, on peut désigner de manière unique et non ambiguë une occurrence de cette entité. Ce sous-ensemble est appelé identifiant de l'entité. Deux occurrences distinctes d'une même entité ne peuvent pas prendre la même valeur d'identifiant. L'identifiant d'une entité est unique et obligatoire.

I. 1. 3. 2. Dépendance fonctionnelle

Il y a dépendance fonctionnelle entre deux attributs d'une entité si à chaque valeur de l'un correspond une et une seule valeur de l'autre.

Par exemple connaissant le nom du centre et le numéro d'un bureau, on connaît le nombre d'inscrits, le type, le président, etc.

Centre, Numéro \rightarrow Nb_Inscrit, Type, Président.

L'ensemble contenant la/les propriété(s) de gauche (Bâtiment, Salle) est appelé **source** et celui contenant la/les propriété(s) de droite (Capacité, Climatisé, Type) est appelé **but**. Une dépendance fonctionnelle dont la source est composée d'un seul attribut est appelée DF simple. Si la source est composée de plusieurs attributs on a une dépendance fonctionnelle composée.

- a. **Dépendance directe** : Une dépendance fonctionnelle $X \rightarrow Y$ est dite directe s'il n'existe pas un attribut Z tel que les dépendances fonctionnelles $X \rightarrow Z$ et $Z \rightarrow Y$ soient correctes.
- b. **Dépendance élémentaire** : Une dépendance fonctionnelle est dite élémentaire si sa source ne contient pas d'attribut superflu. Si $X, Y \rightarrow Z$ et $X \rightarrow Z$ alors Y est superflu dans la dépendance $X, Y \rightarrow Z$ qui n'est pas élémentaire.

I. 1. 3. 3. Recherche de l'identifiant d'une entité

Pour choisir l'identifiant d'une entité on procède comme suit :

1. On construit toutes les DF possibles entre les attributs de cette entité ;
2. On élimine les DF dans lesquelles il y a des attributs manquants ;
3. On élimine les DF dont la source contient plus d'attributs que celle d'un autre DF ;
4. Enfin, la source d'une des DF restantes peut être choisie comme identifiant de l'entité.

Les dépendances fonctionnelles permettent aussi de modéliser les associations.

Exemple :

- Le nom permet d'identifier un centre. *Nom* est un identifiant absolu de l'entité **Centre**;
- Le numéro permet d'identifier un bureau dans le centre dans lequel il se trouve. *Numéro* est un identifiant relatif de l'entité **Bureau**.

I. 2. La notion d'association entre entités

Une association permet de représenter les liens sémantiques qui peuvent exister entre deux ou plusieurs entités. Les associations de même type sont regroupées dans une classe d'associations. Une classe d'associations peut être :

- **Réursive** : Elle relie une entité avec elle-même ;
- **Binaire** : Elle relie deux entités différentes entre elles ;
- **N-aire** : Elle relie n entités différentes (n supérieur à 2).

Remarque :

Si une association est récursive, on donne les rôles aux extrémités. Sur la Figure 1, Parent et Enfant sont les rôles.



Figure 1 : Association récursive

Exemple :

Soient les entités *Etudiant*, *Cours* et *Enseignant* :

- Une relation *Suivre* peut relier les entités *Etudiant* et *Cours* ;
- Une relation *Dispenser* peut relier les entités *Enseignant* et *Cours*.

Dans la suite du cours nous utilisons association à la place de classe d'associations.

Remarque :

Il est possible dans le cas où un attribut ne définit pas directement les entités mais plutôt l'action qui les lie, de le mettre dans l'association entre elles. Une telle association est appelée association porteuse.

Exemple :

L'association *Dispenser* entre *Enseignant* et *Cours* peut prendre l'attribut *Date*.

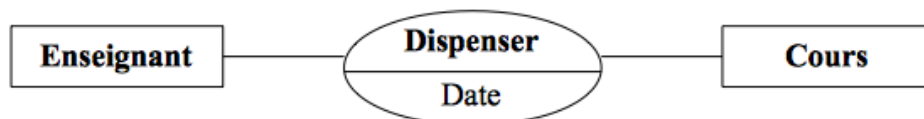


Figure 2 : Association binaire porteuse

I. 2. 1. Les cardinalités d'une association

Chaque association possède des multiplicités à chacune de ses extrémités. Les multiplicités donnent le nombre d'occurrences d'une entité en liaison avec une occurrence de l'autre. Elles sont données sous forme de couple (Val_{min} , Val_{max}).

La valeur minimale (resp. maximale) décrit le nombre minimum (resp. maximum) d'occurrences d'une l'entité en liaison avec une occurrence de l'autre.

Exemple :

- Un électeur d'inscrit au minimum dans un bureau, au max dans un bureau (1, 1) ;
- Dans un bureau s'inscrivent plusieurs électeur au minimum, plusieurs électeurs au max (n, m) ;
- Un centre contient au minimum un bureau, au max plusieurs bureaux (1, n) ;
- Un bureau est dans un centre au minimum, dans un centre au max (1, 1).

Remarque :

- Il faut éviter autant que possible les associations reliant plus de deux entités dont la cardinalité a pour valeur maximale 1 dans un sens. Si ce cas se présente, il faut essayer de le casser en plusieurs associations binaires.
- S'il y'a une association binaire entre deux entités E_1 et E_2 avec une cardinalité (1, 1) du côté de E_2 et ayant un/des attribut(s), il faut intégrer ses attributs dans cette l'entité.
- Il est souhaitable d'éviter la valeur *plusieurs* comme cardinalité minimale. Si ce cas de figure se présente, elle peut être remplacée par la valeur 1.

I. 2. 3. Les associations spécifiques**I. 2. 3. 1. La généralisation/spécialisation**

Deux entités peuvent être caractérisées par plusieurs attributs communs (même nom, même domaine). Dans ce cas on peut regrouper dans une entité indépendante ces attributs et ne garder dans chaque entité que ses attributs spécifiques. L'entité dans laquelle les attributs communs sont regroupés est appelée entité générale. Les deux entités dont une partie des attributs est représentée dans l'entité générale sont des entités spécialisées.

Les entités spécialisées n'ont pas besoin d'identifiant car elles héritent de l'identifiant de l'entité générale.

- **Spécialisation complète :** Si toutes les occurrences de l'entité générale appartiennent à l'instance de l'une des entités spécialisées, on dit que la spécialisation est complète.
- **Spécialisation incomplète :** Si certaines occurrences de l'entité générale n'appartiennent à l'instance d'aucune des entités spécialisées, la spécialisation est incomplète.

I. 2. 3. 2. L'agrégation

Si dans un système, les occurrences d'une entité ne peuvent être distinguées que dans une partie de ce système, on parle alors d'agrégation entre cette entité et ce sous système. L'agrégation permet donc de matérialiser le fait qu'une entité (sous-système) soit nécessaire pour en identifier une autre. L'entité représentant le sous-système est appelée entité *forte* et celle qu'elle identifie est appelée entité *faible*. L'identifiant de l'entité *faible* est appelé *identifiant relatif*. L'identifiant de l'entité forte est appelé *identifiant absolu*.

II. Représentation d'un modèle entité/association

Nous avons la représentation entité/association standard et la représentation Merise qui est la plus utilisée actuellement.

II. 1. Représentation standard

Les éléments utilisés pour la représentation standard des entités et des associations sont :

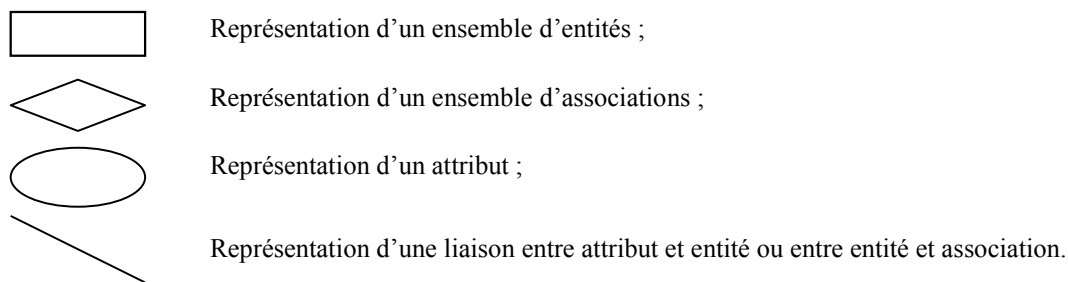


Figure 3 : Symboles utilisés pour la représentation standard d'un modèle entité/association

II. 2. Représentation Merise

En Merise :

- Une entité est représentée par un rectangle ;
- Une association est représentée par une ellipse ;
- Les entités et les associations sont reliées par des lignes pleines.
- Pour chaque entité l'identifiant est souligné ;
- Les cardinalités sont mises sur les extrémités des associations sous forme de couple.

Exemple :

Supposons qu'on ait des nageurs caractérisés par les informations suivantes : nom, prénom et qualité. Ils prennent des bains d'une certaine durée à une certaine date, sur des plages

caractérisées par leur nom, la région où ils se trouvent et la pollution de l'eau. Le modèle entité/association correspondant est le suivant :

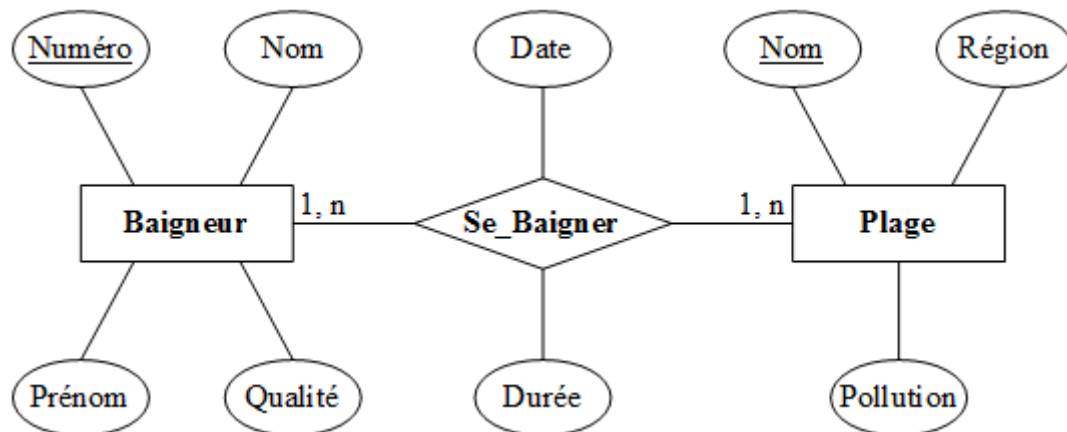


Figure 4 : Représentation standard du modèle entité/association

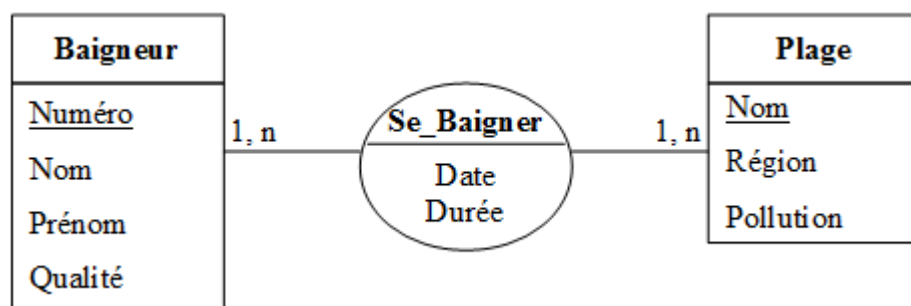


Figure 5 : Représentation Merise de modèle Entité/Association

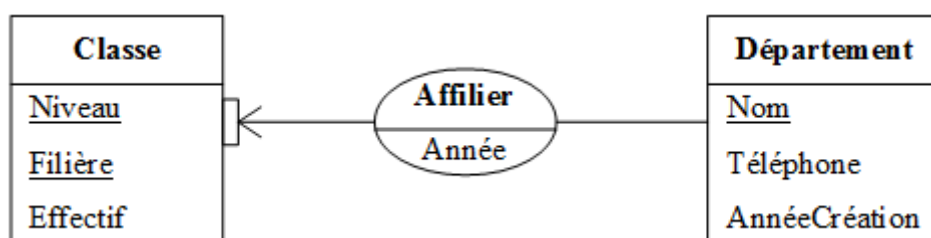


Figure 6 : Représentation d'une agrégation en Merise

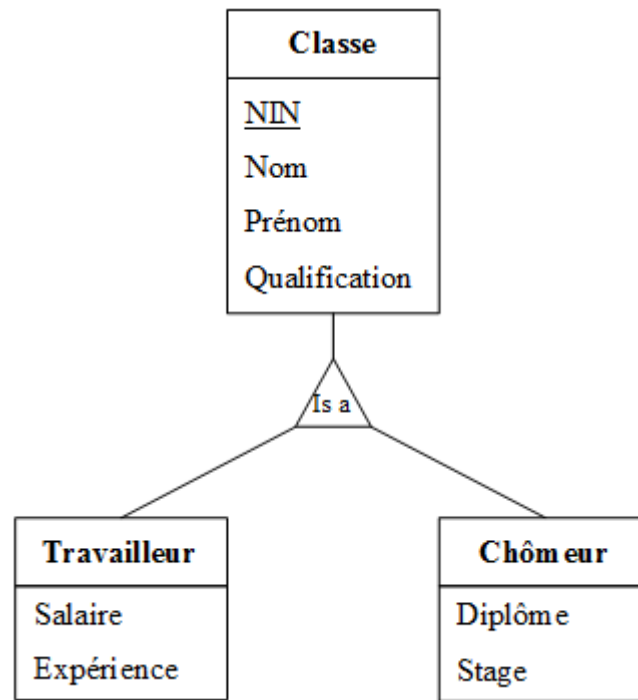


Figure 7 : Représentation de la Généralisation/Spécialisation en Merise

III. Les contraintes d'intégrité

On appelle contrainte d'intégrité toute règle implicite ou explicite que doivent respecter les données. Pour obtenir un bon modèle, il faut respecter toutes ces contraintes. Nous avons les contraintes du modèle entité/association et les contraintes dictées par les règles de gestion dans le cahier des charges. Parmi ces contraintes nous avons :

III. 1. Contrainte d'entité

- Chaque entité doit avoir un identifiant et un seul ;
- Deux entités différentes ne peuvent pas avoir le même nom ;
- Deux attributs d'une même entité ne peuvent pas avoir le même nom ;

III. 2. Contrainte de domaine

Chaque attribut doit avoir un domaine de valeurs. Les valeurs prises par les enregistrements pour cet attribut doivent appartenir à ce domaine.

III. 3. Contrainte d'unicité

- La valeur prise par un enregistrement au niveau de l'identifiant ne peut pas être prise par un autre ;

- Si un attribut est unique deux enregistrements ne peuvent pas avoir la même valeur pour cet attribut.

III. 4. Contraintes générales

Elles sont généralement décelées par le concepteur et permettent de rendre la base cohérente.

Exemple :

- On ne peut pas avoir une classe qui fait deux cours dans deux salles différentes en même temps ;
- Une classe de 200 étudiants ne peut pas faire cours dans une salle de 50 places.

IV. Comment modéliser ?

La méthodologie à suivre pour modéliser un système d'information donné est la suivante :

a. Identification des entités : L'identification des différentes entités est une étape très importante dans la conception d'un modèle entité/association. Durant cette phase, il faut respecter un certain nombre de règles parmi lesquels :

- Identifier toutes les entités ;
- Si une information décrit un objet, elle doit être considérée comme une entité ;
- Si un objet n'a qu'un identifiant, il doit être considéré comme un attribut ;
- Lister les attributs de chaque entité ;
- Les attributs pouvant avoir plusieurs valeurs doivent être transformés en entité ;
- Si un attribut a une association de cardinalité (1, n) avec une entité alors il doit devenir entité ;
- Un attribut doit être rattaché à l'entité qu'il décrit le plus directement ou à une association ;
- Choisir un identifiant pour chaque entité et éviter au maximum les identifiants composites.

a. 1. Identification des Généralisations/Spécialisations : Quand on conçoit une base de données, il peut arriver qu'une entité soit sous-ensemble d'une autre. Dans ce cas on parle de généralisation ou de spécialisation selon l'entité considérée.

Si nous avons dans une base les entités *Animal*, *Vache*, *Mouton* et *Chèvre*, on dira que l'entité *Animal* est une généralisation des entités *Vache*, *Mouton* et *Chèvre*. *Vache*, *Mouton* et *Chèvre* sont donc des spécialisations de l'entité *Animal*.

a. 2. Identification des agrégations : Il arrive souvent qu'un ou plusieurs attributs d'une ou de plusieurs autres entités soient nécessaires pour compléter l'identifiant d'une entité. Il faut repérer tous ces cas s'ils existent lors de la création du modèle entité/association.

b. Définition des associations : Il faut matérialiser les associations qui existent entre les différentes entités. Lors de cette étape, il faut :

- Identifier les associations possibles entre les entités ;
- Lister les attributs éventuels de chaque association ;
- Définir les cardinalités de chaque association ;
- Eliminer les associations redondantes ;
- Eviter les associations n_aire : s'il en existe, il faut essayer de les décomposer en plusieurs associations binaires si possible.

c. Documentation du modèle : Un modèle entité/association doit être validé par le client avant sa transformation en modèle relationnel. Le client étant généralement un non informaticien, il est nécessaire de bien documenter ce modèle pour lui permettre de pouvoir lire le modèle et vérifier la prise en compte ou non de ses exigences. Dans la documentation, la manière dont chaque règle de gestion notée dans le cahier des charges est prise en compte doit y figurer. Les règles ne pouvant être prises en compte à ce niveau doivent être signalées avec une explication sur quand et comment elles le seront.