Cours de Système d'Information

Chapitre 5 - Modèle Conceptuel des Données

Tarek Boutefara

Version 0.3, 2020-12-20

Table of Contents

1. Introduction

Le Modèle Conceptuel des Données est l'un des six modèles proposés par Merise. C'est le premier modèle des données et celui du plus haut niveau d'abstraction. Dans ce chapitre, nous allons voir ce modèle, sa construction, et sa normalisation.

2. Le MCD

Le Modèle Conceptuel des Données décrit de façon formelle les données utilisées par le Système d'Information. Il se base sur une représentation graphique des trois notions : entité, association, et propriété.

Le modèle représente l'aspect statique du système (les données). Il est important de noter que l'appellation "statique" fait référence aux données; le modèle lui-même évolue avec l'évolution de l'organisation et de ses données.

3. Les propriétés

Les propriétés sont les informations de base du Système d'Information. Une propriété est une information élémentaire (qui ne peut pas être décomposée) et monovaluée (possède une seule valeur élémentaire à un instant "t")

4. Les dépendances fonctionnelles

4.1. Définition

Une dépendance fonctionnelle c'est le fait de relier de manière unique une propriété à une autre propriété ou un ensemble de propriétés.

Une dépendance fonctionnelle entre A et B est notée :

$$A \rightarrow B$$

Et se lit "A détermine B", c'est-à-dire, pour une valeur définie de A, nous pouvons connaître d'une manière sûre (et unique) une valeur de B.

Exemple 01

Soit les propriétés suivantes :

- Code Etudiant,
- Nom Etudiant,
- Prénom Etudiant,
- Date de Naissance Etudiant,
- Code Module
- Intitulé du Module

- Coefficient Module
- Crédits Module

A partie de ces propriétés, nous pouvons définir les dépendances fonctionnelles suivantes :

- Premier sous-ensemble
  - Code Etudiant → Nom Etudiant
  - Code Etudiant → Prénom Etudiant
  - Code Etudiant → Date de Naissance Etudiant
- Deuxième sous-ensemble
  - Code Module → Intitulé Module
  - Code Module → Coefficient Module
  - Code Module → Crédits Module

En d'autres termes :

- Premier sous-ensemble :
  - Si on connaît le code de l'étudiant, nous pouvons connaître d'une manière sûre le nom, le prénom et la date de naissance de l'étudiant.
- Deuxième sous-ensemble :
  - Si on connaît le code du module, nous pouvons connaître d'une manière sûre l'intitulé du module, son coefficient et ses crédits.

Nous notons que nous ne pouvons définir aucune dépendance fonctionnelle entre les deux sous-ensembles.

Exemple 02

Reprenons l'ensemble des dépendances fonctionnelles de l'exemple précédent, on ajoute la dépendance suivante

- Code Module, Code Etudiant → Note

Autrement dit :

- Si on connaît le code du module et le code de l'étudiant, alors on peut connaître la note obtenue par l'étudiant dans ce module.

4.2. Graphe des Dépendances Fonctionnelles (*GDF*)

L'ensemble des dépendances fonctionnelles d'une application peut être représenté par une arborescence appelée graphe des dépendances fonctionnelles.

Exemple

Si on reprend l'exmeple 02 (extension de l'exemple 01) cité ci-dessus, le GDF sera comme suit :

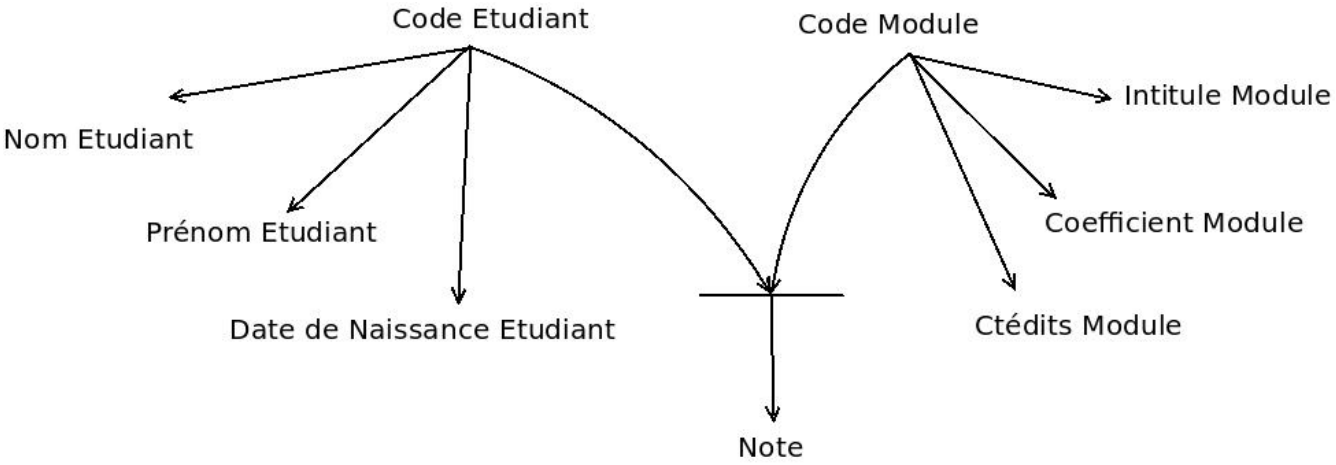


Figure 1. Graphe des Dépendances Fonctionnelles de l'exemple 02

4.3. Propriétés des Dépendances Fonctionnelles

- Réflexivité :
  - Si Y est un sous-ensemble de X alors X Y
- Augmentation :
  - Si X → Y alors XZ → YZ
- Transitivité :
  - Si X → Y et Y → Z alors X → Z
- Union :
  - Si X → Y et X → Z alors X → YZ
- Pseudo-transitivité :
  - Si X → Y et WY → Z alors WX → Z
- Décomposition :
  - Si X → Y et Z est un sous-ensemble de Y alors X → Z

4.4. La couverture minimale

La couverture minimale est l'ensemble des dépendances où chaque dépendance fonctionnelle ne peut pas être déduite à partir des autres dépendances fonctionnelles.

5. Les entités

5.1. Définition

Une entité permet de modéliser un ensemble d'objets de même nature (qui ont les mêmes propriétés).

Nous pouvons obtenir les entités par :

- Analyser les objets qui constituent le système ou bien qui sont manipulés par lui (approche descendante).

    - *Exemple* : à l'université, nous pouvons citer les "éléments", à titre d'exemple, les éléments Etudiant, Module, Enseignant, Classe, Groupe, Session. Tous ces éléments font partie du système ou bien ils sont manipulés par lui.
- Regrouper les propriétés qui dépendent de la même propriété ensemble (approche ascendante).
    - *Exemple* : dans le graphe des dépendances fonctionnelles cité ci-dessus, il est possible de
        - Regrouper "Code Etudiant, Nom Etudiant, Prénom Etudiant et Date de Naissance Etudiant" dans une entité "Etudiant".
        - Regrouper les propriétés "Code Module, Intitulé Module, Coefficient Module et Crédits Module" dans une entité "Module".

Il n'y a pas une règle précise pour déterminer toutes les entités. Le concepteur doit faire appel à ses capacités d'analyse.

5.2. Une occurrence

Une occurrence est une "instance" de l'entité, un objet parmi les objets représentés par l'entité.

Exemple

Dans le cas de l'entité Etudiant ci-dessus, nous pouvons donner comme exemples d'occurrences :

- Etudiant 01 : 18172256, Benahmed, Ahmed, 23/01/1999
- Etudiant 02 : 17179822, Benomar, Omar, 06/09/2000
- Etudiant 03 : 19183952, Benameur, Amer, 14/05/2001

5.3. Identifiant

L'identifiant est une propriété qui permet d'identifier une occurrence d'une manière unique et sûre.

Si l'approche suivie est l'approche descendante, l'identifiant est choisi parmi les propriétés de l'entité. Cette propriété (ou ensemble de propriétés) doit être :

- Non-nul (doit exister pour toutes les occurrences)
- Unique (chaque valeur est donnée une et une seule fois)
- Stable (ne doit pas changer dans le temps)

Si l'approche suivie est l'approche ascendante, il suffit de prendre les racines utilisées pour le regroupement des propriétés dans des entités.

5.4. Représentation graphique

Figure 2. Repérsentation graphique d'une entité

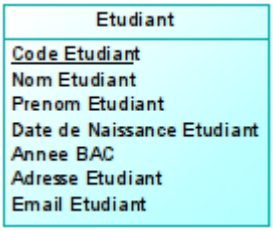


Figure 3. Exemple : entité "Etudiant"

6. Les associations

6.1. Définition

Une association décrit un lien entre deux ou plusieurs entités. Son existence est conditionnée par l'existance des entités qui y participent.

Nous pouvons obtenir (la majorité) des associations par :

- Analyser l'aspect dynamique (actions et tâches dans le système) ou bien les liens structurels (qui définissent une structure ou une forme d'organisation)
    - *Exemple* : à l'université, nous pouvons citer comme exemple de lien structurel : un Etudiant appartient à un Groupe. Nous pouvons citer comme exemple sur la dynamique au sein de l'organisation : un Enseignant enseigne un Module.
- La présence de propriétés qui nécessitent plusieurs identifiants (ou clés) d'autres entités pour les déterminer.
    - *Exemple* : dans l'exemple précédent, Note nécessaire Code Etudiant et Code Module pour pouvoir la déterminer.

Il n'y a pas une règle précise pour déterminer toutes les associations. Le concepteur doit faire appel à ses capacités d'analyse.

A son tour, une association peut porter des propriétés.

6.2. Représentation graphique

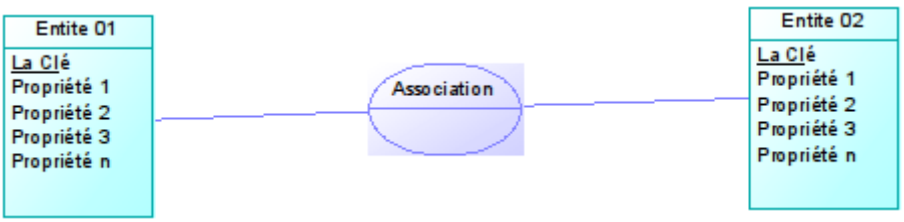


Figure 4. Repérsentation graphique d'une association

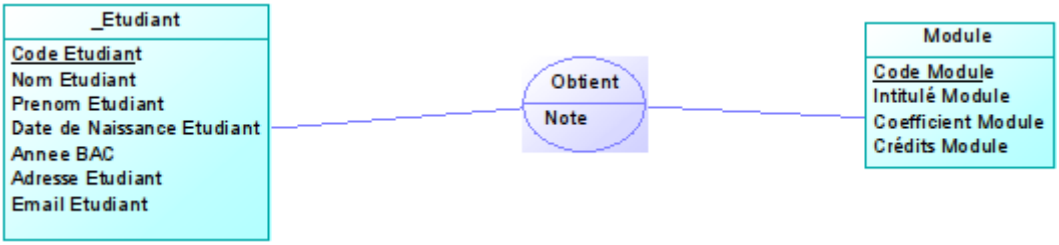


Figure 5. Exemple : association "Obtient"

6.3. Dimension

C'est le nombre de pattes d'une association (différent du nombre d'entités).

Exemple

L'association "Obtient" dans l'exemple ci-dessus est de dimension 2.

6.4. Cardinalité

Elle précise le nombre de participations de chaque occurrence de l'entité à l'association. Elle est notée sur la patte.

Exemple

Prenons les occurrences des Etudiants et des Modules suivants (diagramme d'occurrences)

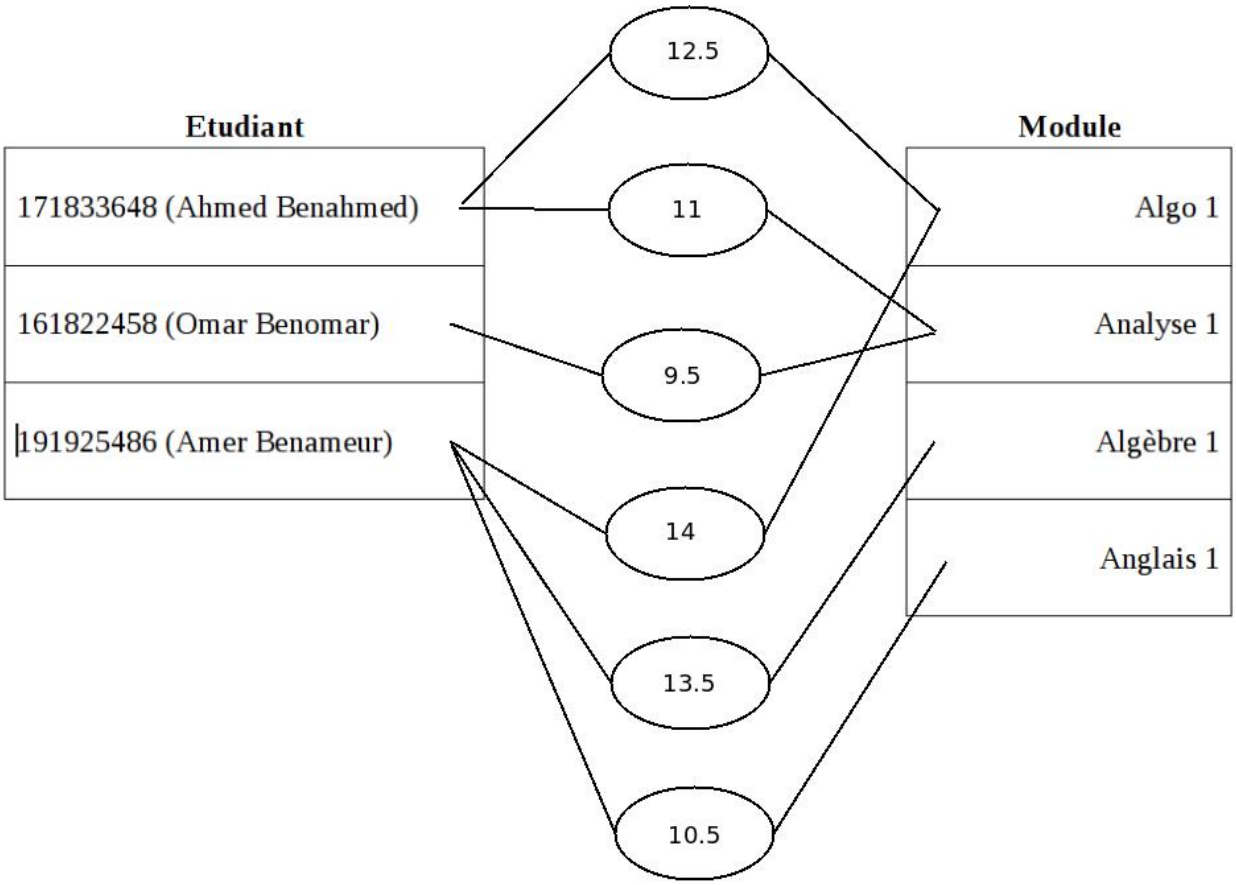


Figure 6. Diagramme d'occurrences

Dans cet exemle, nous pouvons voir que :

- **UN** étudiant peut entrer en relation avec **PLUSIEURS** modules,
- **UN** module peut entrer en relation avec **PLUSIEURS** étudiants.

Types de Cardinalités

Il existe quatre (04) types de cardinalites :

1. (0, 1) : au plus une fois.
2. (1, 1) : une et une seule fois.
3. (1, n) : au moins une fois.
4. (0, n) : aucune précision (plusieurs).

Exemple

Reprenons le dernier MCD (exemple Etudiant/Module) avec le diagramme d'occurrences associés. Le MCD final sera comme suit :

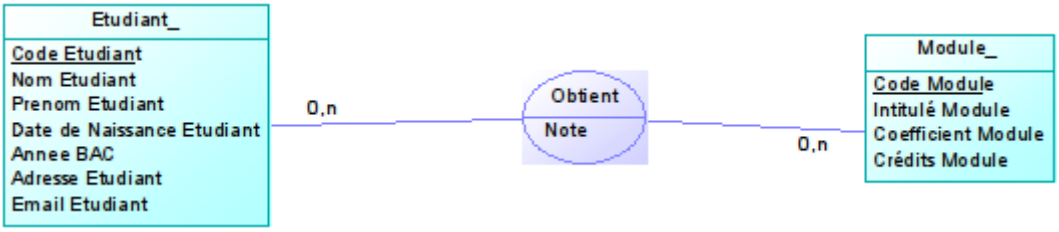


Figure 7. L'association "Obtient" avec les cardinalités

7. Normalisation du modèle

La normalisation du modèle permet d'enlever toute anomalie qui peut rendre le modèle incorrect (difficile, ou même impossible, à implémenter et à gérer). L'anomalie la plus courante est la redondance de l'information.

La redondance dans le modèle peut engendrer :

- Des erreurs lors de mise à jour : puisque l'information est répétée, il est possible qu'au moment de mise à jour, quelques copies soient mises à jour tandis que d'autres ne le seront pas (garde l'ancienne valeur).
- Lectures incohérentes : si des erreurs de mise à jour sont survenues, il est possible de lire des valeurs différentes (incohérentes) pour la même information.

7.1. La première forme normale (1FN)

Dans une entité, toutes les propriétés sont élémentaires et il existe au moins une clé.

Exemple

Soit le modèle suivant :

Si on considère qu'un livre peut avoir plusieurs auteurs, le modèle deviendra non normalisé (ne respecte pas la première forme normale). Pour le normaliser, la propriété "Auteur" doit être extraite comme une entité. Le modèle devient alors :

7.2. La deuxième forme normale (2FN)

- Le modèle doit être en 1FN
- Toutes les propriétés doivent dépendre de toute la clé et non pas d'une partie de la clé.

Exemple

Soit le modèle suivant :

Il est clair que la propriété "Nombre étages" dépend du "N° Bloc", c'est une propriété du bloc et est complètement indépendante de la notion d'appartement. Ainsi, ce modèle ne respecte pas la 2FN.

Pour le normaliser, la Dépendance Fonctionnelle qui cause problème doit être représentée par une entité séparée. Le modèle en 2FN devient alors :

Note : si la clé de l'entité est élémentaire (une seule propriété) alors l'entité respecte la 2FN automatiquement.

7.3. La troisième forme normale (3FN)

- Le modèle doit être en 2FN,
- Toutes les propriétés doivent dépendre de la clé d'une manière directe.

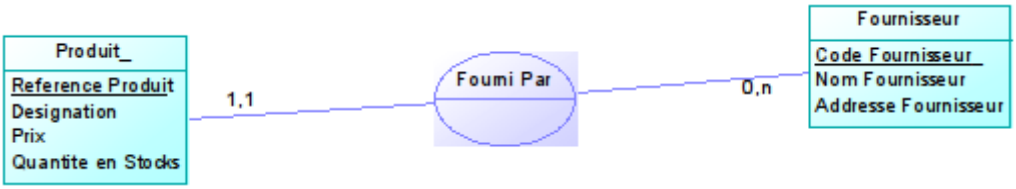
Exemple

Soit le modèle suivant :

Les propriétés Nom Fournisseur et Adresse Frounisseur dépendent de la clé Référence Produit, mais cette dépendance est indirecte. En effet, les dépendances sont clairement :

- Référence Produit → Code Frounisseur (si on connaît le produit, on peut identifier d'où il a été acheté),
- Code Fournisseur → Nom Fournisseur, Adresse Fournisseur (si on connaît le code du fournisseur, on peut connaître son nom et son adresse).

Pour le normaliser, les dépendances fonctionnelles qui provoquent l'erreur doivent être représentés par une nouvelle entité. Le modèle devient alors :



8. Conclusion

Dans ce chapitre, nous avons vu le premier modèle de données défini par Merise. Il s'agit du MCD (Modèle Conceptuel des Données). Nous pouvons constater qu'au cours d'élaboration de ce modèle, on se concentre exclusivement sur le "Quoi ?" sans se préoccuper de l'organisation de ces données ou comment seront-elles implémentées physiquement sur la machine.

Le chapitre présente aussi la notion de normalisation pour enlever toute anomalie. Le résultat sera un modèle dit "en 3ème forme normale". Cette forme normale est une condition nécessaire (et minimale) pour avoir un modèle correct et "implémentable".

Le chapitre suivant traite le Modèle Conceptuel des Traitements.