

## LA THEORIE DES DEPENDANCES ET LA NORMALISATION DU MCD

Une dépendance représente une contrainte exprimée par une règle que doivent vérifier les données pour que le modèle conceptuel de données soit *cohérent*.

Ex : Un étudiant est inscrit dans une seule filière

Normaliser une entité consiste à la représenter dans une forme *canonique* afin de s'assurer de l'*intégrité* des données. C'est un moyen fondamental mis à la disposition du concepteur pour éviter des erreurs de conception graves de conséquences.

**Cohérent** : les données entretiennent un rapport logique entre elles.

**Intégrité** : qui ne subit aucun dommage, aucune diminution ou aucune altération.

**Canonique** : qui respecte une règle

### I- DEPENDANCES FONCTIONNELLES (Formalisme relationnel de Codd)

#### I.1 Dépendance fonctionnelle (df) :

On dit que deux propriétés **a** et **b** sont reliées par une dépendance fonctionnelle :

$$\mathbf{a} \text{ — df —> } \mathbf{b}$$

si la connaissance de la valeur de **a** détermine une et une seule valeur de **b**.

Ex : Ninscr — df —> Nom

La connaissance du numéro d'inscription de l'étudiant (Ninscr) détermine une et une seule valeur de nom d'étudiant (Nom).

La dépendance fonctionnelle peut porter sur la concaténation (groupe) de plusieurs propriétés.

Ex : Ninscr + CodMod — df —> MoyMod

Le numéro d'inscription + le code du module détermine la moyenne modulaire de l'étudiant

#### I.2 Dépendance fonctionnelle élémentaire :

On dit qu'il y a dépendance fonctionnelle élémentaire entre les propriétés **a** et **b** et on note :

$$\mathbf{a} \text{ —> } \mathbf{b} \text{ si } \mathbf{a} \text{ — df —> } \mathbf{b} \text{ et si aucune partie de } \mathbf{a} \text{ ne détermine } \mathbf{b}.$$

Ex 1: Ninscr —> Adr est une df élémentaire (Adr est l'adresse de l'étudiant)

mais Ninscr + Nom — df —> Adr n'est pas élémentaire car

Ninscr — df —> Adr : Ninscr est une partie de Ninscr + Nom.

Ex2 : Ninscr + CodMod — df —> MoyMod est une df élémentaire : Il faut connaître le Ninscr et le CodMod pour avoir la moyenne modulaire (MoyMod).

#### I.3 Dépendance fonctionnelle élémentaire directe:

On dit que la propriété **b** dépend fonctionnellement de **a** par une dépendance fonctionnelle élémentaire directe si cette dépendance est élémentaire et s'il n'existe pas de propriété **c** telle que **a** — df —> **c** et **c** — df —> **b**. Autrement dit on élimine toute transitivité.

Ex : si entre les trois propriétés NInscr, CodFil et NomFil existent les df suivantes :

1) NInscr —> CodFil (CodFil : Code de la filière de l'étudiant)

2) CodFil —> NomFil (NomFil : Nom de la filière de l'étudiant)

3)  $NInscr \longrightarrow NomFil$

Les deux premières df sont directes mais la troisième est une df élémentaire seulement en raison de la transitivité :  $NInscr \longrightarrow CodFil \longrightarrow NomFil$

#### I.4 Propriétés des dépendances fonctionnelles :

- **Réflexivité :**

$a \longrightarrow df \longrightarrow a$

Ex:  $NInscr \longrightarrow df \longrightarrow NInscr$

- **Projection**

$a \longrightarrow df \longrightarrow b+c \implies a \longrightarrow df \longrightarrow b \text{ et } a \longrightarrow df \longrightarrow c$

$NInscr \longrightarrow df \longrightarrow Nom + Adr \implies NInscr \longrightarrow df \longrightarrow Nom \text{ et } NInscr \longrightarrow df \longrightarrow Adr$

- **Augmentation:**

$a \longrightarrow df \longrightarrow b \implies \forall c: a+c \longrightarrow df \longrightarrow b$

Ex :  $NInscr \longrightarrow df \longrightarrow Adr \implies NInscr + Nom \longrightarrow df \longrightarrow Adr$

- **Additivité:**

$a \longrightarrow df \longrightarrow b \text{ et } a \longrightarrow df \longrightarrow c \implies a \longrightarrow df \longrightarrow b+c$

Ex :  $\left. \begin{array}{l} NInscr \longrightarrow df \longrightarrow Nom \\ NInscr \longrightarrow df \longrightarrow Adr \end{array} \right\} \implies NInscr \longrightarrow df \longrightarrow Nom+ Adr$

- **Transitivité:**

$a \longrightarrow df \longrightarrow b \text{ et } b \longrightarrow df \longrightarrow c \implies a \longrightarrow df \longrightarrow c$

Ex :  $\left. \begin{array}{l} NInscr \longrightarrow df \longrightarrow CodFil \\ CodFil \longrightarrow df \longrightarrow NomFil \end{array} \right\} \implies NInscr \longrightarrow df \longrightarrow NomFil$

- **Pseudo-transitivité :**

$a \longrightarrow df \longrightarrow b \text{ et } b+c \longrightarrow df \longrightarrow d \implies a+c \longrightarrow df \longrightarrow d$

Ex :  $\left. \begin{array}{l} NInscr \longrightarrow df \longrightarrow CodFil \\ CodFil + Adr \longrightarrow df \longrightarrow NomFil \end{array} \right\} \implies NInscr+Adr \longrightarrow df \longrightarrow NomFil$

#### I.5 Clé d'une entité

Une clé (identifiant) d'une entité est une propriété (ou groupe (concaténation) de propriétés) de cette entité telle que toutes les autres propriétés de cette entité dépendent d'elle fonctionnellement. La dépendance n'est pas obligatoirement élémentaire.

La clé de l'entité est toujours soulignée. Aucune propriété de la clé ne doit contenir de valeur nulle. Toutes les propriétés de la clé doivent être renseignées.

Ex :

ETUDIANT
<u>NInscr</u> , Nom, Prenom, Adr, CodFil, Nomfil

La clé prend des valeurs distinctes qui permettent d'identifier chacune des occurrences de l'entité. La clé est choisie pour :

- Son caractère pratique
- Son importance
- Elle peut être inventée (crée) si aucune propriété de l'entité ne peut jouer ce rôle.

## **II-NORMALISATION DES ENTITES**

### **II.1 Entité normalisée**

Une entité est normalisée si une même propriété n'est pas représentée plusieurs fois dans l'entité et si les propriétés sont élémentaires, c'est à dire qu'aucune propriété n'est décomposable en d'autres propriétés.

**Remarque** : Une propriété dont on ne manipule pas les parties qui la composent est élémentaire. Par exemple l'adresse d'un client n'est pas élémentaire si on a besoin du code postal, du nom de la ville ou tout autre élément de cette adresse. Par contre, si à chaque fois l'adresse est considérée comme un seul élément alors on dira qu'elle est élémentaire. Le même cas pourra se poser avec la propriété date.

### **II.2 Nécessité de la normalisation**

Soit l'entité

ETUDIANT
<u>NInscr</u> , Nom, Prenom, Adr, CodFil, Nomfil

NInscr est la seule clé possible. L'entité est correcte mais va poser des problèmes de maintenance et provoquer des anomalies. L'entité n'est pas dans une « bonne forme » c'est à dire une bonne schématisation (conception) car elle est coûteuse en place :

- Pour chaque étudiant on doit saisir le code de la filière et le nom de la filière
- si une filière change de nom on doit changer le nom de cette filière dans toutes les occurrences des étudiants qui s'y sont inscrits.

### **II.3 Les trois premières formes normales**

L'objectif des trois premières formes normales est de permettre, à l'aide des dépendances fonctionnelles, la décomposition des entités. Avec cette décomposition on aboutira à un modèle conceptuel (schéma conceptuel) en « bonne forme ». Il représentera les entités et les associations essentielles en conformité avec les règles du monde réel.

#### **II.3.1 Première forme normale (1FN)**

Sachant qu'une entité a une clé, toute entité normalisée est en première forme normale (1FN). La 1FN revient à appliquer la définition de la df entre la clé et les autres propriétés.

Exemple d'entité en 1FN qui représente la filière et les moyennes modulaires de chaque étudiant :

ETUDIANT
<u>NInscr</u> , CodMod, Nom, Prenom, Adr, NomMod, CoefMod, MoyMod, CodFil, NomFil

Inconvénient :

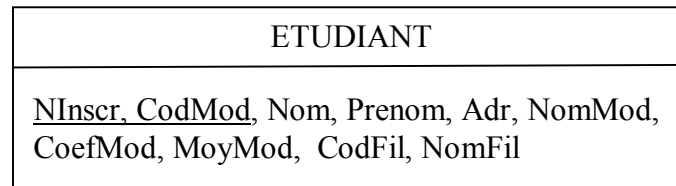
- Pour chacune des moyennes modulaires à saisir pour un étudiant, il faudra répéter la saisie de son nom et de son adresse ainsi que le code et le nom de sa filière.

**II.3.2 Deuxième forme normale (2FN)**

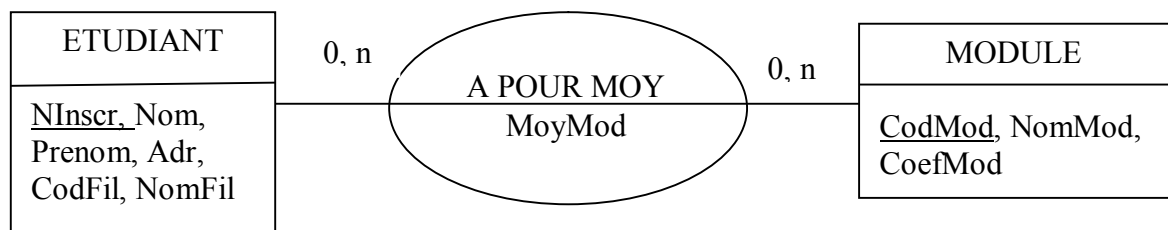
Une entité est en 2FN si :

- Elle est en 1FN.
- Toutes les DF issues de la clé sont élémentaires

En garantissant qu'aucune propriété non clé n'est déterminée par une partie de la clé, on évite certaines redondances. Soit l'entité ETUDIANT en 1FN :



Pour être en 2FN, on décomposera cette entité afin d'éviter les inconvénients constatés avec la 1FN. On obtiendra :



Les entités ETUDIANT et MODULE sont en 2FN.

On ne saisira avec ces entités qu'une seule fois le nom et l'adresse de l'étudiant de même qu'une seule fois le code, le nom et le coefficient des modules.

Inconvénient :

La df entre CodFil et NomFil n'est pas représentée, d'où la nécessité de saisir autant de fois le code et le nom de filière qu'il y a d'étudiants qui s'y sont inscrits.

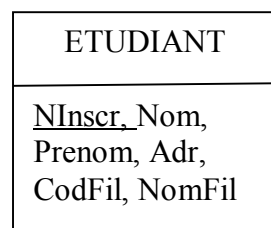
**II.3.3 Troisième forme normale (3FN)**

Une entité est en 3FN si :

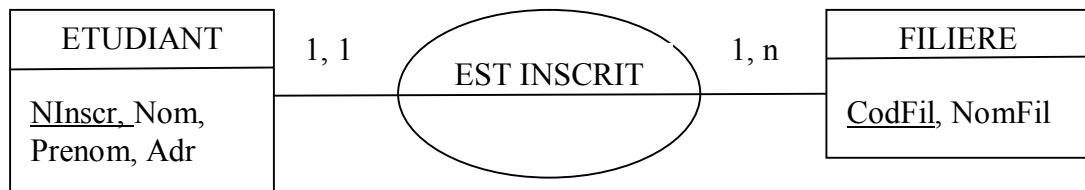
- Elle est en 2FN ;
- Toutes les DF issues de la clé sont directes.

La 3FN permet d'éviter les redondances dues aux transitivités.

Soit l'entité ETUDIANT en 2FN :



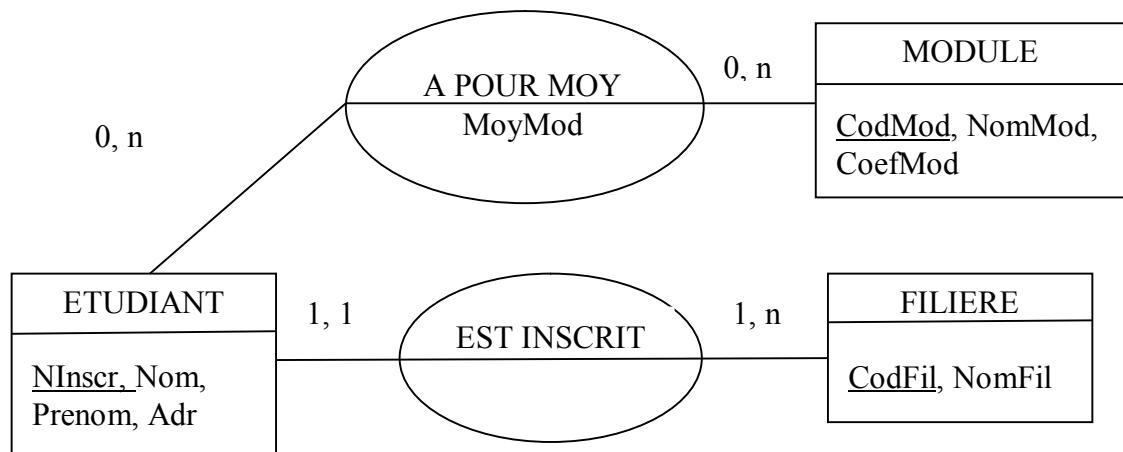
Pour être en 3FN, on décomposera cette entité afin d'éviter les inconvénients constatés avec la 2FN. On obtiendra :



On remarquera que lorsqu'une entité est en 3FN :

- aucune DF n'est issue d'un sous-ensemble de la clé
- aucune DF n'est issue d'une propriété non clé vers une autre propriété non clé.

En appliquant les règles de normalisation, à partir de l'entité ETUDIANT qui était en 1FN, on obtient le modèle conceptuel de données (MCD) en 3FN suivant :



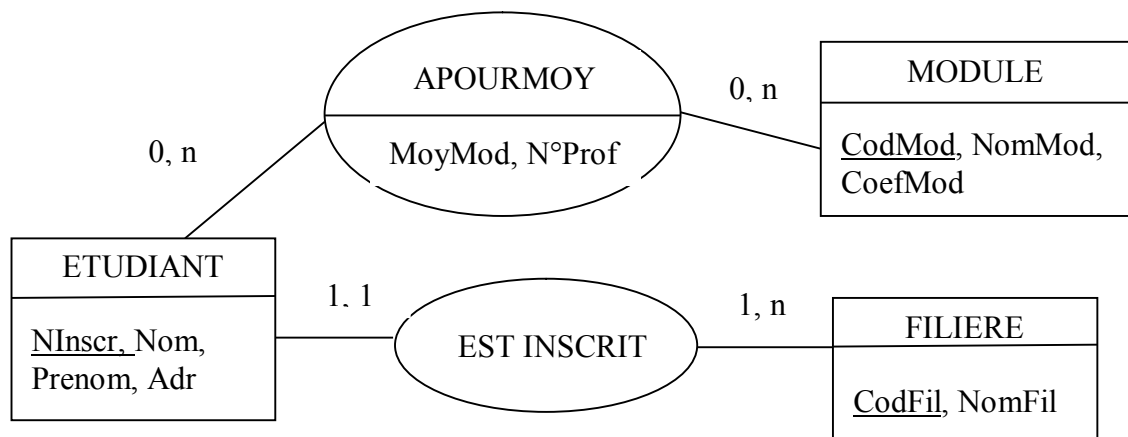
#### II.4 Forme normale de Boyce-Codd-Kent (3FNBCK)

Un MCD est en 3FNBCK si la source de chaque DF existant entre les propriétés de ce MCD est clé dans une entité.

$a \text{ --- } df \longrightarrow b$  : *a* partie gauche est la source de la *df* et *b* partie droite est la cible de la *df*.

Un MCD en 3FN peut présenter des anomalies en particulier si une propriété non clé est source de DF ayant pour cible (but) une partie (sous-ensemble) de la clé.

**Exemple** : Soit le MCD suivant :



Soit l'association APOURMOY qui représente la moyenne obtenue dans un module par un étudiant :

$APOURMOY(N^{\circ}Inscr, CodMod, N^{\circ}Prof, MoyMod)$  :  $N^{\circ}Inscr, CodMod$  est la clé

Supposons qu'on a la DF :  $N^{\circ}Prof \rightarrow CodMod$  : Un professeur enseigne un seul module  
Inconvénient :

Lors des saisies cette règle de gestion peut être violée au risque de voir par exemple différents modules être enseignés par le même professeur puisque dans ce MCD on n'a pas pu traduire qu'un enseignant enseigne un module uniquement.

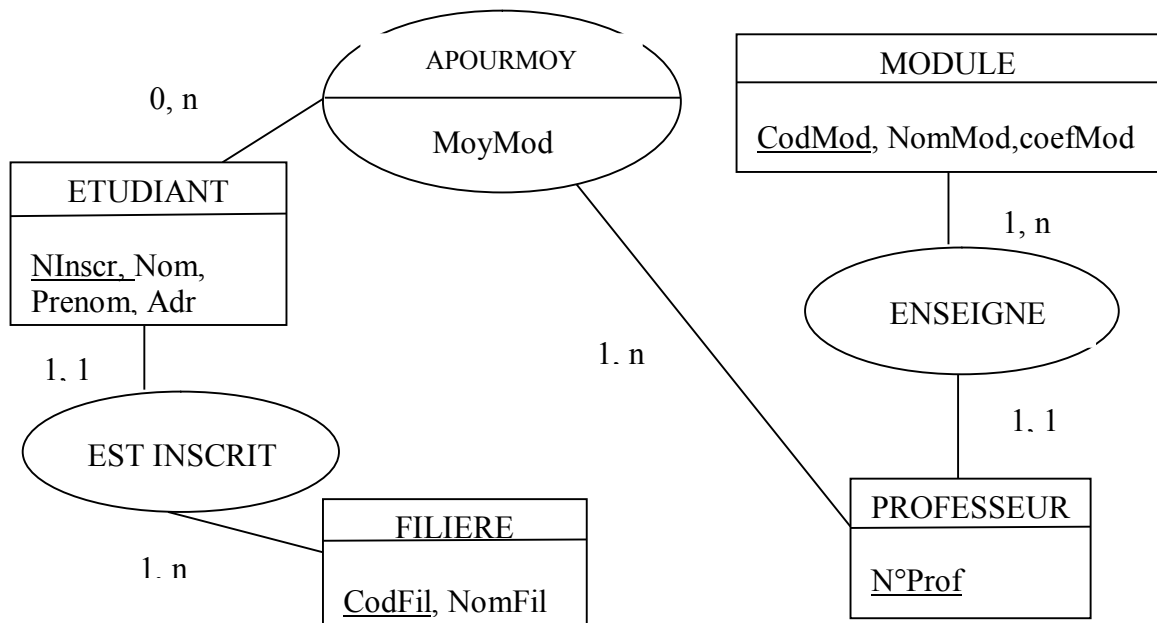
En traduisant les règles de gestion par des dépendances fonctionnelles on obtient :

$N^{\circ}Prof \rightarrow CodMod$

$NInscr, CodMod \rightarrow MoyMod$

En appliquant la propriété de la pseudo-transitivité on obtient :  $NInscr, N^{\circ}Prof \rightarrow MoyMod$

Le MCD en 3FNBCK sera :



### III. Normalisation des associations

Toute propriété d'une association doit dépendre de la clé de cette association par une dépendance fonctionnelle élémentaire directe.

En plus clair, toute propriété d'une association ne doit pas dépendre d'une partie de la clé seulement mais de toute la clé de cette association.