



Collège  
Lionel-Groulx

*Département d'informatique*

420 –KED-LG, conception de bases de données

# Normalisation en bref

## Table des matières

Bref historique.....	3
Concepts de base.....	3
1. Définition d'une base de données.....	3
2. Objectifs des SGBD. : .....	4
3. Types de SGBD .....	4
4. Modélisation des données.....	4
Modélisation des données : .....	5
Le Modèle Relationnel.....	6
Introduction : .....	6
Définitions.....	7
Vérification et normalisation du modèle relationnel.....	8
La vérification.....	8
La normalisation du modèle relationnel.....	8
Première forme normale : La clé (énoncé 1)--- il faut d'abord s'assurer que les tables respectent les règles de vérification .....	9
Première forme normale : La clé (énoncé 1).....	9
Deuxième forme normale Dépendance totale de la Clé (énoncé 2).....	10
Troisième forme normale : Et rien que la clé (aucune dépendance transitive.) (Énoncé 3).....	12
La forme normale de Boyce-Codd.(BCNF) (énoncé 4).....	14
La forme quatrième forme normale 4FN (énoncé 5) .....	15

## **Bref historique**

Les données représentent une ressource organisationnelle et informationnelle cruciale que l'entreprise se doit de maîtriser. La majorité des organisations ne saurait réussir sans connaître les données exactes de leur entreprise et de leur environnement externe. Pendant très longtemps, les systèmes d'information des entreprises structuraient leurs données sous forme de fichiers, chaque application utilisait donc son propre fichier. Beaucoup de problèmes étaient connus de cette façon de gérer les données d'une entreprise : la redondance d'information, la dépendance des données des traitements, et le manque d'intégration de données en font partie.

Devant le volume d'information grandissant et la complexité des traitements, le stockage des données dans de simples fichiers n'est plus la solution pour le stockage du patrimoine informationnel de l'entreprise. Un moyen de palier aux problèmes de la gestion des fichiers est le SGBD. (Systèmes de Gestion de Bases de Données).

Vers la fin des années 60 et le début des années 70, sont apparus les premiers SGBDs hiérarchiques et réseaux. Vers le milieu des années 70, nous avons vu naître les SGBDs relationnels, qui utilisent le modèle relationnel de Edgar Frank « Ted » Codd pour la modélisation des données. Aujourd'hui les SGBD relationnels sont présents dans la majorité des entreprises.

## **Concepts de base**

### **1. Définition d'une base de données**

Une base de données est un ensemble de données modélisant les objets d'une partie du monde réel et se servant de support à une application informatique.

Un SGBD (système de gestion de base de données) peut être perçu comme un ensemble de logiciels système permettant aux utilisateurs d'insérer, de modifier et de rechercher efficacement des données spécifiques dans une grande masse d'information partagée par de multiples utilisateurs.

## 2. Objectifs des SGBD. :

- Non redondance des données : permet de réduire le risque d'incohérence lors des mises à jour, de réduire les mises à jour et les saisies.
- Partage des données : ce qui permet de partager les données d'une base de données entre différentes applications et différents usagers.
- Cohérence des données : ce qui permet d'assurer que les règles auxquelles sont soumises les données sont contrôlées surtout lors de la modification des données.
- Sécurité des données : ce qui permet de contrôler les accès non autorisés ou mal intentionnés. Il existe des mécanismes adéquats pour autoriser, contrôler ou d'enlever des droits à n'importe quel usager à tout ensemble de données de la base de données.
- Indépendance physique : assurer l'indépendance des structures de stockage au structure des données du monde réel.
- Indépendance logique : possibilité de modifier un schéma externe sans modifier le conceptuel. Cette indépendance assure à chaque groupe les données comme il le souhaite à travers son schéma externe appelé encore une VUE.

## 3. Types de SGBD

Relationnel

Réseau

Hiérarchique

## 4. Modélisation des données

Un modèle de données est un ensemble de concepts utilisés pour décrire la structure d'une base de données. Par structure de base de données, nous entendons, les types de données, les relations, et les contraintes qui définissent le gabarit de la base de données. Tout modèle peut être exprimé à divers niveaux de précision, conceptuel, logique physique et externe. Cette définition sera retenue lorsque nous présenterons la partie « Présentation des approches traditionnelles pour l'intégration des données »

Les techniques de modélisation de données permettent d'une part de comprendre le fonctionnement d'un système et la façon dont les données sont organisées et d'autre part de construire un système d'information qui reflète la réalité.

Il existe plusieurs niveaux de représentation (ou de modélisation des données).

**4.1. Niveau conceptuel :** le niveau conceptuel ou le schéma conceptuel de la base de données décrit la réalité du système d'information d'une organisation indépendamment du SGBD d'implantation. Le modèle utilisé à ce niveau peut-être le modèle d'Entité/Association (modèle d'Entité/Relation). Ce niveau est de la responsabilité de l'analyste concepteur ou de l'administrateur de la base de données

**4.2. Niveau interne (logique et physique) :** le niveau interne spécifie comment les données sont représentées ou stockées sur les supports de stockage. Ce niveau est complètement pris en charge par les SGBD. (SGBDR Oracle)

**4.3. Niveau Externe :** ce niveau correspond à la vision de tout ou une partie du schéma conceptuel par un groupe d'utilisateurs concernés par une application ou une partie de l'application. Il s'agit de décrire à l'aide de schémas externes (vues) la façon dont

## **Modélisation des données :**

Les techniques de modélisation des données permettent d'une part de comprendre le fonctionnement du système et la façon dont les données sont organisées et d'autre part de construire un système informatisé qui reflète le monde réel.

Il existe plusieurs façons de représenter ou de modéliser les données, la plus utilisée est celle du modèle relationnel appelé encore **base de données relationnelle**.

# Le Modèle Relationnel

## Introduction :

Le modèle relationnel (mis de l'avant par Edgar Frank Codd) est le plus implémenté et le plus stable des modèles actuellement utilisés. Dans ce modèle la gestion des données est simplifiée grâce aux contraintes d'intégrité. Le système de récupération des données permet à l'utilisateur de visualiser la base de données à travers des structures de tables relationnelles.

L'unité de stockage élémentaire est la « table ». En général, c'est aux tables que les utilisateurs font référence pour accéder aux données. Une base de données est constituée de plusieurs tables reliées entre elles.

Une table est constituée de lignes et de colonnes

Exemple de table

Voici la table **Livres** qui contient des informations par rapport à des livres de notre bibliothèque.

NumeroLivre	TitreLivre	Auteur	Langue
101	Ainsi parlait Zarathoustra	Friedrich Nietzsche	Allemand
102	La paix des profondeurs	Aldous Huxley	Anglais
103	Les misérables	Victor Hugo	Français
104	La liberté n'est pas une marque de yogourt	Pierre Falardeau	Français

Remarquez

- 1- Chaque colonne définit un attribut. Un attribut est appelé également champ de la table. Les champs servent à donner la Définition d'une table.
- 2- Chaque ligne définit une instance (ou une occurrence) de la table Livres. Une ligne représente également un enregistrement d'une table.
- 3- Une table est donc un ensemble d'enregistrements. Pour définir un enregistrement on utilise des champs ou des attributs.

Dans une base de données Oracle, on utilise la commande CREATE pour créer une table.

Exemple

```
CREATE TABLE Livres (NumLivre NUMBER(5), TitreLivre CHAR (20), Auteur  
CHAR(20), Langue CHAR (10));
```

Pour insérer des données dans une table, on utilise la commande INSERT

Exemple

```
INSERT INTO Livres VALUES (101,'SQLPLUS', 'Oracle','Français');
```

Pour afficher le contenu d'une table on utilise la commande SELECT

Exemple

```
SELECT * FROM Livres, permet d'afficher Tous les enregistrement de la table Livres.
```

## Définitions

Termes	Définitions et représentation
Domaine	Ensemble de valeur caractérisé par un nom. Exemple le domaine de noms de jeux vidéo peut être Jeux {Word of Warcraft, Starcraft, Vampire, Civilization } Sexe {F, M}
Relation	Sous-ensemble du produit cartésien d'une liste de domaines caractérisé par un nom exemple
Schéma d'une relation	Nom de la relation suivi de la liste des attributs : Etudiant (Nom, Prenom, Code_permanent, adresse)
Attribut	Sous groupe d'information à l'intérieure d'une relation Code_permanent, Nom Un attribut peut être considéré comme une colonne d'une relation
Valeur d'un attribut	La valeur de l'attribut en rapport avec le domaine de valeurs. Exemple valeur de l'attribut Nom est Martin
Occurrence	Instance d'une entité ou d'une relation
Identifiant	Attribut permettant d'identifier les occurrences d'une entité ou d'une relation d'une manière unique

Table	Objet d'une base de données relationnelle. Ensemble de lignes et de colonnes. (ce concept sera défini en détail dans le chapitre suivant)
Clé primaire	Identifiant dans une table. Attribut permettant d'identifier chaque occurrence d'une table de manière unique. Ou encore attribut permettant d'identifier chaque ligne d'une table d'une manière unique.
Clé étrangère	Attribut d'une table (table A) qui est clé primaire d'une table (table B)

**Une base de données relationnelle est un ensemble de TABLES.** Une table est un ensemble de lignes. Chaque ligne constitue un enregistrement de la base de données

## Vérification et normalisation du modèle relationnel

### La vérification

On dit qu'un modèle conceptuel (entité/relation) est vérifié s'il répond aux conditions suivantes :

- Tous les attributs non calculés sont présents dans une entité ou table
- Aucun attribut n'est redondant
- Toutes les tables ont une clé primaire.
- Tous les attributs sont élémentaires (non décomposable ou se traitent comme un tout, exemple l'attribut adresse peut-être décomposable)

### La normalisation du modèle relationnel

Les règles de normalisation du modèle conceptuel permettent d'assurer :

- La non redondance des données
- L'intégrité des données
- La facilité de mise à jour

Lors de la normalisation d'un modèle de données (peu importe le niveau de modélisation, de conceptualisation ou de représentation) nous devons avoir au moins la 3eme forme normale.



## Définition :

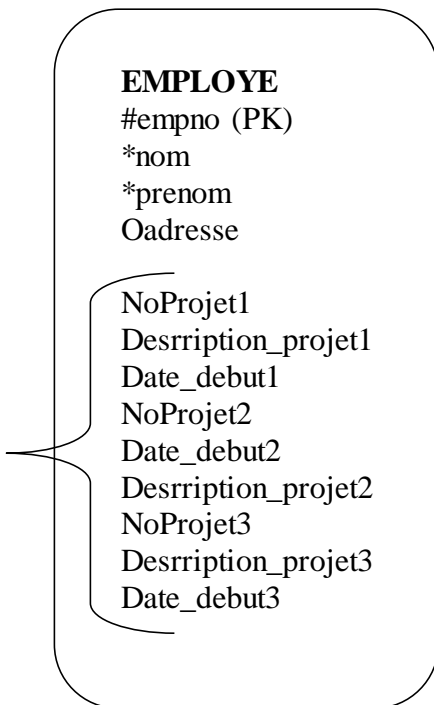
Une base de données est normalisée si celle-ci est au moins en 3eme forme normale.

**Première forme normale : La clé (énoncé 1)--- il faut d'abord s'assurer que les tables respectent les règles de vérification**

**Première forme normale : La clé (énoncé 1)**

Les tables ne doivent pas contenir des groupes de données répétitives. Si tel est le cas, il faut sortir le groupe de données et créer une autre entité qui va contenir ce groupe de données.

Dans les schémas qui suivent, PK désigne la Clé primaire. La clé primaire est précédée du symbole #



Problèmes avec cette représentation :

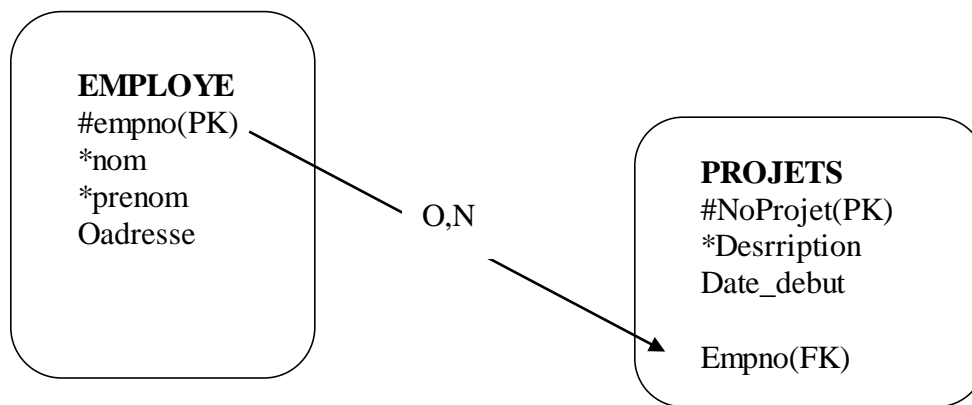
- ✓ On ne sait pas combien de projets sont affectés à chaque employé. Ce qui veut dire que lors de la création de la base de données, on ne sait pas combien de champs en rapport avec projet que nous allons créer (0?, 10?, 45? 1000?,...)
- ✓ Problème de redondance si la compagnie qui gère ces employés a affecté 200 employés à un projet alors on retrouvera 200 fois la même description du projet et 200 fois la date début du projet pour le projet en question.

Solution :

- 1- Créer une table Projet
- 2- Lier la table **EMPLOYE** avec la table **PROJETS**. Le type de lien sera déterminé par l'analyse du système.

Si un projet est affecté à un seul employé, alors la solution suggérée pour avoir la 1FN est la suivante :

Un employé peut avoir O ou N projets



Le lien entre les tables EMPLOYES et PROJETS est réalisé par le Empno qui se trouve dans la table PROJETS. Empno est dit **clé étrangère** (Foreign KEY)

## Deuxième forme normale **Dépendance totale de la Clé (énoncé 2)**

Une entité (Table) ou une relation est en deuxième forme normale 2FN, si elle est :

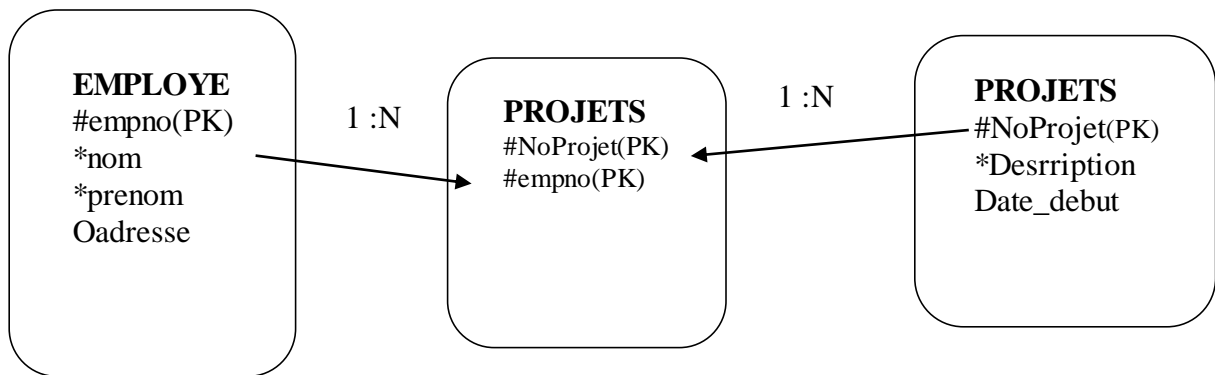
- a. En 1FN
- b. Tous les attributs de la relation ou de l'entité dépendent de **toute la clé** (notion de clé primaire composée) et non d'une partie de la clé.

La deuxième forme normale ne s'applique qu'aux tables ayant une clé primaire composée

Si on reprend l'exemple précédent, il est possible qu'un projet soit affecté à plusieurs employés. Dans ce cas , nous ne pouvons pas inclure TOUS les

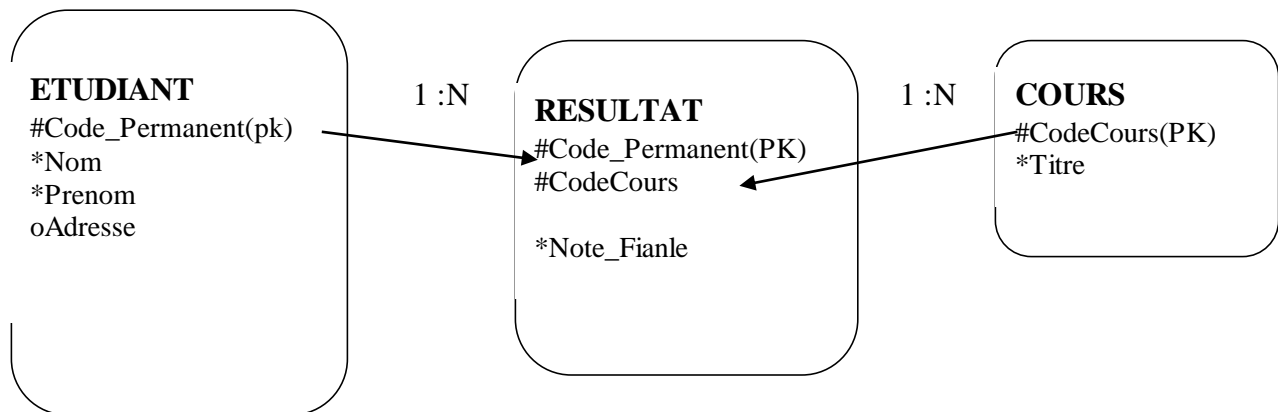
employés (Empno) dans la table PROJETS. (Problème de première forme normale).

La solution serait alors de créer une table de lien entre PROJETS ET EMPLOYES. Cette table de lien contiendra les attributs qui dépendent des deux clé Empno et NoProjet



(NoProjet,empno) constitue une clé primaire composée

### Autre exemple



Dans le modèle précédent, il est clair que la note finale d'un étudiant dépend de l'étudiant et du cours qu'il suit.

Voici un exemple de notes finales que l'on peut avoir.

Code_Permanent	CodeCours	Note_Finale
Poif9859-06	420-KED	85
Lef765-06	420-KED	73
Faf231-05	420-KED	77
Poif9859-06	420-KEG	72
Lef765-06	420-KEG	82
Poif9859-06	420-KEF	80

(Les codes permanents ne sont pas exacts)

Remarquez dans le tableau précédent que si l'on connaît le que code permanent Poif9859-06 on ne peut pas être renseigné sur sa note finale, il, faudrait connaître en plus du code permanent, le code cours. De la même façon, si je connais le code cours 420-KED, je ne peux pas être renseignée sur la note finale dans le cours correspondant. L'attribut Note\_Finale dépend ET du Code\_Permanent ET du code cours. La place de l'attribut.

Si nous devons placer l'attribut Note\_Finale dans la table Étudiant (Ou cours), cet attribut dépendra d'UNE PARTIE DE LA CLÉ. Nous ne sommes pas en 2FN.

### Troisième forme normale : **Et rien que la clé (aucune dépendance transitive.) (Énoncé 3)**

Une entité ou une relation est en troisième forme normale (3FN) si

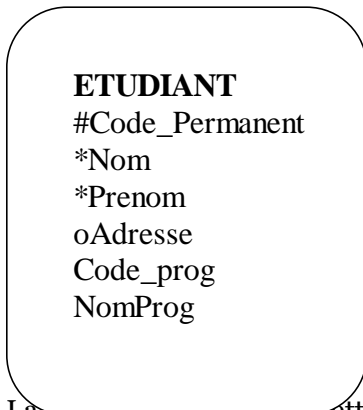
- Elle est déjà en 2FN
- Tous les attributs non clé dépendent uniquement de la clé. Il n'y a pas de dépendance entre deux attributs non clé.

La table ETUDIANT suivante, est en 2FN.

Elle est en 1FN, puis que pas de groupe répétitif, et tous les identifiants dépendent de la clé

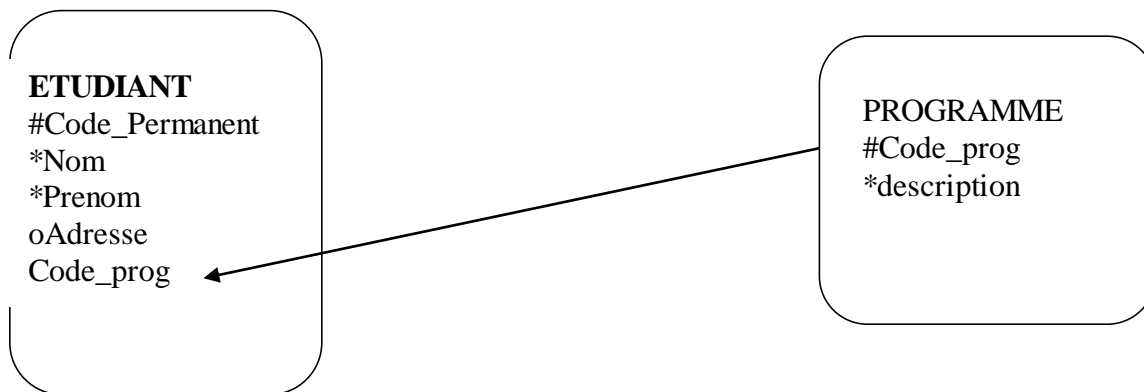
Elle est en 2FN, puis que tous les attributs dépendent de la totalité de la clé.

La même table n'est pas en 3FN Pourquoi?



La raison pour laquelle cette n'est pas en est que l'attribut NomProg dépend aussi de l'attribut Code\_prog. (le code 420 implique informatique.)

Pour avoir la table ETUDIANT en 3FN, il faut sortir les attributs Code\_prog et NomProg de la table, les mettre dans une table à part (PROGRAMME) et créer une relation entre ETUDIANT et PROGRAMME



L'attribut Code\_prog est une **clé étrangère** de la table Etudiant

## La forme normale de Boyce-Codd.(BCNF) (énoncé 4)

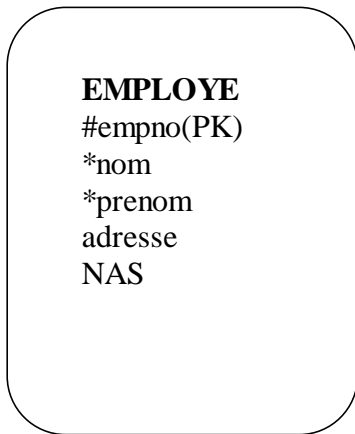
Une relation (table) est en BCNF si

- a. Elle est déjà en 3FN
- b. Les seules dépendances fonctionnelles sont celles dans lesquelles une clé (entière si composée) détermine un attribut.

Lorsque la clé primaire est composée, il arrive que certaine partie de la clé détermine un attribut non clé. La table peut être en 2FN, en 3FN mais pas en BCNF

Exemple1 :

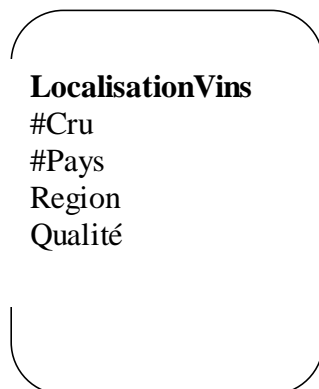
Voici une table Employe avec les attributs suivants :



Cette table, bien qu'elle soit en 3FN n'est pas en BCNF puisque le nom et prénom dépendent aussi du NAS.

Exemple2 :

Soit la table suivante : Cette table est bel et bien en 3FN



- Dans la table LocalisationVin, il y'a une clé primaire composée de (Cru et Pays). Le Cru et le pays détermine la qualité.
- Le Cru et le pays detrmine la région.
- MAIS, Région détermine Pays, donc il y a un attribut élémentaire qui détermine une partie de la clé

On peut dite que la table n'est pas en BCNF.

Pour qu'elle soit en BCNF, il faut

Avoir une table CRUS avec les 3 attributs : **cru** (clé primaire), **Pays**, **Qualité**  
Et une table REGIONS avec les attributs **région** (clé primaire), **Pays**

### La forme quatrième forme normale **4FN (énoncé 5)**

Une table (relation) est en quatrième forme normale si

- Elle est déjà en 3FN
- Les seules dépendances multivaluées élémentaires sont celles dans lesquelles une clé composée détermine un attribut.

La quatrième forme est une généralisation de la FNBC

Exemple

Soit la table suivante :

Dans cet exemple on considère qu'un cours est donné par un seul prof et qu'un prof peut donner plus d'un cours.

Un étudiant peut suivre plusieurs cours et un cours et suivi par plusieurs étudiants

**ETUDIANT-COURS-PROF**  
#Code\_Permanent  
#Code\_cours  
#numProf

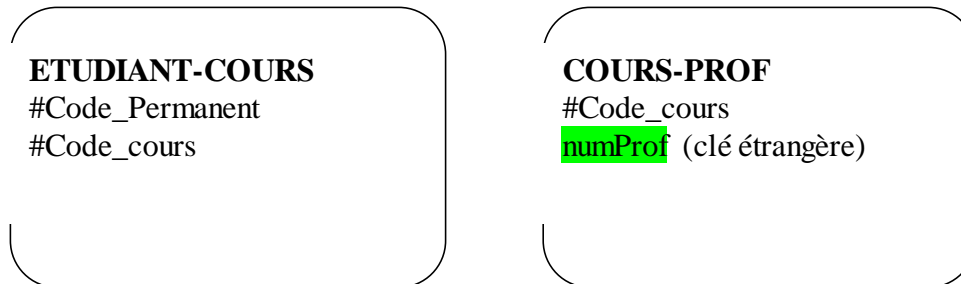
Ce qu'on veut représenter est la relation entre prof, cours et etudiants.  
Exemple d'occurrences

Code_Permanent	Code_cours	Num_Prof
100	Français 101	50
100	Français 102	51
100	Anglais101	53
101	Français102	50
100	Anglais102	55
101	Français 101	50
102	Français 101	50

La table Etudiant-Cours-Prof est en 3FN deux problèmes se posent :

- 1- contient beaucoup de redondances
- 2- le fait que le **code-cours** implique le **numProf** alors elle n'est pas en 4FN

Ce que nous pouvons faire est la chose suivante :



On peut retrouver facilement avec une jointure d'un prof dans un cours donné.  
(pas de perte d'information)

### **Attention !!!**

Si un cours est donné par plusieurs professeurs alors le numProf doit être une clé primaire dans la table **COURS-PROF**. **Dans ce cas nous avons une perte d'information : On ne peut pas retrouver les étudiants d'un prof dans un cours**

**Pour pouvoir retrouver les étudiants d'un prof dans un cours, il faudra revenir à la première table qui est :**

