

**« Conception d'une base de données relationnelle
appliquée à la biologie »**

Cyril GALLUT

UMR 7205

Institut de Systématique, Évolution, Biodiversité.

Sophie PASEK

UMR 7205

Institut de Systématique, Évolution, Biodiversité.

Bernard BILLOUD

UMR 8227

Laboratoire de Biologie Intégrative des Modèles Marins.

7 – 10 mars 2022

Plan du cours

1	Introduction	1
1.1	Qu'est-ce qu'une base de données?	1
1.2	Système de gestion de bases de données.	1
1.3	Différents niveaux d'une base de données.	1
1.3.1	Niveau physique	1
1.3.2	Niveau conceptuel.	2
1.3.3	Niveau externe.	2
1.4	Étapes de la réalisation d'une base de données.	2
1.5	Qu'est-ce qu'un modèle?	3
1.5.1	UML : Unified Modeling Language	3
1.5.2	Modèle « Entité-Association »	3
1.5.3	Modèle Relationnel	3
1.6	SQL : Structured Query Language.	3
1.7	Types de données.	4
1.8	Étapes de la modélisation	4
2	Modèle Entité-Association	5
2.1	Entité	5
2.1.1	Diagramme de classe	5
2.2	Association	6
2.2.1	Diagramme d'association	6
2.3	Modélisation	8
3	Modèle relationnel	9
3.1	Schéma de relation	9
3.2	Clefs	10
3.3	Association 1 à n	11
3.4	Association n à m	12
3.5	Schéma conceptuel	14
3.6	Passage du schéma Entité-Association au schéma conceptuel	14
3.7	Modélisation	14
4	Normalisation	15
4.1	1 ^{re} forme normale 1FN	15
4.1.1	Attributs composés	15
4.1.2	Valeurs Multiples	15
4.1.3	Attributs dupliqués	17
4.2	2 ^e forme normale 2FN	18
4.3	3 ^e forme normale 3FN	18
4.4	Domaines	19
4.5	Dénormalisation	20

*‘ I’d have you lot up in front of the University authorities
first thing in the morning, if it wasn’t for the fact
that you are the University authorities...’*
Terry PRATCHETT

1 Introduction

1.1 Qu'est-ce qu'une base de données ?

Une base de données est un ensemble de données, structurées et organisées de manière à pouvoir être facilement manipulées.

Une base de données permet de :

- modéliser et stocker de grandes quantités de données ;
- garantir leur cohérence ;
- les mettre à jour ;
- les retrouver facilement et rapidement, selon de nombreux critères ;
- les rendre accessibles simultanément à plusieurs utilisateurs.

1.2 Système de gestion de bases de données.

Un système de gestion de bases de données (SGBD) est un système informatique permettant la mise en œuvre et l'utilisation de bases de données :

- assure la persistance des données ;
- gère le support permanent ;
- indexe les données ce qui évite l'accès séquentiel ;
- garantit la sécurité des données ;
- autorise le partage des données ;
- protège l'accès aux données ;
- cache les détails de la représentation physique ;
- gère de nombreux utilisateurs simultanément.

1.3 Différents niveaux d'une base de données.

On distingue trois niveaux dans un SGBD : le niveau physique, le niveau conceptuel et le niveau externe. Il y a indépendance entre ces niveaux.

1.3.1 Niveau physique

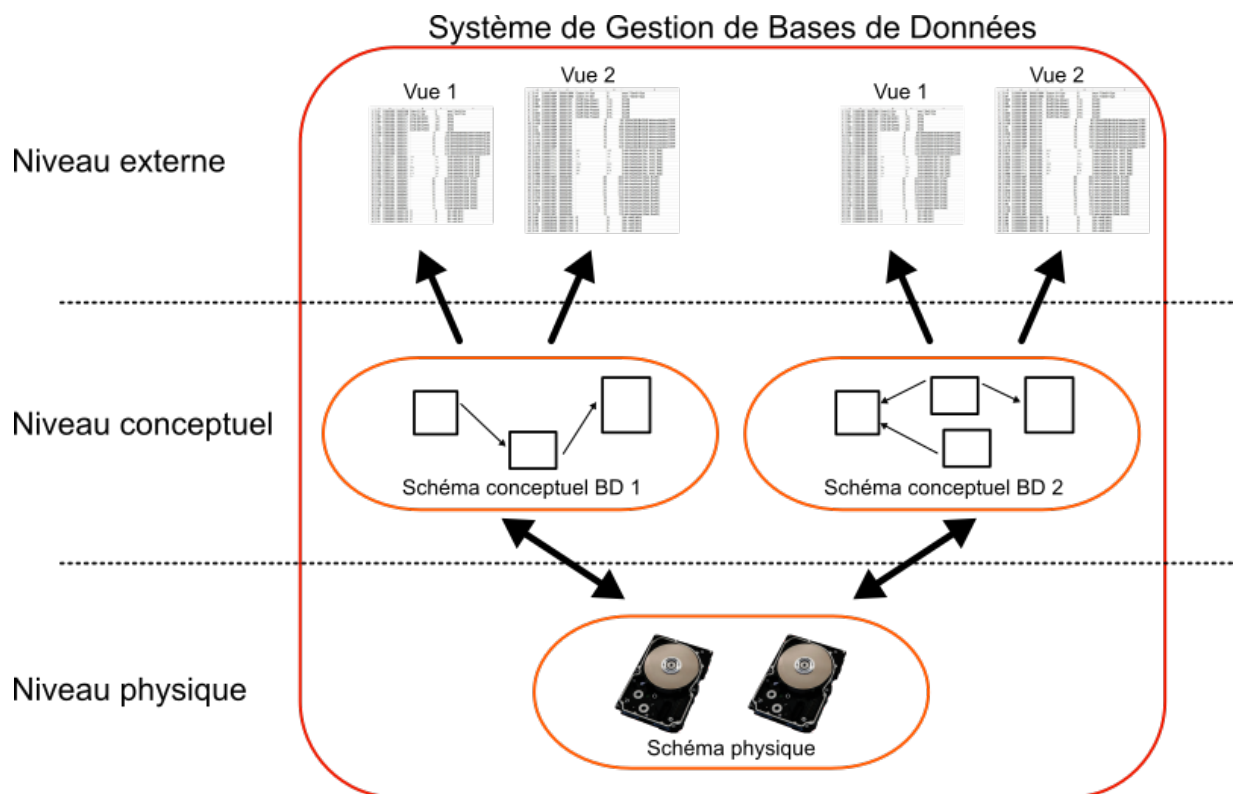
Le stockage physique des données est entièrement géré par le SGBD et le système d'exploitation. Ce niveau est invisible pour utilisateur, ainsi l'utilisateur n'a jamais à se préoccuper des noms de fichiers dans lesquels sont stockées les données, on peut les changer, l'organisation (le niveau conceptuel) de la base reste inchangée.

1.3.2 Niveau conceptuel.

Le schéma conceptuel (SC) d'une base de données est la représentation logique, à l'intérieur du SGBD, de la réalité que la base est censée transcrire. La création du SC revient à modéliser l'information. Il existe différents types de modèles : hiérarchique, en réseau, relationnel, orienté objet.

1.3.3 Niveau externe.

C'est le niveau d'interrogation de la base, on accède aux données par des vues. Une vue se présente comme un tableau et n'est généralement pas physiquement stockée mais elle est le plus souvent construite au moment de la requête.



1.4 Étapes de la réalisation d'une base de données.

1. Définition du cahier des charges. Quelles Données? Pour Quelles usage? pour quels utilisateurs?
2. Choix de l'infrastructure. Quel serveur? Quel hébergement?
3. Conception. Modélisation des données.
4. Déploiement informatique. Tests et améliorations.
5. Utilisation, maintenance et évolution.

1.5 Qu'est-ce qu'un modèle ?

Le terme de modèle revêt deux significations :

- un modèle est une représentation abstraite de la réalité, « une carte est une représentation d'un pays ». Un modèle représente une réalité mais il ne constitue pas cette réalité et il n'a pas à en avoir toutes les propriétés ;
- un modèle est un moyen de réaliser une représentation abstraite de la réalité, « la projection de Mercator est un modèle de représentation de la terre sur un plan ». Un modèle est un système formel qui définit des éléments permettant de représenter les objets de la réalité. Par exemple : modèle statistique, modèle empirique, modèle numérique, modèle relationnel.

1.5.1 UML : Unified Modeling Language

L'UML a été créé pour le génie logiciel orienté-objet. C'est un langage graphique où les éléments sont représentés sous forme de diagrammes. C'est un langage très vaste mais pas directement prévu pour la conception de bases de données relationnelles. Il manque certains éléments du modèle relationnel.

1.5.2 Modèle « Entité-Association »

Le modèle entité-association (EA) (Entity-Relationship en anglais) a été introduit par Peter CHEN en 1976. Ce modèle est fréquemment utilisé pour la modélisation de bases de données et il est particulièrement adapté à la conception de bases de données relationnelles. Il est constitué de trois notions principales : Entité, Attribut, Association. C'est un modèle de haut niveau qui permet de décrire des modèles de données conceptuels sous forme de diagrammes simples.

1.5.3 Modèle Relationnel

Il est basé sur l'algèbre relationnelle développée par Edgar CODD en 1970, qui permet de manipuler des relations en s'appuyant sur la théorie des ensembles. Le Modèle Relationnel permet de représenter les éléments qui sont nécessaires à la mise en œuvre d'une base de données. C'est le modèle formel de la plupart des SGBD, qui portent alors le nom de « Systèmes de Gestion Bases de Données Relationnelles », ou SGBDR.

1.6 SQL : Structured Query Language.

Le langage SQL est utilisé pour construire et manipuler les bases de données relationnelles. Il est basé sur l'algèbre relationnelle et fournit des possibilités supplémentaires. C'est un langage déclaratif, contrairement à la plupart des langages qui sont procéduraux. Il existe plusieurs standards qui sont plus ou moins disponibles dans les différents SGBDR.

1.7 Types de données.

Les données peuvent être de différentes natures :

- entiers : 0 10 -3 527 19 etc.;
- réels : 3.67 3/7 3.14 etc.;
- chaînes de caractères : 'toto' 'titi' 'Gros minet' etc.;
- dates : '2009-09-28' ;
- heures : '10h05' ;
- booléens : Vrai, Faux;
- etc.

En fonction du type de données on peut définir des opérateurs pour manipuler ces données.

- numériques : +, -, *, /, % etc.;
- booléens : ET, OU, NON;
- comparaisons : <, >, ==, != etc.;
- caractères : concaténation, comparaison, minuscule, majuscule...

1.8 Étapes de la modélisation

La modélisation consiste à représenter de manière abstraite une partie de la réalité au moyen d'un modèle. Il est nécessaire de formaliser les informations contenues dans les données avant de les représenter au moyen des outils fournis par le modèle. Cette représentation sera ensuite implémentée sous la forme d'une base de données. cela comporte plusieurs étapes :

1. Délimitation et caractérisation des « objets » présents dans les données,
2. Représentation sous forme de schéma Entité-Association,
3. Transcription en modèle relationnel → Schéma Conceptuel.

La première étape est cruciale pour obtenir une base de données fonctionnelle. Une bonne connaissance des données à modéliser est indispensable.

2 Modèle Entité-Association

2.1 Entité

Une « entité » est un objet du monde réel qui se distingue des autres et se caractérise par des attributs. C'est un objet physique, ou un fait tel qu'un événement ou une action. Par exemple : une personne ou un cours. Ainsi, une personne peut être caractérisée par son nom, son prénom, son sexe et son âge.

Un « attribut » se caractérise par son nom et son domaine, c'est-à-dire l'ensemble des valeurs qu'il peut prendre. Chaque attribut est donc une information, d'une nature (ou « type ») définie et qui contribue à caractériser l'entité. Par exemple, pour des données numériques, le domaine est la plage de valeurs entre le min et le max.

Un ensemble homogène d'entités de même nature, c'est-à-dire décrites par les mêmes attributs constituent une classe. Chaque entité de la classe est caractérisée par un ensemble de valeurs données aux différents attributs.

Par exemple, on peut définir une classe « **Personne** » par les attributs :

Nom	Prénom	Sexe	Date de naissance
-----	--------	------	-------------------

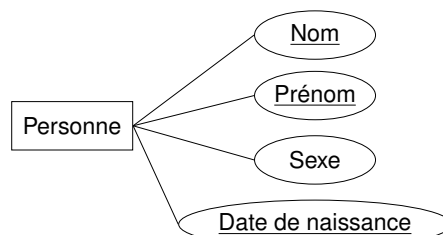
On peut alors créer deux instances de la classe **Personne** :

Reclus	Élisée	Masculin	15 03 1830
Sand	George	Féminin	01 07 1804

Un identificateur d'une classe d'entités est un ou plusieurs attributs qui permettent de distinguer une entité des autres entités de la classe.

2.1.1 Diagramme de classe

Une classe d'entités est représentée par un rectangle avec le nom de la classe. Les attributs de la classe sont représentés par des ovales rattachés à la classe par une arête et avec le nom au centre. Le ou les identificateurs sont soulignés.



2.2 Association

Une « association » est un lien fonctionnel entre deux entités. Une association peut être caractérisée par des attributs. Par exemple : un enseignant peut faire un cours. On peut décrire l'association en français :

Un enseignant « fait » un cours.

Un ensemble ou classe d'associations porte un nom et regroupe les associations de même type : mêmes attributs et reliant des entités de la même classe. Par simplification on parle d'association en lieu et place de classe d'association.

Une association est caractérisée par sa cardinalité. Ainsi, une association peut relier une ou plusieurs instances d'une classe à une ou plusieurs instances de l'autre classe. On définit ainsi trois grands types de cardinalités :

— 1 à 1 ;

Un lien de 1 à 1 représente en réalité deux caractéristiques du même objet, il n'y a donc pas lieu de faire deux classes.

— 1 à n ;

Un lien de 1 à n relie une instance d'une classe à plusieurs instances de l'autre classe.

— de n à m.

Un lien de n à m relie plusieurs éléments d'une classe à plusieurs éléments de l'autre classe.

Dans les deux derniers cas, il demeure possible qu'une instance d'une classe ne soit reliée à aucune instance de l'autre classe. Les cardinalités possibles sont donc les suivantes :

0..1 à 0..n et 0..n à 0..m.

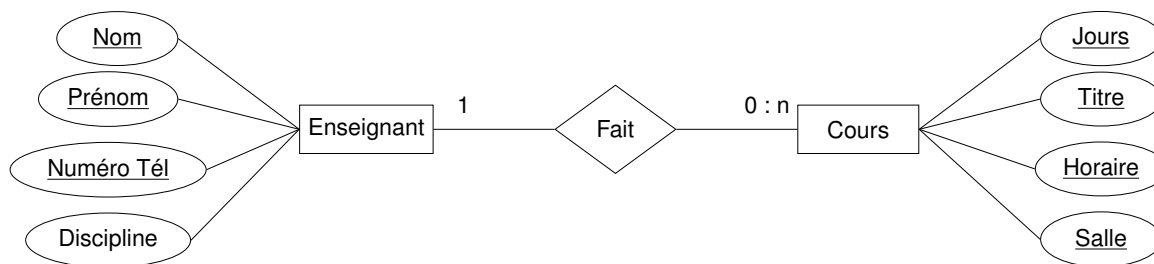
On précise donc les bornes si elles sont connues. Par exemple, un enseignant peut faire de 0 à une infinité de cours, un cours est forcément fait par un enseignant et un seul. C'est un lien de 1 à n ou plus précisément de 1 à 0..n.

2.2.1 Diagramme d'association

Une association est représentée par un losange avec le nom au centre. Elle est rattachée à chaque classe qu'elle relie par une arête. Les éventuels attributs qui la caractérisent sont représentés par des ovales rattachés au losange par une arête. Les cardinalités sont indiquées au dessus des arêtes reliant le losange de l'association au rectangle de l'entité.

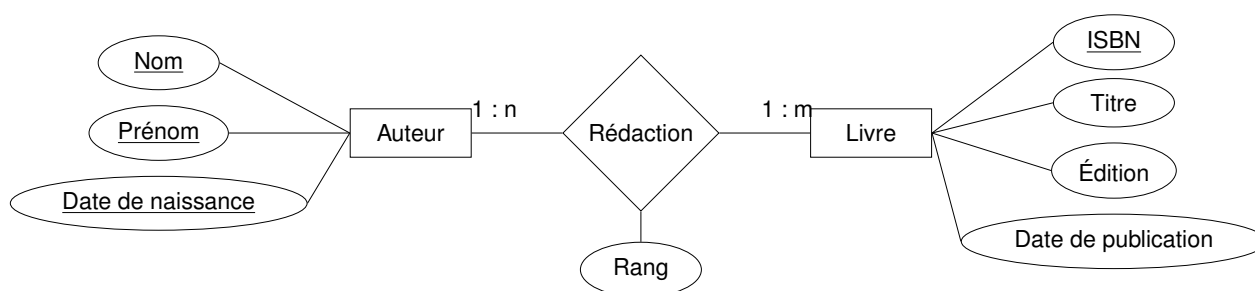
Association 1 à n

Un enseignant peut faire de 0 à une infinité de cours, un cours est forcément fait par un enseignant et un seul. C'est un lien de 1 à 0 : n :



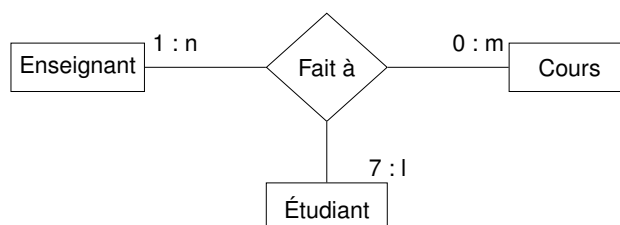
Association n-m

Un auteur écrit de un à une infinité de livres et un livre est écrit par un à une infinité d'auteurs. C'est une association de n à m ou de 1..n à 1..m. Chaque auteur d'un livre a un rang dans la liste des auteurs, il peut être le premier auteur, le deuxième auteur etc. L'association a donc un attribut « rang ».



Association ternaire

Une association peut relier plus de deux entités. Une association ternaire relie trois entités, par exemple une enseignante, une matière et des étudiants.



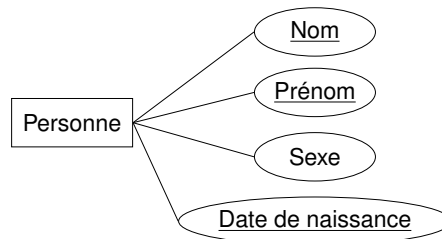
2.3 Modélisation

Afin de caractériser les informations contenues dans les données que l'on désire formaliser au moyen des outils fournis par le modèle EA il faut plusieurs étapes :

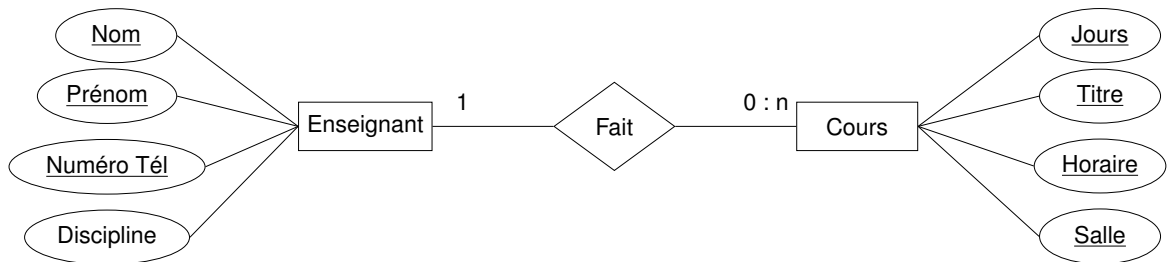
- délimiter les concepts présents dans les données, concepts physiques (objets) ou immatériels (actions, événements) ;
- identifier les attributs qui caractérisent ces concepts ;
- établir les types liens entre ces concepts.

Une fois établis ces concepts et les liens qui les relient il faut les représenter sous forme de classes et d'associations, trois règles s'appliquent en fonction du type de lien :

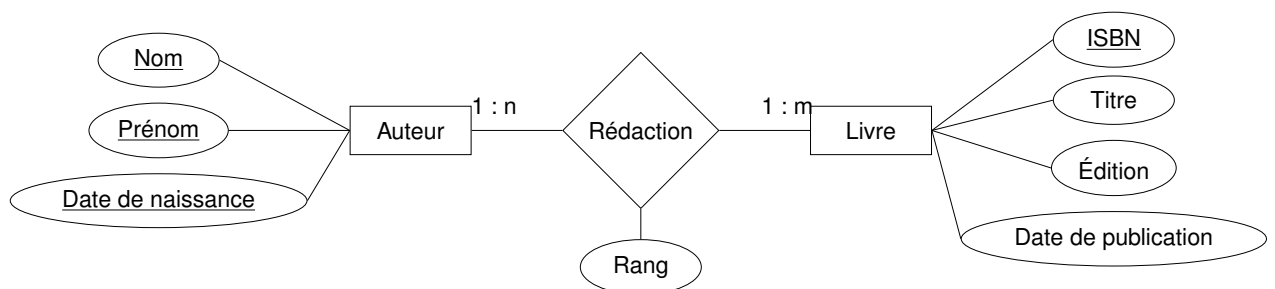
- 1 à 1 entre deux concepts → une seule classe avec deux attributs ;



- 1 à n entre deux concepts → deux classes et une association 1 à n ;



- n à m entre deux concepts → deux classes et une association n à m.



3 Modèle relationnel

Dans le Modèle Relationnel une classe est représentée par un schéma de Relation. C'est un ensemble d'attributs et de règles qui définissent les valeurs qui peuvent leur être assignées. Ces ensembles de valeurs correspondent aux domaines des attributs. Chaque instance du schéma de relation est un tuple. Un tuple est une fonction qui assigne une valeur du domaine à chaque attribut du schéma.

Cela implique un certain nombre de caractéristiques :

- un schéma de relation ne peut pas contenir d'éléments dupliqués,
- pour chaque attribut la valeur est unique (atomicité),
- l'ordre des attributs ne compte pas, de même que l'ordre des valeurs du tuple.

Une relation est un ensemble de tuples basés sur le même schéma. Dans une relation chaque tuple est unique et ils ne sont pas ordonnés.

On peut appliquer des opérateurs d'ensembles, tels que Union, Intersection, Différence, à plusieurs relations à partir du moment où elles sont basées sur le même schéma.

On peut joindre des relations basées sur des schémas différents, la jointure se fait sur l'association qui lie les relations. Le résultat d'une jointure est une nouvelle relation contenant un sous-ensemble du produit cartésien des relations correspondant aux associations valides.

On peut également appliquer des opérateurs unaires à une relation, tels que :

- projection ;
la projection construit une relation contenant un sous ensemble des attributs d'une relation.
- sélection.
la sélection construit une relation contenant un sous-ensemble des tuples d'une relation.

3.1 Schéma de relation

Représentation du schéma de relation :

Nom de la relation → **Personnes**

Nom des attributs →

Nom	Prenom	Sexe	DateDeNaissance
-----	--------	------	-----------------

Il n'existe pas de moyens graphique évident pour représenter le domaine de chaque attribut.

Ce schéma de relation se représente de façon standard sous la forme :

Personnes(Nom, Prenom, Sexe, DateDeNaissance).

Représentation de la relation :

Nom de la relation → **Personnes**

Nom des attributs →	Nom	Prenom	Sexe	DateDeNaissance
Tuple →	Sand	George	Féminin	01 07 1804
Tuple →	Sandea	Jules	Masculin	19 02 1811

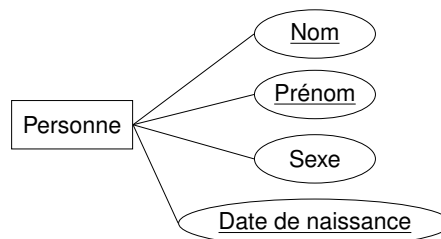
Valeur du domaine
assignée à l'attribut

Dans le langage SQL une relation est une table, les attributs sont les colonnes, les tuples sont les lignes de la table et les cellules sont les champs.

3.2 Clefs

Comme chaque tuple de la relation est unique, il ne peut y avoir deux lignes de la table qui aient exactement les mêmes valeurs pour tous les attributs. Pour s'assurer de cela il faut définir un sous-ensemble d'attributs dont les valeurs prises ensemble garantissent l'unicité. Tout sous-ensemble minimal d'attributs remplissant ce rôle forme une « clef ». Les clefs sont une propriété de la relation. Il peut exister plusieurs clef, il convient alors d'en choisir une qui servira de clef Primaire, on privilégiera l'ensemble minimal.

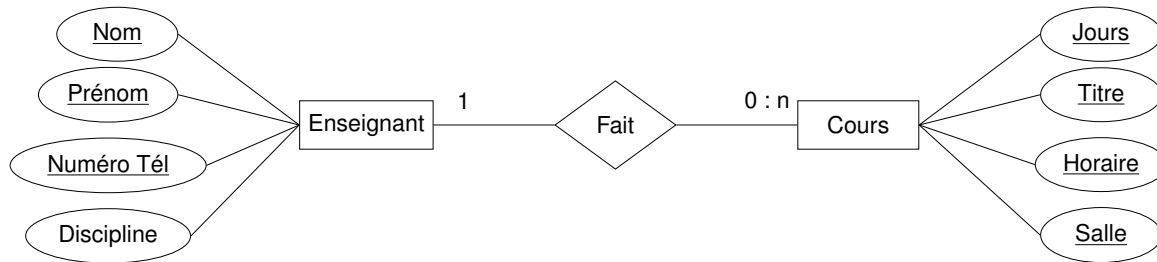
Pour la relation **Personnes**, il est possible que deux personnes aient le même nom et soient du même sexe mais il est peu probable qu'elles soient nées le même jour. On utilise comme clefs les attributs identificateurs de la classe **Personne** :



Personnes(Nom, Prénom, Sexe, DateDeNaissance).

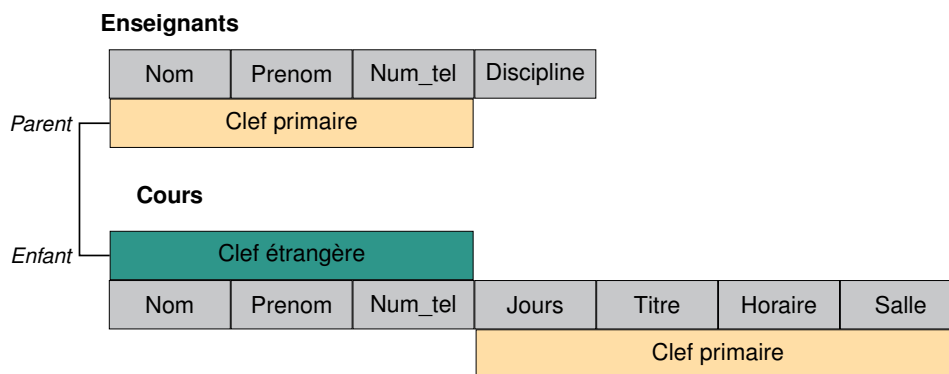
Personnes			
Nom	Prenom	DateDeNaissance	Sexe
Clef primaire			

3.3 Association 1 à n



Le schéma de relation **Cours** contient tous les attributs de la classe **Cours**.

Pour représenter l'Association, il faut représenter une information supplémentaire : qui fait le cours. Pour cela on ajoute au schéma de relation **Cours** les attributs de la clef primaire du schéma de relation **Enseignants**. Cela constitue alors pour le schéma de relation **Cours** une clef étrangère :

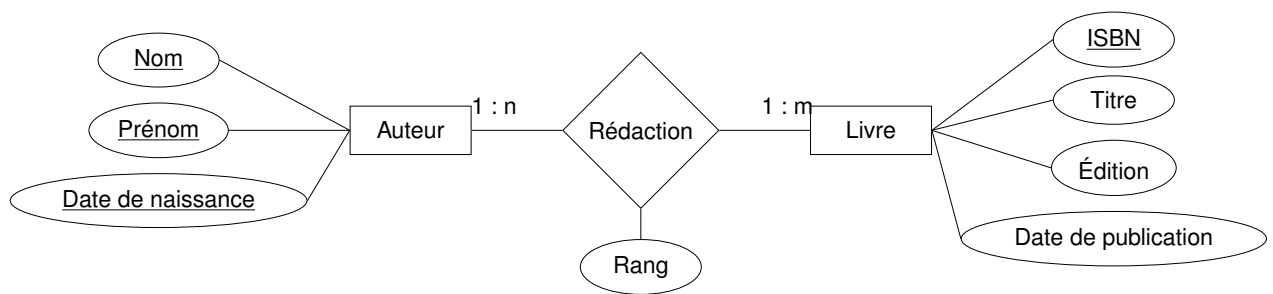


Étant donné qu'il ne peut y avoir de cours sans enseignant, on appelle le schéma **Enseignants** « parent » et le schéma **Cours** « enfant ». C'est toujours le côté 1 de l'association le parent et le côté n l'enfant. C'est donc toujours le schéma « enfant » qui a la clef étrangère. Les attributs de la clef étrangère doivent avoir exactement les mêmes noms et types que ceux de la clef primaire.

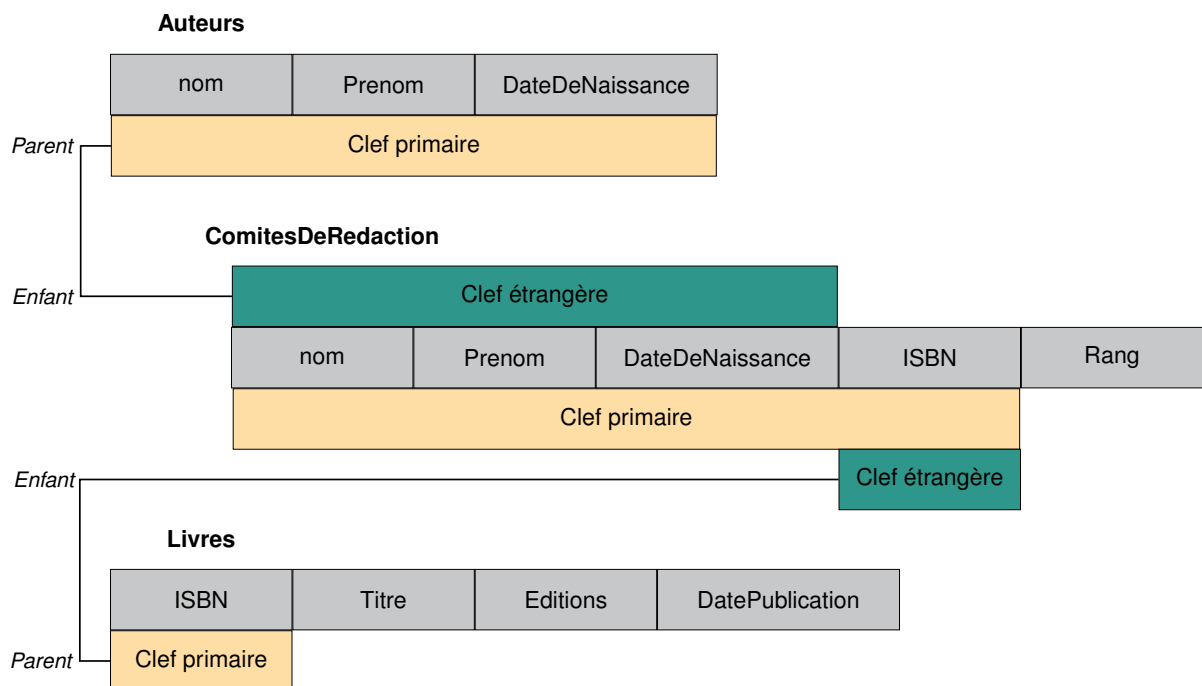
La clef étrangère permet d'assurer l'intégrité référentielle.

La taille de la clef primaire est importante à prendre en compte lorsqu'elle sert de clef étrangère pour une autre relation. Lorsqu'elle n'est pas limitée à un attribut cela duplique une grande quantité de donnée ce qui n'est pas optimal du point de vue du stockage. On remplace alors la clef par un nouvel attribut, soit artificiel (un index numérique) soit de substitution qui possède une valeur descriptive comme l'abréviation à deux lettres pour les pays : france → fr.

3.4 Association n à m



Le schéma de relation **Auteurs** ne peut contenir la clef étrangère du schéma de relation **Livres** et vice versa, en effet il est impossible d'avoir plusieurs valeurs par attribut (atomicité). On introduit par conséquent un schéma de relation intermédiaire qui correspond à la classe d'association **Rédaction** qui contient les clefs étrangères des deux autres schémas :



ISBN, La clef primaire de la relation **Livres**, permet d'identifier un ouvrage et remplit donc le rôle de clef dans le monde réel. Il convient de toujours choisir ce type d'attributs comme clefs primaires lorsqu'ils existent.

Exemple de relations basées sur ces schémas :

Livres

ISBN	Titre	Editions	DatePublication
2742759808	Histoire d'un ruisseau	Hetzel	1869
2858821860	Le Commissionnaire	B. Renault	1831
274276187X	Histoire d'une montagne	Hetzel	1880

Auteurs

Nom	Prenom	DateDeNaissance
Reclus	Élysée	15 03 1830
Sand	George	01 07 1804
Sandeau	Jules	19 02 1811

Redaction

Nom	Prenom	DateDeNaissance	ISBN	rang
Reclus	Élysée	15 03 1830	2742759808	1
Sand	George	01 07 1804	2858821860	1
Sandeau	Jules	19 02 1811	2858821860	2
Reclus	Élysée	15 03 1830	274276187X	1

Jointure de ces trois tables :

Nom	Prenom	DateDe Nais- sance	ISBN	rang	Titre	Editions	Date Publi- cation
Reclus	Élysée	15 03 1830	27427 59808	1	Histoire d'un ruisseau	Hetzel	1869
Sand	George	01 07 1804	28588 21860	1	Le Commissionnaire	B. Renault	1831
Sandeau	Jules	19 02 1811	28588 21860	2	Le Commissionnaire	B. Renault	1831
Reclus	Élysée	15 03 1830	27427 6187X	1	Histoire d'une montagne	Hetzel	1880

3.5 Schéma conceptuel

Le schéma conceptuel d'une base de données est constitué de l'ensemble des schémas de relation qui permettent de représenter toutes les informations contenues dans les données. Le schéma conceptuel peut être mis en œuvre directement sous forme de base de données relationnelle.

3.6 Passage du schéma Entité-Association au schéma conceptuel

Pour passer du schéma Entité-Association au schéma conceptuel il faut suivre les étapes suivantes :

1. Pour chaque classe définir le schéma de relation correspondant, la clef de ce schéma est l'identificateur de la classe,
2. pour chaque association dont la cardinalité est 1 à n, ajouter la clef primaire de la relation côté 1 au schéma de relation côté n comme clef étrangère,
3. pour les autres cas d'association créer un schéma de relation qui comprend les clefs primaires des relations représentant les classes d'entités associées et s'il y en a les attributs de l'association. La clef primaire de cette relation est formée par l'ensemble des clefs primaires des relations représentant les classes d'entités associées

3.7 Modélisation

Plusieurs règles informelles doivent s'appliquer à la conception d'une base de données :

- Éviter les redondances dans les données, elles font perdre de la place et sont des causes d'incohérences potentielles.
- Nommer les tables et les attributs de façon explicite, pour faciliter la maintenance et l'utilisation.
- Les attributs doivent exister dans le monde réel.
- Homogénéiser la nature des valeurs (même unité).

4 Normalisation

La normalisation permet d'éviter une mauvaise modélisation qui entraînerait des problèmes potentiels tels que la redondance des informations, des incohérences et des difficultés de maintenance.

Pour normaliser un schéma conceptuel on applique successivement différentes formes normales (FN) qui sont des règles que doivent respecter les relations.

1 ^{re} forme normale	— tous les attributs possèdent une valeur atomique.	— diviser les attributs composés ; — éliminer les valeurs multiples ; — éliminer les attributs dupliqués.
2 ^e forme normale	— un attribut non clef ne doit pas dépendre d'une partie de la clef.	— <i>décomposer la relation pour déplacer les attributs dépendants dans une autre relation.</i>
3 ^e forme normale	— un attribut non clef ne doit pas dépendre d'un autre attribut non clef.	— <i>décomposer la relation pour déplacer les attributs dépendants dans une autre relation.</i>

4.1 1^{re} forme normale 1FN

4.1.1 Attributs composés

Les attributs composés contiennent plusieurs valeurs qui correspondent à des attributs différents.

Auteurs		Auteurs		
Personne	DateDeNaissance	Nom	Prenom	DateDeNaissance
Reclus Élysée	15 03 1830	Reclus	Élysée	15 03 1830
Sand George	01 07 1804	Sand	George	01 07 1804
Sandeau Jules	19 02 1811	Sandeau	Jules	19 02 1811

Par exemple ici l'attribut Personne contient le nom et le prénom, il convient dans ce cas de diviser les attributs composés en autant d'attributs que nécessaire afin que les valeurs soient atomiques.

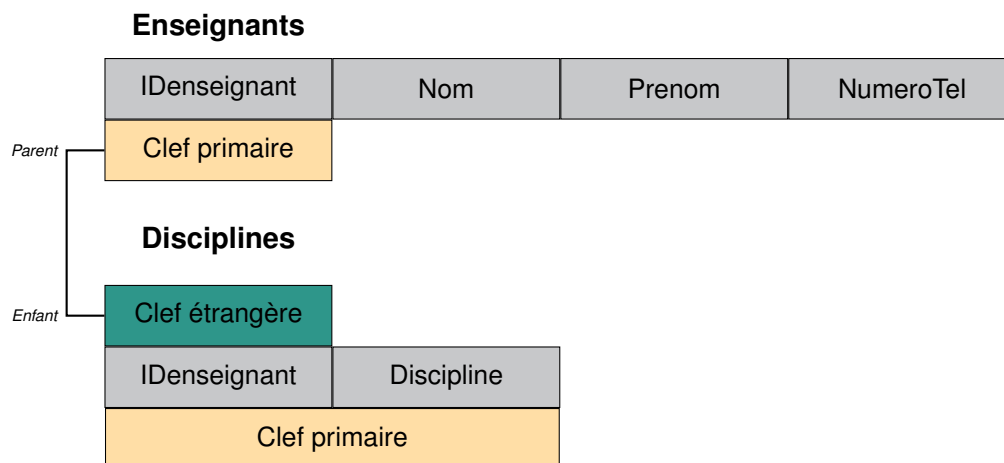
4.1.2 Valeurs Multiples

Les valeurs multiples dans un attribut (ici « Disciplines ») ne permettent pas une recherche optimale. Il devient difficile de faire une recherche sur une seule valeur et il est impossible de lister chaque valeur individuellement.

Enseignants

IDenseignant	Nom	Prenom	NumeroTel	Disciplines
1	Gallut	Cyril	01 40 79 40 79	Phylogénie, Bases de données.
2	Billoud	Bernard	01 44 27 32 10	Bases de données, analyse de séquences.
3	Pasek	Sophie	01 44 27 81 47	Analyse de séquences, Génomique.

La première forme de normalisation implique que la valeur d'un attribut doit être atomique. Pour palier ce problème on remplace l'attribut contenant des valeurs multiples par un nouveau schéma de relation avec l'attribut correspondant. On a ainsi deux schémas de relation :



Enseignants				Disciplines	
IDenseignant	Nom	Prenom	NumeroTel	IDenseignant	Discipline
1	Gallut	Cyril	01 40 79 40 79	1	Phylogénie
2	Billoud	Bernard	01 44 27 32 10	1	Bases de données
3	Pasek	Sophie	01 44 27 81 47	2	Bases de données
				2	Analyse de séquences
				3	Analyse de séquences
				3	Génomique

4.1.3 Attributs dupliqués

Si certains attributs d'une même relation portent sur des valeurs de même nature la relation doit être décomposée pour réunir ces attributs dans un seul attribut d'une autre relation.

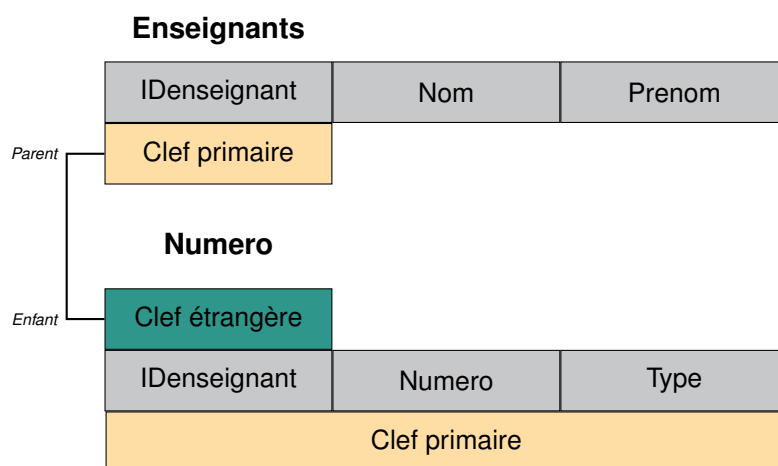
Enseignants

Nom	Prenom	NumeroTel	Fax	Portable
Gallut	Cyril	01 40 79 40 79	01 88 85 86 87	
Billoud	Bernard	01 44 27 32 10		06 98 76 54 32
Pasek	Sophie	01 44 27 81 47	01 55 55 00 01	

Aucun des enseignants n'a une valeur pour chacun des attributs. Les attributs non renseignés prennent la valeur NULL. Il est préférable d'éviter au maximum les valeurs NULL.

De plus, il est toujours possible que de nouveaux types de numéros de téléphone soient créés. Avec la relation ci-dessus, il faudrait alors ajouter des attributs. Cette opération est toujours compliquée lorsque la base est déjà en fonction.

Pour palier à ces problèmes, on remplace les attributs dupliqués par un nouveau schéma de relation.



Enseignants

IDenseignant	Nom	Prenom
1	Gallut	Cyril
2	Billoud	Bernard
3	Pasek	Sophie

Telephones

IDenseignant	Numero	Type
1	01 40 79 40 79	Tel
1	01 88 85 86 87	Fax
2	01 44 27 32 10	Tel
2	06 98 76 54 32	Portable
3	01 44 27 81 47	Tel
3	01 55 55 00 01	Fax

4.2 2^e forme normale 2FN

Il y a dépendance fonctionnelle lorsque un ou plusieurs attributs implique d'autres attributs. C'est le cas de la clef primaire qui implique les autres attributs du schéma de relation mais cela peut être le cas pour d'autres attributs.

Un schéma de relation respecte la 2^e forme normale si il est conforme à la 1^{re} forme normale et si tout attribut n'appartenant pas à la clé primaire ne dépend pas d'une partie de la clef primaire mais de sa totalité.

Remarque : une relation dont la clef primaire ne comporte qu'un seul attribut est de 2^e forme normale.

Cours

NumSalle	NumProf	Matiere	Nom	Prenom	Horaire
RC03	1	Bioinfo	Bouhane	Ahmed	13h
RC03	1	BD	Bouhane	Ahmed	8h
RC05	2	Phylogénie	Laurentin	Sylvie	14h
RC07	3	Genetique	Santos	Maria	9h30

4.3 3^e forme normale 3FN

Pour la 3^e forme normale une relation doit remplir les conditions de la 2^e forme normale et ne doit comporter aucun attribut non clef qui dépende d'un autre attribut non clef.

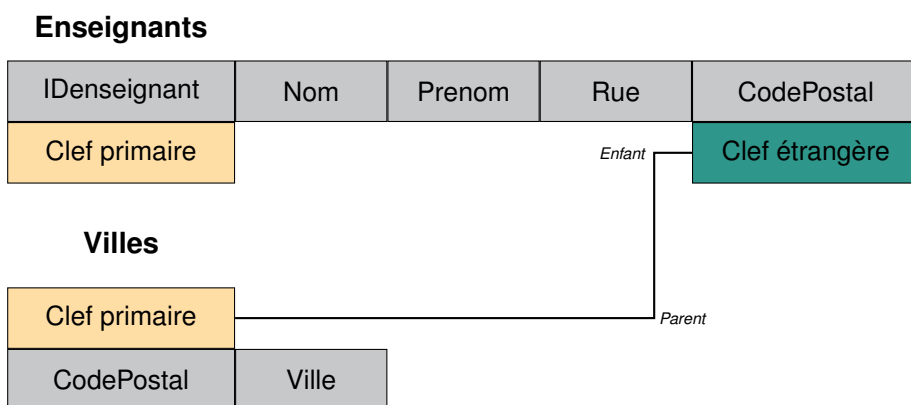
Enseignants

IDenseignant	Nom	Prenom	Rue	CodePostal	Ville
1	Gallut	Cyril	Toulliguin	29680	Roscoff
2	Billoud	Bernard	An Illiz	29680	Roscof
3	Pasek	Sophie	Charles	45000	Orléans

Dans cet exemple le code postal implique la valeur de ville. C'est une clef potentielle. Cette relation ne respecte pas les conditions de la 3^e forme normale car code postal et ville ne font pas partie de la clef (IDenseignant).

Pour résoudre ce type de problème il faut décomposer la relation de telle manière que pour chaque clefs potentielles :

- on crée un nouveau schéma de relation avec la clef potentielle comme clef primaire ainsi que tous les attributs fonctionnellement dépendants ;
- on supprime les attributs fonctionnellement dépendants du schéma de départ ;
- on garde la clef potentielle comme clef étrangère dans le schéma de départ.

**Enseignants**

IDenseignant	Nom	Prenom	Rue	CodePostal
1	Gallut	Cyril	Toulliguin	29680
2	Billoud	Bernard	An Illiz	29680
3	Pasek	Sophie	Charles	45000

Villes

CodePostal	Ville
29680	Roscoff
45000	Orléans

On constate que la redondance a été éliminée grâce à cette normalisation. Ce qui permet d'assurer l'intégrité des données (voir erreur de frappe dans la première table) et limite la taille de la base.

4.4 Domaines

Il est important de préciser les domaines de tous les attributs à la création.

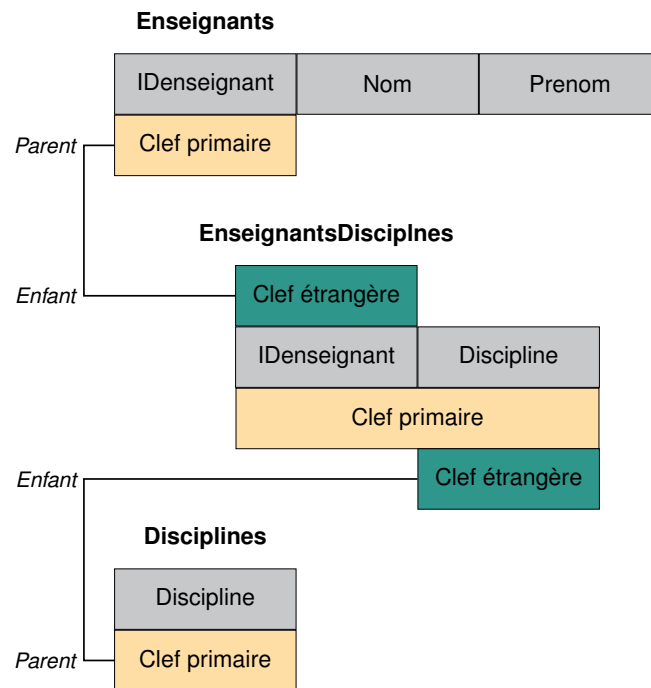
Certains sont difficiles à décrire précisément : Nom de personne.

Certains possèdent un patron : adresse mail par exemple.

Certains ont une forme précise et immuable : numéro ISBN ou numéro de sécurité sociale.

Enfin certains peuvent être une énumération connue (au moins en partie) à la création. Dans ces cas, il est bon de faire un schéma de relation indépendant.

Par exemple : les différentes disciplines enseignées.

**Enseignants**

IDenseignant	Nom	Prenom
1	Gallut	Cyril
2	Billoud	Bernard
3	Pasek	Sophie

EnseignantsDisciplines

IDenseignant	IDdiscipline
1	1
1	2
2	2
2	3
3	3
3	4

Disciplines

IDdiscipline	Discipline
1	Phylogénie
2	Bases de données
3	Analyse de séquences
4	Génomique

4.5 Dénormalisation

Il peut arriver que pour des raisons de performances il soit nécessaire de fusionner certaines tables, ce qui va dans le sens contraire de la normalisation. Il faut garder en

tête qu'il faut maintenir l'intégrité des données à un autre niveau, par exemple lors de la saisie des données.