

CONCEPTION

Analyse des besoins

La première étape peut passer par le dispositif papier/crayon pour analyser les besoins du demandeur.

En fonction de ses besoins il faudra déterminer le dictionnaire de données.

Attention ne pas confondre données et traitements qui seront à voir plus tard.

Chaque métier a ses spécificités, bien faire préciser le fonctionnement par plusieurs intervenants au besoin.

Conseil personnel : suivre physiquement un process (exemple parcourir toute la chaîne de production, se mettre à la place d'une caissière, d'un produit sur un tapis, d'un serveur de restaurant avec son terminal de commandes, du cuisinier qui la reçoit ...etc...)

Le dictionnaire de données :

La 1ère analyse aboutit à la création du dictionnaire de données

une donnée a une signification , une désignation et un type.

D'autres caractéristiques moins importantes peuvent préciser la nature des données (contrainte de valeur, minimum, maximum, intervalle, valeur calculée...).

Attribut (ou propriété)

- nom d'une personne
- prenom
- rue
- code postal
- ville
- no de sécurité sociale

Entité (ou classe)

Une entité (classe) est une description d'un ensemble homogène d'objets (concrets ou abstraits), chacun caractérisé par différents attributs.

Ensemble d'attributs regroupés par un identifiant. (exemple no de sécu)

classe Personne composée des attributs

- id
- numero de sécu
- nom

prénom

Vous devrez affecter, à chacune de vos classes, un identifiant. Un identifiant est une propriété naturelle ou artificielle d'une classe permettant de distinguer un objet de tous les autres (et ce quel que soit l'objet considéré). Un identifiant peut être composé d'un ou de plusieurs attributs.

Qui dit libellé dit identifiant.

exemple

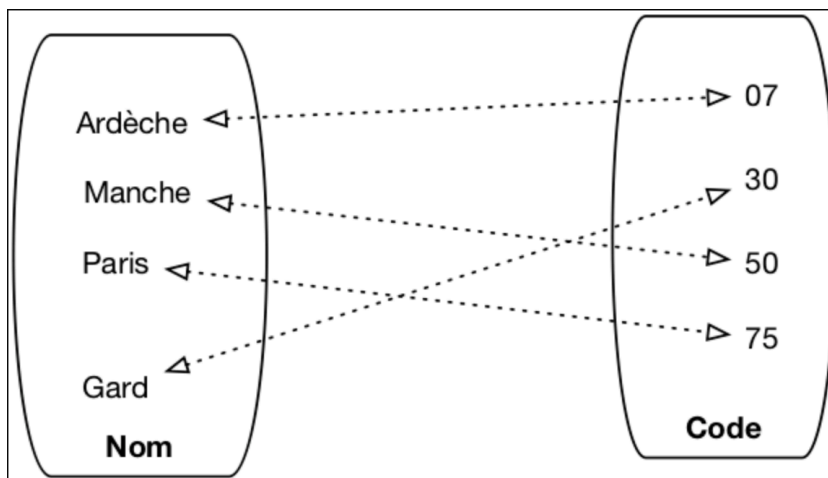
nom_produit est identifié par son code produit existant au niveau du fournisseur, du détaillant, de la facture, du grossiste, du transporteur...etc...

Représentation conceptuelle de la classe livre. isbn_10 est ici la clé primaire.

Livre	
- <u>isbn_10</u>	: int
- titre	: String
- nbre_pages	: int
- isbn_13	: String
- code_barre	: int
- code_catalogue	: int

Le modèle relationnel

Relation binaire



Dans une base de données la relation est représentée sous forme de table, plus pratique pour une représentation binaire. Cette table est composée de champs ou attributs

nom	code
Ardèche	07
Gard	30
Manche	50
Paris	75

Les associations binaires

Pour modéliser une association entre 2 classes



Les associations *un-à-un* sont les moins courantes. À titre d'exemple, dans le contexte des assurances, citons le contrat (classe) qui concerne (association) un véhicule (classe). Tout véhicule n'est concerné que par un seul contrat.

Les associations *un-à-plusieurs* (hiérarchies) sont très courantes car omniprésentes dans notre environnement. À titre d'exemple, une compagnie aérienne (classe) possède (association) des avions (classe). Tout avion n'appartient qu'à une seule compagnie.

Les associations *plusieurs-à-plusieurs* sont aussi courantes car très présentes dans notre environnement. À titre d'exemple, une compagnie aérienne (classe) loue (association) des avions (classe). Un avion peut être affrété par différentes compagnies.

Association un-a-plusieurs :

un thème inclut plusieurs sujets (ou zéro !)

Le sujet appartient au minimum à 1 thème et au maximum à 1 thème.

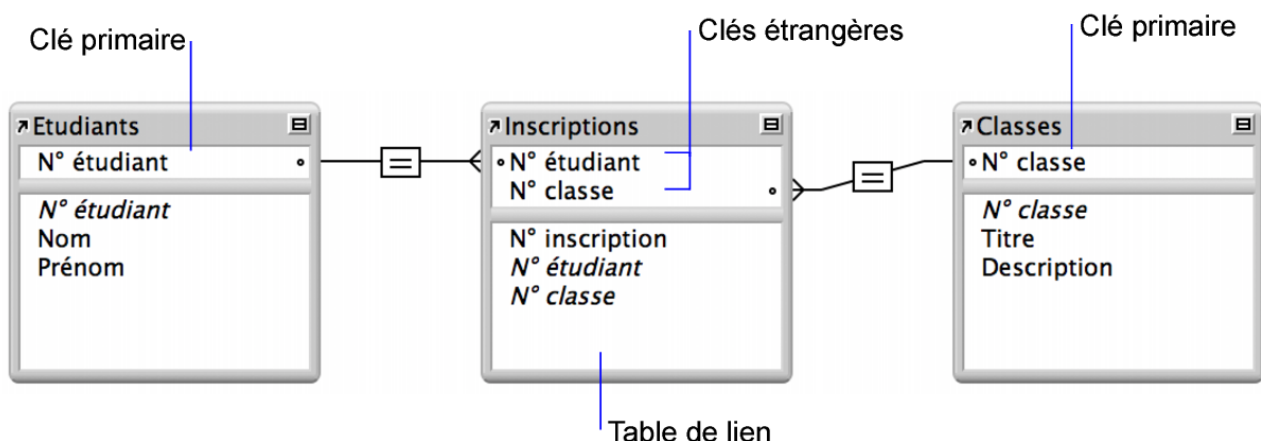
Le thème a au minimum zéro sujet et au maximum plusieurs sujets

Chaque association est nommée par un verbe ou une forme nominale la plus parlante possible.

ici : un sujet dépend d'un thème

Un *lien plusieurs à plusieurs* intervient quand plusieurs enregistrements d'une table sont associés à plusieurs enregistrements d'une autre table. Par exemple, une relation plusieurs à plusieurs existe entre les clients et les produits : les clients peuvent acheter plusieurs produits et les produits peuvent être achetés par plusieurs clients.

Les systèmes de base de données relationnelle ne vous autorisent pas à mettre en œuvre un lien plusieurs à plusieurs direct entre deux tables il faut donc créer une table intermédiaire.



les lignes d'une table sont appelées des *nuplets* ou tuples

Ordre indifférent
Jamais 2 fois le même tuple

Les clés:

Clé primaire: attribut (ou séquence d'attributs appelé **clé composite**) qui sert à distinguer un **tuple** de manière unique

Clé candidate: n'importe quel champ de table peut être une clé.

Clé étrangère [Foreign Key]: Clé qui permet de faire la liaison entre 2 tables.

Exemple de clé **composite**:

La clé est unique et sert à retrouver rapidement un élément

Une clé n'est pas forcément composée d'un seul champ , il peut y avoir plusieurs clés sur une seule table. Cette clé peut être composée

Exemple: **Table livre**

pk-composite — —					
id_table[pk]A.I.	id_livre[pk]	lang[pk]	titre	auteur	ISBN
1	1	FR	L'Origine des espèces	Darwin	112131236567
2	1	EN	On The Origin of Species	Darwin	132442225656
3	2	FR	Le gène égoïste	Dawkins	112131635644
4	2	EN	The Selfish gene	Dawkins	132442726577

L'id livre n'est pas unique, donc l'id livre n'est pas la clé .
un clé composite avec l'id_livre et la lang forment la clé composite.

```
SELECT * FROM livre WHERE id = 1
```

nous donne toutes les information du livre 1 de Darwin

```
SELECT * FROM livre WHERE id = 1 AND lang = FR
```

nous donne toutes les information de l'édition française.

Normalisation des données

Dans une **base de données relationnelle**, une **forme normale** désigne un type de relation particulier entre les **entités**. La **normalisation** consiste à construire ou restructurer une base de données pour respecter certaines formes normales, afin d'éviter la redondance des données (des données apparaissent plusieurs fois) et d'assurer l'**intégrité des données**.¹

1^{ère} forme normale: 1NF

On *ne peut pas* avoir une valeur d'attribut qui soit un tant soit peu construite, comme par exemple une liste, ou une sous-relation. Une relation est en première forme normale si toutes les valeurs d'attribut sont connues et **sémantiquement atomiques** et si elle ne contient aucun doublon. Cette contrainte conditionne tous les autres aspects du modèle relationnel, et notamment la conception, et l'interrogation.

Exemple de table qui n'est pas en 1^{ère} forme normale

nom	age
John Cavarec	30

John Afpa n'est pas atomique.

décomposer en nom, prénom

auteur	livre	année
Orwell	1984	1949
Montaigne	Essais	(1580, 1582, 1588)
Pascal	Pensées	

Cette table est en 1^{ère} forme normale ?

Remarque:

Certains attributs apparaissent posséder plusieurs éléments, mais ne sont pas toujours pour autant, "non atomique". Exemple le plus frappant: la date.

le numéro de Sécurité sociale qui est considéré atomique peut être atomiser en plusieurs champs sexe, anné_de_naissance, mois_de_naissance, département, code1, code2, code complémentaire.

¹ wikipedia

Toujours dans le cadre de la 1ere forme normale on peut avoir une table comme ceci

Table Personne

id	Nom	Prénom	Téléphone
1	Cavarec	JB	065656654,43454444

Ce n'est pas en 1NF , téléphone n'est pas atomique

on rajoute une colonne téléphone:

Nom	Prénom	Téléphone1	Téléphone2
Cavarec	Jam	065656654	434544446

Le problème est , que se passe-t-il si Jam a un 3ème téléphone ? On rajoute une colonne à la structure de notre table ?

Dans ce cas le bon sens veut qu'on crée 2 tables.

Personne

id[pk]	Nom	Prénom
1	Cavarec	JB

Et une table tph

id[pk]	id_personne[fk]	téléphone
1	1	065656654
2	1	434544446

Nom et prénom employé	Adresse	Enfant1	Âge	Enfant2	Âge	Enfant3	Âge
DURAND Albert	3, rue de la Mare - 03000 Moulins	Martin	13				
LACOMBE Josette	12, boulevard Victor Hugo - 03000 Moulins	Sylvie	10	Camille	8	Nathan	4

Que pensez-vous de cette table ?

2ème forme normale (2NF)

Elle doit être en 1ère forme normale.

La deuxième forme normale permet d'éliminer les dépendances entre des parties de clé et des attributs n'appartenant pas à une clé..

ça veut dire qu'une relation 1NF avec 1 seul attribut en clé primaire est forcément en 2ème forme normale

Par contre si un attribut ne faisant pas partie de la clé est lié à une partie de la clé, dans le cas où la clé est composée à partir de plusieurs attributs (clé **composite**), elle n'est pas en 2NF.

skater [PK]	figure [PK]	difficulte	note
Richie	Ollie	2	8
Richie	Flip	5	10
Richie	Hard-Flip	10	9
Nyjah	Flip	5	7
Nyjah	Hard-Flip	10	8

L'attribut difficulté ne dépend que de la partie de la clé primaire, donc pas en 2NF
exemple de openclassrooms.com

3ème forme normale (3NF)

Un attribut ne doit pas dépendre d'un autre attribut non **clé**:

Exemple, une table personne

id[PK]	civilité	nom	prénom	sexe
<u>1</u>	Monsieur	Darwin	Charles	M
<u>2</u>	Madame	Curie	Marie	F
<u>3</u>	Monsieur	Bigood	Joni	M

Ne respecte pas la 3NF car Monsieur peut se déduire de l'attribut sexe.

Pour régler le problème on va créer 2 tables :

Table civilité

sexe _[PK]	civilité
M	Monsieur
F	Madame

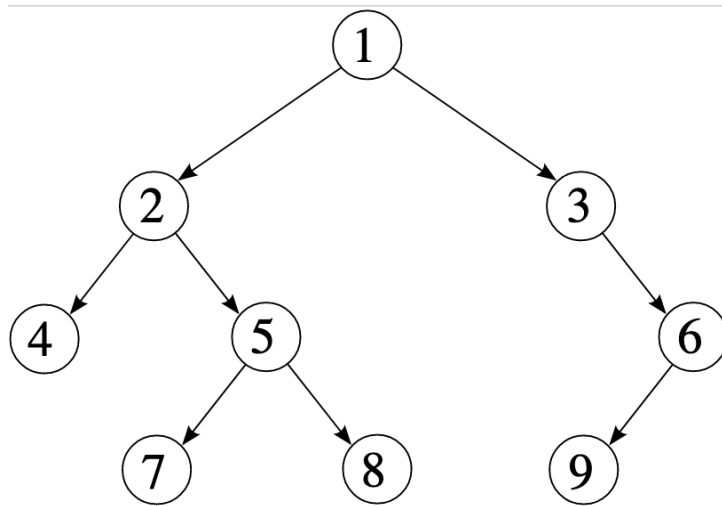
et table personne

id _[PK]	nom	prénom	sexe _[FK]
1	Darwin	Charles	M
2	Curie	Marie	F
3	Bigood	Joni	M

Et on rajoutera à la table Personne une clé étrangère qui relie la civilité

Quand on déclare un champs ou plusieurs champs comme clé un index binaire est créé

En ce qui concerne Mysql (ou MariaDB) un index binaire est créé:



- 1) racine (root)
- 2) chaque nœuds a 2 fils maximum
- 3) un noeud sans fils est appelé une feuille
- 4) Le nombre de niveaux est appelé hauteur de l'arbre