



— Conception d’une base de données

Table des matières

Outils : 3

Introduction : qu’est-ce que la méthode Merise ? 3

Le modèle conceptuel..... 3

1- Le dictionnaire des données..... 3

 a- Recueil d’informations 3

 b- Classification des données 4

 c- Les types de données..... 4

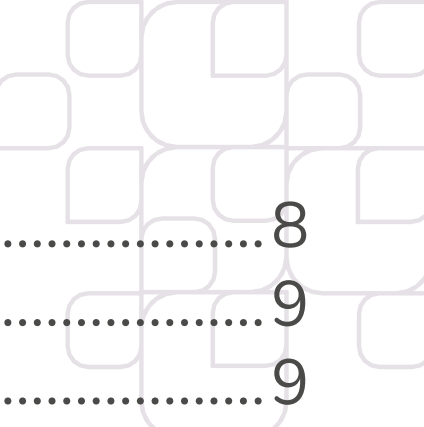
 d- Forme du dictionnaire des données..... 6

2- Le Modèle Conceptuel des Données..... 7

 a- Les entités..... 7

 b- Les propriétés 7

 c- L’identifiant..... 7



d- Les associations	8
e- Les cardinalités.....	9
f- Associations réflexives	9
g- Les contraintes d'intégrité fonctionnelle.....	10

Le modèle logique10

1- Notions de clés primaires et clés étrangères.....	10
2- Passage du MCD au MLD	11

Convention de nommage pour les bases de données14

Noms de tables.....	14
Noms de colonnes	14



Outils :

- Analyse SI : <https://launchpad.net/analyses-i>
- JMerise : <http://www.jfreesoft.com/JMerise/>
- Java Runtime Environment : <https://www.commentcamarche.net/download/telecharger-34055318-java-runtime-environment>
- Looping MCD: <https://www.looping-mcd.fr/>
- Videos : https://www.youtube.com/playlist?list=PLcOmJ-JvAV1dhlZrmJ3XiC_oLbeJCOg81

Introduction : qu'est-ce que la méthode Merise ?

MERISE est une méthode française née dans les années 70, développée initialement par Hubert Tardieu. Elle fut ensuite mise en avant dans les années 80, à la demande du Ministère de l'Industrie qui souhaitait une méthode de conception des SI.

MERISE est donc une méthode d'analyse et de conception des système d'information basée sur le principe de la séparation des données et des traitements. Elle possède un certain nombre de modèles (ou schémas) qui sont répartis sur 3 niveaux :

- Le niveau conceptuel (MCD)
- Le niveau logique ou organisationnel (MLD)
- Le niveau physique

Le modèle conceptuel

1- Le dictionnaire des données

a- Recueil d'informations

Tout comme en programmation, il est indispensable d'effectuer une collecte des données avant de construire une base de données. Cette phase est appelé « recueil d'informations ».

Ces informations peuvent être recueillies de plusieurs manières :

- Documents : cahier des charges, textes de lois, normes informatiques...
- Les entrevues : clients, utilisateurs, commerciaux, ...



- Les questionnaires (à utiliser avec prudence)

b- Classification des données

Il existe trois types de données :

- **Les données élémentaires** : l'information se confond avec la valeur prise par la donnée. Par exemple, un nom, une date... Ces données doivent être recensées de manière exhaustive.
- **Les données calculées ou déduites** : elles sont obtenues par l'application d'un traitement mathématique ou logique. Ces données sont associées à des règles de calcul ou règles de gestion. Il faut penser à bien identifier et conserver la règle de gestion qui permet d'arriver au résultat. Cette règle permettra ensuite par traitement d'obtenir le résultat désiré.
- **Les données composées** : certaines données sont regroupées en une même entité sémantique (par exemple une adresse). Ces informations doivent être décomposées en données élémentaires. Toutefois, s'il est montré qu'une donnée composée n'est jamais décomposée dans la chaîne de traitement de l'information, on peut envisager de la conserver telle quelle.

c- Les types de données

Les types numériques

- **Les nombres entiers** :

INTEGER (ou INT) et SMALLINT, permettent de coder des entiers sur 4 octets (-2.147.483.648 à 2.147.483.647) ou 2 octets (-32.768 à 32.767)

Type	Nombre d'octets	Minimum	Maximum
TINYINT	1	-128	127
SMALLINT	2	-32768	32767
MEDIUMINT	3	-8388608	8388607
INT	4	-2147483648	2147483647
BIGINT	8	-9223372036854775808	9223372036854775807



- **Les nombres décimaux**

DECIMAL (X, Y), où X et Y sont optionnels et désignent respectivement le nombre de chiffres maximum pouvant composer le nombre, et le nombre de chiffres après la virgule.
Exemple : 56.91 -> DECIMAL(4,2)

- **Les nombres à virgule flottante**

FLOAT(X), avec X définissant la précision (nombre de bits de codage de la mantisse). A utiliser seulement pour les arrondies.

Les types chaîne de caractères standard

CHAR définit des chaînes de longueur fixe (complétée à droites par des espaces, si la longueur est inférieure à X). Exemple le mot « livre » vaut CHAR(5).

VARCHAR des chaînes de longueurs variables.

Les types date

DATE qui représente une date selon un format de type "AAAA-MM-JJ" .

DATETIME qui représente une date plus une heure, dans un format tel que "AAAA-MM-JJ HH:MM:SS".

Les autres types

MONEY pour représenter des décimaux associés à une monnaie.

BOOLEAN pour représenter des booléens. Exemple : vrai ou faux.

Auto-increment : valeur spécifique de l'id.

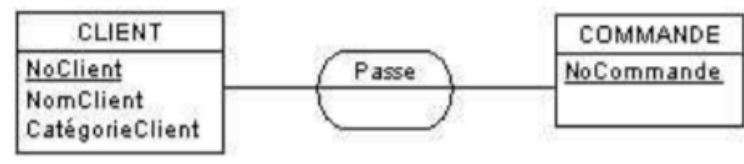


Dans le dictionnaire des données on regroupe ces différents types de données en 4 familles :

- Alphanumérique (AN) (on cherchera à déterminer la taille maximale)
- Numérique (on peut préciser entier, réel, monétaire...)
- Date (Date/Heure, Date, Heure)
- Logique ou booléen (L ou B)

d- **Forme du dictionnaire des données**

Les données sont présentées dans un tableau.



Nom	Description	Type	Taille
NoClient	N° de client	INT_Auto_increment	25
NomClient	Nom du client	Varchar	100
CatégorieClient	Catégorie du client	Varchar	100
NoCommande	Identifiant de la commande	INT_Auto_increment	25



2- Le Modèle Conceptuel des Données

Après avoir réalisé le dictionnaire des données nous pouvons passer à l'étape suivante : le MCD

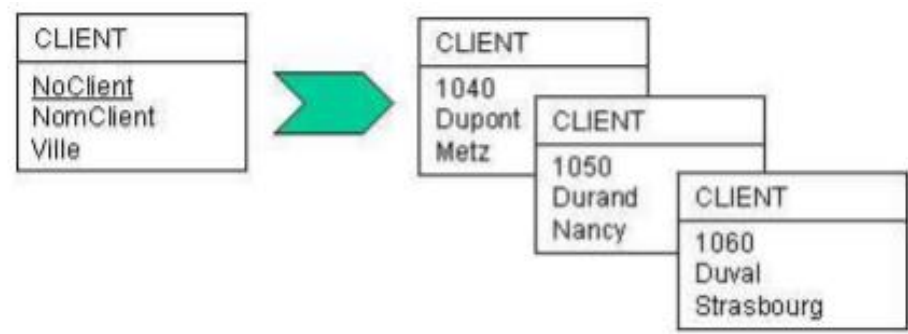


a- Les entités

Elles correspondent aux objets du système d'information. Par exemple, l'entité CLIENT rassemble toutes les informations communes aux clients de l'entreprise.

- Les entités sont nommées
- Leur nom est unique dans le modèle et s'écrit en majuscule
- Leur nom est unique dans le modèle et s'écrit en majuscule

Dans le système d'information, l'entité CLIENT représentera l'ensemble des clients. Chaque client constitue alors une **occurrence** de l'entité. Une occurrence représente un « exemplaire » de l'objet.



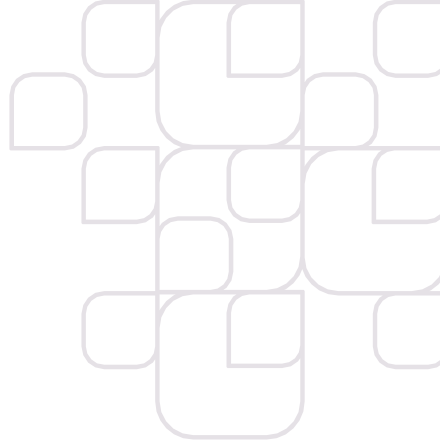
b- Les propriétés

Une entité possède toujours au moins une propriété. Chacune de ces propriétés doit pouvoir être valorisée de manière unique. Les propriétés des entités sont issues du dictionnaire des données.

c- L'identifiant

Parmi les propriétés d'une entité, il y en a une qui joue un rôle particulier. En effet, l'identifiant d'une l'entité est une propriété telle qu'à chaque valeur de la propriété corresponde une et une seule occurrence de l'entité. Donc :

- On ne peut trouver deux occurrences d'une entité ayant le même identifiant.



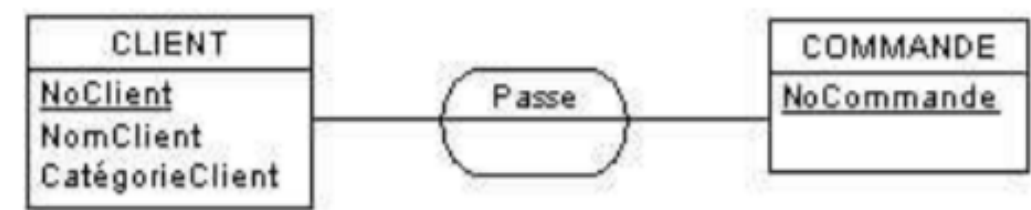
- Si l'entité ne possède qu'une propriété, cette propriété est l'identifiant
- Toutes les propriétés sont en dépendance fonctionnelle élémentaire et directe avec l'identifiant
- L'identifiant est représenté souligné dans le modèle.

d- Les associations

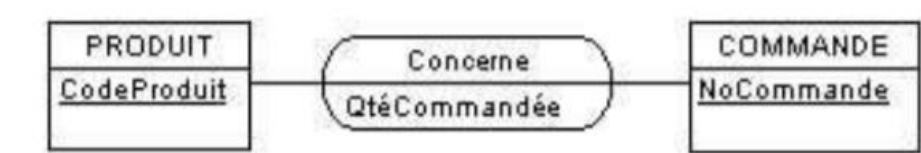
Une association est un lien entre des entités. Contrairement aux entités, les associations n'ont pas d'existence propre mais elles peuvent porter des propriétés.

Exemples :

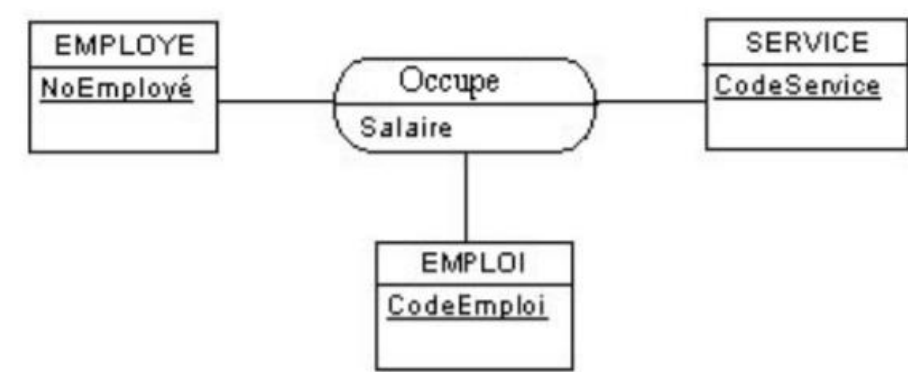
Association binaire :



Association porteuse de propriété :



Association ternaire :



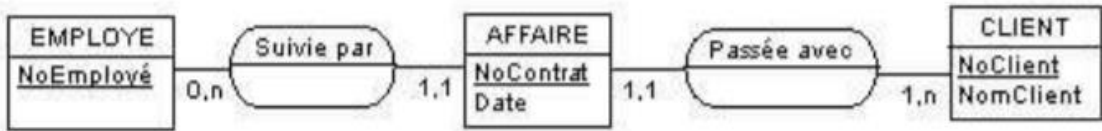
Attention : les associations ternaires sont assez rares et peuvent très souvent être remplacées par une entité supplémentaire.



e- Les cardinalités

- La cardinalité représente le nombre d’occurrences minimum et maximum d’une association par rapport à une entité.
- La cardinalité minimale représente le nombre de fois « au minimum » où une occurrence de l’association participe aux occurrences de l’entité. Cette cardinalité est choisie parmi 0 ou 1.
 - La cardinalité maximale représente le nombre de fois « au maximum » où une occurrence de l’association participe aux occurrences de l’entité. Cette cardinalité est choisie parmi 1 ou n où n indique une cardinalité maximale supérieure à 1 mais non quantifiée. Si la valeur de n est connue, on peut la mentionner.

Exemple :

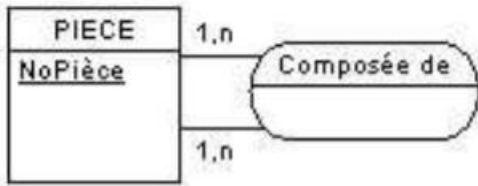


- Un employé peut suivre (0) ou plusieurs (n) affaires
- Une affaire est passée avec un (et un seul) client
- Un client passe une (sinon ce n’est pas un client) ou plusieurs (n) affaires

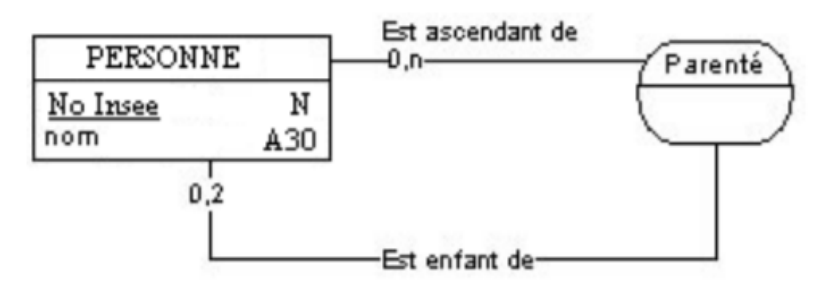
La cardinalité minimale à 1 ou à 0 dépend des règles de gestion. Si sur le schéma précédent, on admettait à gauche une cardinalité 0,n cela voudrait dire qu'il existe des clients n'ayant pas passé d'affaire.

f- Associations réflexives

Ce sont des associations qui relient deux fois la même entité.



Une pièce (en mécanique par exemple) est composée de 1 ou plusieurs pièces.
Dans les associations réflexives, la notion de rôle prend tout son sens.

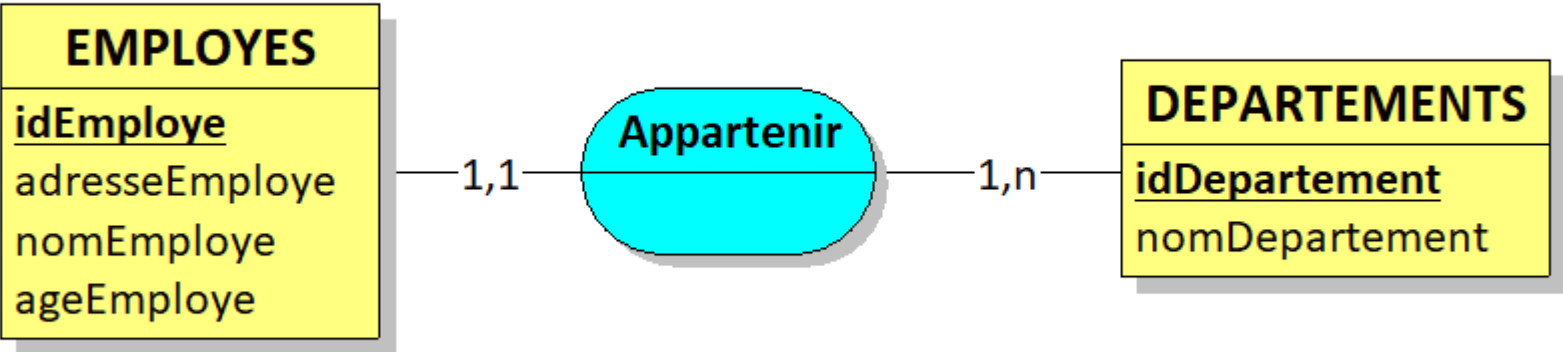


g- Les contraintes d'intégrité fonctionnelle

Quand on détermine une association présentant une cardinalité maximale à 1, on est en présence d'une contrainte d'intégrité fonctionnelle (CIF). Une CIF traduit la présence d'une dépendance fonctionnelle entre les identifiants des entités participant à l'association.

Le modèle logique

1- Notions de clés primaires et clés étrangères



Voici un MCD comme nous l'avons abordé précédemment.

Ici la clé primaire est l'identifiant de l'entité.

- IdEmploye pour l'entité EMPLOYE



- IdDepartement pour l'entité DEPARTEMENT
- Le modèle logique des données va introduire la notion de clés étrangères.

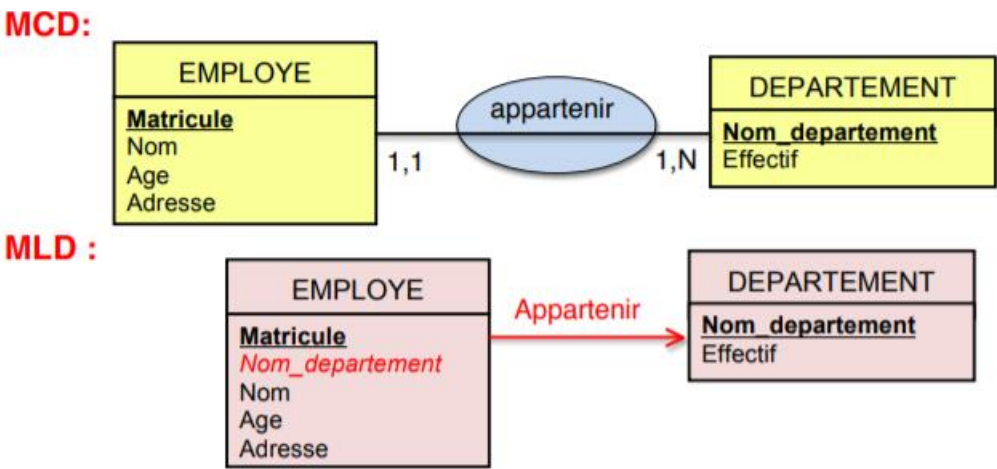
Voici le MLD découlant du MCD vu ci-dessus :

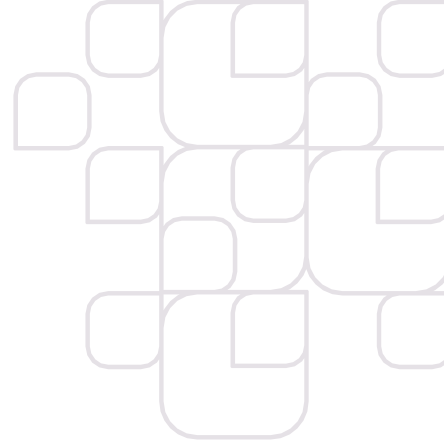


On peut voir l'apparition d'une clé étrangère dans l'entité EMPLOYE : idDepartement.
Grâce à cette clé étrangère nous allons pouvoir voir dans quelle département travail chaque employé de l'entreprise.

2- Passage du MCD au MLD

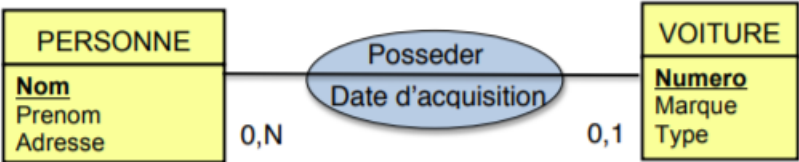
- Avec une relation (*,n)- (1,1)





- Avec une relation (0/1,N)-(0,1)

MCD :

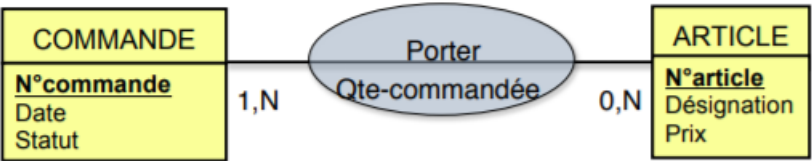


MLD :

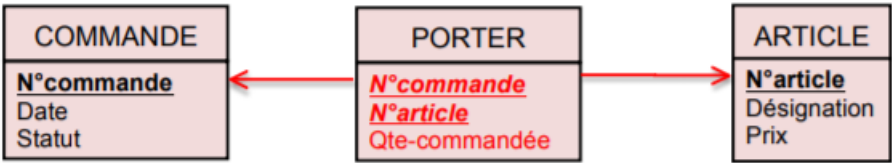


- Avec une relation (0/1,N)-(0/1,N)

MCD :



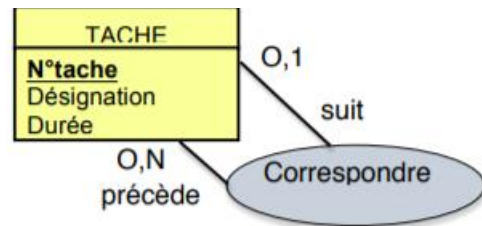
MLD :



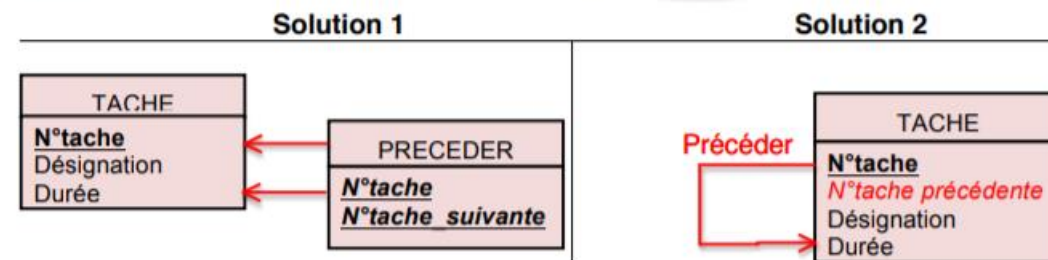


- Avec une relation réflexives (0,N)-(0,1)

MCD :



MLD :



En résumé le modèle conceptuel de données (MCD) permet :

- Permet de modéliser la sémantique des informations d'une façon compréhensible par l'utilisateur de la future base de données.
- Utilise le formalisme (graphique) Entité-Relation.
- Ne permet pas d'implémentation informatique de la base de données dans un SGBD (Système de Gestion de Base de Données).

Le modèle logique de données (MLD) :

- Permet de modéliser la structure selon laquelle les données seront stockées dans la future base de données
- Est adapté à une famille de SGBD : ici les MySQL
- Utilise le formalisme graphique Merise
- Permet d'implémenter la base de données dans un SGBD donné
- Et surtout ce modèle est généré automatiquement à partir du MCD !



Convention de nommage pour les bases de données

Noms de tables

Voici une liste de bonnes pratiques :

- Utiliser un nom représentatif du contenu
- Utiliser un seul mot lorsque c'est possible
- Privilégier le singulier (mais c'est parfois un grand débat ...)
- Penser à des noms génériques et envisager les futures évolutions. Par exemple une table "client" qui pourrait aussi contenir les prospects et commerciaux devrait plutôt s'intituler "utilisateur"
- Préfixer les noms des tables
- Permet d'éviter d'utiliser accidentellement des mots réservés.
- Permet d'éviter les conflits lorsqu'il y a plusieurs logiciels similaires sur une même base de données (par exemple, si 2 logiciels utilisent chacun une table intitulée "utilisateur").
- Utile pour séparer facilement les tables associées à un système ou à un autre. Par exemple si un blog WordPress et une boutique e-commerce Prestashop sont placés sur une même base de données, le blog aura des tables commençant par "wp_" tandis que la boutique aura des tables commençant par "ps_".
- C'est plus simple pour ré-installer un backup. Par exemple, pour réinstaller une sauvegarde du blog, il est possible d'ajouter des tables commençant par "wp2013_" puis de modifier le code de l'application pour tout migrer d'un coup.
- Sur des gros projets ça peut être pratique pour que toutes les tables associées aux utilisateurs commencent par exemple par "user_", toutes celles concernant les produits par "product_" et ainsi de suite.

Noms de colonnes

Voici la liste de bonnes pratiques :

- **Préfixer toutes les colonnes** de la même façon pour chaque table. C'est beaucoup plus pratique lorsqu'il convient d'effectuer des jointures.
- Dans le cas d'un site à vocation multilingue : **indiquer la langue et la zone géographie** pour les champs alphanumériques (fr_fr pour le français de France, fr_ca pour le français du Canada, fr_be pour le français de Belgique ...). C'est extrêmement pratique si un jour une application doit devenir multilingue. Si la base de données doit s'internationaliser il suffira d'ajouter une colonne supplémentaire avec les traductions.
- Lorsqu'une clé étrangère est utilisée (traduction anglaise : "Foreign Key"), il est pratique de l'indiquer dans le nom de la colonne. La colonne peut contenir le préfixe, puis "fk" pour Foreign Key, puis le nom de la table et enfin se terminer par "id". Ainsi, une colonne pourrait s'intituler "wp_fk_user_id" (cf. préfixe "wp", foreign key sur la table utilisateur de la colonne "id").
- Toujours intitulé de façon similaire certains champs tels que les DATE ou DATETIME. Cela permet d'aider un développeur à savoir ce que va contenir un champ sans nécessairement regarder le contenu.