

7. Codification des valeurs

Dès que l'on commence d'établir un modèle des données dans la pratique on se trouve confronté au besoin de systématiser la gestion de certaines informations en codifiant leurs valeurs.

Exemples:

Pays, monnaies, unités de mesure, langues, catégories d'articles, catégories de client, notes de qualité, état d'avancement d'un produit, ...

Nous avons montré au chapitre 6 un exemple décrivant les connaissances linguistiques de personnes:

Personne (#-personne, nom, prénom, sexe, date-naissance)

Langue parlée (#-personne, langue, degré de connaissance)

Rien que dans cet exemple, nous trouvons trois types d'information qu'il faut absolument codifier. Il s'agit bien sûr de *sexe*, *langue* et *degré de connaissance*. Il serait en effet inadmissible qu'un utilisateur, intentionnellement ou par erreur, saisisse une valeur inacceptable dans l'un de ces champs. Il faut, dès le départ, définir une table des valeurs autorisées pour chacun de ces types d'informations.

Le problème devient encore plus complexe si l'on prend en compte le fait que les utilisateurs du futur système travailleront éventuellement en différentes langues et que ces informations pourront figurer sur des documents qui seront lus par des destinataires parlant des langues différentes. Les tables contenant la définition des codes devront souvent être multilingues.

Prenons l'exemple d'un collaborateur parlant l'allemand qui doit saisir les données d'une cliente privée situé en Grande-Bretagne. Dans le champ *titre* de l'adresse il voudra spécifier "Frau" (Madame), mais sur le bulletin de livraison et la facture devra apparaître "Mrs."

Les langues elles-mêmes peuvent être définies en plusieurs langues. Voici une solution:

Langue (#-langue, #-langue-référence, code-abrégé, libellé langue)

#-langue: une numérotation unique de toutes les langues utiles
#-langue-référence: le numéro de la langue de l'utilisateur
Code-abrégé: une lettre spécifiant la langue dans la langue de l'utilisateur
Libellé-langue: le nom de la langue en toutes lettres dans la langue de l'utilisateur

Langues			
#-langue	#-langue référence	Code-abrégé	Libellé-langue
1	1	F	Français
1	2	F	Französisch
1	3	F	French
2	1	A	Allemand
2	2	D	Deutsch
2	3	G	German

Les langues sont donc numérotées comme suit: 1: français, 2: allemand, 3: anglais

La deuxième ligne de la table spécifie le français en allemand, la troisième le français en anglais; la quatrième l'allemand en français, la cinquième l'allemand en allemand, la sixième l'allemand en anglais.

Reprenant l'exemple du début de ce chapitre, nous allons codifier *sexe*, *langue* et *degré de connaissance* et aurons alors:

Personne (#-personne, nom, prénom, #-sexe, date-naissance)

Langue parlée (#-personne, #-langue, #-degré de connaissance)

Nous avons donc transformé ces attributs en clés étrangères dont les valeurs doivent être définies dans des tables correspondantes.

Au secours! criera le lecteur, bien conscient de la nécessité de procéder à cette codification, mais soucieux du nombre de nouvelles tables à définir et à gérer. Nous allons proposer une solution à ce problème dans la prochaine section. Cette solution s'appelle Table des codes.

Table des codes

Nous allons créer une table unique, la Table des codes, silo contenant la définition de **toutes** les valeurs de codifications **simples**. Elle sera divisée en sous-tables qui contiennent chacune la définition d'un type d'information codifiée.

Les esprits pointilleux objecteront ici que a) nous parlons déjà d'implémentation de la base de données et que b) nous allons fourrer dans une seule table des types d'objets différents. Ils ont en principe raison.

Nous affirmons par contre que le concept de table des codes a une **immense valeur conceptuelle**. Sachant qu'une telle table existe, avec un ensemble de fonctionnalités conviviales et puissantes que nous décrirons par la suite, nous allons lors de la modélisation des données:

Codifier sans remord ni arrière-pensée tout ce qui doit ou vaut la peine d'être codifié.

Tombant sur un attribut devant être codifié, nous dirons simplement: "Table des codes" et en ferons une clé étrangère, éventuellement en lui attribuant un numéro de sous-table.

Ceci sans nous soucier du travail et du coût que pourrait engendrer la création d'une nouvelle table, parce que l'infrastructure Table des codes sera développée une fois pour toutes pour gérer l'ensemble des codifications nécessaires.

Dans la pratique, une table des codes réelle va contenir des dizaines de sous-tables.

Dans le monde globalisé que nous connaissons, mieux vaut créer dès le départ une table des codes multilingue. Voici comment elle sera définie:

Table des codes (#-sous-table, #-code, #-langue, code-abrégé, libellé)

Il est logique de définir la sous-table 0 comme la table définissant les sous tables, par exemple spécifiant que la sous-table 1 contient la définition des langues, la sous-table 2 celle des titres de personnes (M, Mme, Mlle, ...), etc.

Table des codes

#-sous-table	#-code	#-langue	Code abrégé	Libellé
0	0	1		Table des sous-tables
0	1	1		Langues
0	2	1		Titres
0	3	1		Genres (sexes)
1	1	1	F	Français
1	1	2	F	Französisch
1	1	3	F	French
1	2	1	Al	Allemand
1	2	2	D	Deutsch
1	2	3	G	German
1	3	1	An	Anglais
1	3	2	E	Englisch
1	3	3	E	English
2	1	1	M	Monsieur
2	2	1	Mme	Madame
2	1	2	H	Herr
2	2	2	F	Frau
2	1	3	Mr	Mr
2	2	3	Mrs	Mrs
3	1	1	M	Mâle
3	2	1	F	Femelle
3	1	2	M	Männlich
3	2	2	W	Weiblich
3	1	3	M	Male
3	2	3	F	Female

Fonctionnalités rattachées à la table des codes

Une infrastructure de table des codes performante doit comprendre les fonctionnalités suivantes:

- Possibilité d'afficher et de lister chacune des sous-tables dans n'importe quelle langue, y compris la table des sous-tables.
- Programme de saisie permettant de saisir ou modifier confortablement les valeurs d'une sous-table, en une ou plusieurs langues.
- Fonctions qui ramènent dans un programme pour l'affichage ou l'impression le code abrégé ou le libellé d'un code:

```
print code_abrégé(#-sous-table, #-code, #-langue)
display libellé(#-sous-table, #-code, #-langue)
```

- Fonction qui affiche l'ensemble des codes abrégés ou des libellés d'une sous-table dans une langue donnée et permet de sélectionner une de ces valeurs ("pop up")
- Fonction qui vérifie si une valeur donnée est dans une certaine langue une valeur valable d'une certaine sous-table.

Table des codes encore plus sophistiquée?

Certaines sociétés ont jugé utile pour leur implémentation de la table des codes de voir encore plus sophistiqué, notamment en rajoutant d'autres attributs, au contenu souvent laissé libre et potentiellement différent de sous-table à sous-table. Nous n'y sommes pas très favorables.

Nous sommes au contraire d'avis que la table des codes doit rester un concept simple qui apporte les avantages principaux suivants:

- Lors de la conception, donc de la modélisation des données, elle constitue un "placard mental" pour la codification de tous les types d'information simples qui doivent être codifiés, sans que l'on doive chaque fois définir une nouvelle table.
- Lors de l'implémentation, elle constitue une infrastructure puissante qui, une fois réalisée, simplifie considérablement la réalisation du reste de l'application.
- Lors de l'utilisation, un endroit unique pour gérer tous les paramètres de l'application.

Table des codes: les limites

Que ne faut-il pas fourrer dans la table des codes?

Certainement pas des types d'objets qui doivent être décrits au moyen d'attributs autres qu'un code abrégé et un libellé.

Deux types d'objets qui apparaissent dans la majorité des applications sont les pays et les monnaies. Il ne sera possible de les reléguer dans une table des codes que dans des cas extrêmement simplistes.

Pour la majorité des applications, il faudra bien définir des tables séparées pour décrire ces types d'objets.

Voici des définitions possibles:

Pays (#-pays, code-pays, libellé-pays, #-monnaie, #-langue)

Monnaie (#-monnaie, code-monnaie, libellé, taux-de-change)

Il existe bien des codes officiels pour les pays et pour les monnaies, mais vu que ceux-ci sont tout de même susceptibles de changer un jour (le Zaïre devenant le Congo, par exemple), mieux vaut définir une clé primaire neutre ici aussi.

Dans la définition de *Monnaie* ci-dessus, la présence de taux-de-change n'est évidemment valable que par rapport à une monnaie de référence unique. S'il s'agit d'enregistrer des taux croisés, une autre table sera bien sûr nécessaire.

Table des textes

Faut-il créer d'autres tables sur le modèle de la table des codes?

Un candidat sérieux est constitué par une ou plusieurs **tables des textes**.

La nécessité de définir une telle table se présente notamment lorsqu'on est confronté au problème de gérer plusieurs langues, ce qui est pratiquement toujours le cas.

La table des textes peut contenir des textes en plusieurs langues tels que, par exemple, les libellés des articles vendus, des descriptifs, etc.

Mais la table des textes peut aussi comprendre les textes qui constituent les masques des écrans des utilisateurs d'une application, qui apparaissent sur les états et listes produits par les applications ou sur les pages des sites Web. Ces écrans, listes, pages deviennent ainsi totalement dynamiques.

Voici une définition possible d'une table des textes:

Textes (#-table, #-texte, #-langue, texte)

Théoriquement, la table des textes pourrait simplement s'intégrer à la table des codes, mais elle prend en général une telle ampleur qu'il vaut mieux en faire une table séparée.