

Organisation des données dans les tables

Catégorie : Fiches techniques

Publié par [Fooups](#) le 16/04/2005

Avant de se lancer dans la programmation proprement dite, vous aurez besoin de créer les tables pour le stockage des données.

Une part importante du code découlant de la structure de ces tables, mieux vaut partir sur des bases saines, plutôt que d'être obligé en cours de route de changer la répartition des données, et être obligé de récrire les requêtes et le code qui les exploite!

Ce guide vous explique comment répartir les données dans vos tables en suivant les règles de normalisation.

Vous trouverez peut-être sa lecture superflue

, dommage car il s'agit de notions essentielles. **1) Normalisation des tables** Normaliser ses tables consiste à construire celles-ci selon des règles permettant notamment

->D'éviter toute duplication d'information ->D'accéder aux données de manière unique et rationnelle. Cette normalisation est importante car elle apporte :

->des requêtes plus simples à écrire,

->des données plus facilement accessibles ;

->une meilleure intégrité des données ;

->la diminution des erreurs lors de l'insertion ou de la suppression de nouvelles

Il s'agit de règles de bon sens, que l'on applique peut-être sans le savoir, mais leur formalisation permet de vérifier que vos tables respectent ces règles. Il existe 5 formes normales, mais nous ne traiterons que les 3 premières qui sont les plus importantes. **2) Première forme normale** Cette règle stipule 3 points

Les champs de chaque table doivent être atomiques; c'est à dire qu'on ne peut pas les décomposer.

Exemple d'une table (non conforme) d'un carnet d'adresse :

Nom	Adresse	Ville
Albert Martin	12 avenue Jean Jaures	F-69003
Lyon	Benoît Solo	71 rue des martyrs B-1070 Bruxelles

L'exemple ci-dessus n'est pas conforme puisque le champ nom contient à la fois le nom et le prénom, le champ ville, la ville et le code postal.

Pour normaliser cette table, il suffira de scinder certains champs :

Nom	Prénom	Adresse
Code_postal	Ville	
Martin	Albert	12 avenue Jean Jaures
F-69003	Lyon	
Solo	Benoît	71 rue des martyrs
B-10708	Bruxelles	

Maintenant que tous les champs sont atomiques, les tri et les recherches seront plus simples et plus directs.

Il ne peut exister de champs vides.

Exemple d'une table (non conforme) d'une gestion de DVD

id_user	dvd_1	dvd_2	dvd_3
205	Psychose	Moby Dick	
206	Men in black	Le grand bleu	
Fahrenheit 451			

Cette table n'est pas conforme puisqu'elle contient des champs r p titifs. Il faudra donc la scinder en 2 tables :

Table 1

id_dvd	Titre	Id_emprunteur
43	Psychose	205
44	Men in black	206
45	Les choristes	207

Table 2

id_user	nom	pr�nom
205	Leroy	Marc
206	Lascience	Olivier
207	Lechat	Claude

Chaque champ doit avoir une signification pr cise et constante dans le temps.

Exemple d'une table (non conforme) de gestion de la production d'une ferme

animal	date	quantit�
Poule01	31/07/2004	2
Vache01	31/07/2004	25
Poule02	31/07/2004	3

Cette table n'est

pas conforme puisque le champ quantit  peut repr senter des litres de lait ou des nombres d'oeufs. Pour  tre conforme   la norme, il faudrait avoir une table pour les poules, une table pour les vaches, mais cette normalisation a ses limites, car si cette ferme produit  galement des quintaux de bl , des tonnes de fourrage,  c s . Une solution simple (mais non rigoureusement conforme) consiste   rajouter un champ unit .

3) Seconde forme normale

Le respect de la seconde forme normale est  galement important, m me si sa d finition peut para tre un peu floue :

Toutes les propri t s non-cl  doivent  tre totalement d pendantes de la totalit  de la cl  primaire

Exemple d'une table (non conforme) g rant les heures des ouvriers d'un atelier

Heures	NumSalari�	Nom	NumAtelier
	20036	Durand	1
	20036	Durand	2
	36900	Leroux	2
	45002	Frank	3
	45002	Frank	1

Cette table respecte la 1 re forme normale, mais ne respecte pas la seconde. Si nous fixons p.ex. la clef primaire NumSalari  + NumAtelier, le champ (non-clef) heures est bien en totale d pendance de la clef primaire, puisqu'  partir de cette clef, nous pouvons isoler un compte

d'heures unique pour le couple NumSalari  + NumAtelier. Par contre nous ne pouvons pas le faire pour le champ (non-clef) Nom, qui ne d pend que d'un morceau de la clef primaire. Vous n'avez pas tout compris ? Regardez comment ci-dessous comment cette table a  t  scind e pour  tre normalis e.

Table 1

NumSalari�	Nom
20036	Durand
36900	Leroux
450002	Frank

Table 2

NumSalari�	NumAtelier	Heures
20036	1	18,5
20036	2	6,7
36900	2	8,5
45002	3	23,5
45002	1	4,8

Clef primaire : Numsalari  + Numatelier Deux avantages : ->on  vite ainsi la duplication de renseignements (le nom du salari  n'appara t plus qu'une seule fois) ->on peut d truire des enregistrements d'heures sans cons quence sur les informations relatives au salari  **4) Troisi me forme normale**

Un champ non-cl  ne doit  tre en d pendance transitive avec la cl  primaire. Autrement dit, si la valeur d'un champ   non-cl    peut  tre d duite de la valeur d'un autre champ   non-cl    alors sa relation   la cl  primaire est transitive (puisque'elle transite par un autre champ) et la table n'est pas dans la troisi me forme normale. Exemple d'une table (non conforme) g rant les employ s d'une entreprise :

Nom	NumSalari�	Date_naissance
Service	NomService	NumChef
Durand	5001	15/01/1948
Vente	4580	
Martin	5002	12/04/1957
Informatique	4120	

Dans cet exemple, il est possible de d terminer le nom du service et le code salari  de son chef uniquement   partir du code service qui est un champ   non-clef  . Quels sont les risques ? Si nous supprimons tous les employ s d'un service donn , lors de la suppression du dernier enregistrement nous perdrons  galement les informations concernant le service lui-m me (nom du service et n  du chef). De la m me fa on, si on cr e le nouveau service dans l'entreprise, nous ne pourr  pas l'ajouter tant qu'il n'y aura pas un salari  affect    ce service. La solution passe par un d coupage de la table en deux autres tables r pondant chacune aux 3 premi res formes normales.

Table 1

Service	NumSalariÃ©	Nom	Date_Naiss	
	5001	Durand	15/01/1948	5
	5002	Martin	12/04/1957	6

Table 2

Service	Nom	NumSalariÃ©_Chef
5	Vente	4580
6	Informatique	4120