Normalisation

Quand on a affaire à un ensemble de données que l'on veut incorporer dans une base de données, la question fondamentale qui est posée est la suivante:

**quelle devra être la structure logique appropriée ?**, en d'autres mots: quelles seront les relation (tables) et les attributs nécessaires?

Cette structure logique choisie aura une influence profonde sur les opérations d'ajout, de suppression et de modification de la base. Il importe donc de faire un choix judicieux.

La théorie de la normalisation est un aide utile et pratique dans le design des bases de données, mais la normalisation ne devrait pas être le facteur unique dans la décision portant sur la structure logique d'une base.

La théorie de la normalisation est basée sur le concept des formes normales (FN).

En pratique, **la première et la deuxième forme normale** sont nécessaires pour avoir un modèle relationnel juste. **Les formes normales supplémentaires ont leurs avantages et leurs inconvénients**.

**Les avantages sont :**

* de limiter les redondances de données (multiples écritures)
* donc de limiter l'espace disque nécessaire
* de limiter les incohérences de données qui pourraient les rendre inutilisables
* d'éviter les processus de mise à jour (réécritures)

**Les inconvénients** sont :

* des temps d'accès potentiellement plus longs si les requêtes sont trop complexes (lectures plus lentes), cela dit, avec des index, notamment sur les clefs primaires, on peut généralement obtenir des résultats satisfaisants
* une plus grande fragilité des données étant donné la non redondance (lecture impossible)
* la difficulté de réorganiser la structure (dans le cas où l'on ne sait pas à l'avance le lien entre les entités, ce qui est à éviter si possible)

Pour des petites bases de données, se limiter à la troisième forme normale est généralement une des meilleures solutions d'un point de vue architecture de base de données, mais pour des bases de données plus importantes, cela n'est pas toujours le cas. Il s'agit de faire l'équilibre entre deux options :

* la génération dynamique des données via les jointures entre tables
* l'utilisation statiques de données correctement mises à jour

Si une table (base de donnée) est **plus intensivement écrite que lue** il sera préférable **de normaliser le plus possible**. À contrario, si une table (base de donnée) **est plus intensivement lue qu'écrite** il pourra **être judicieux d'être moins strict sur le respect de la normalisation** pour permettre d'améliorer les performances **d'accès aux données**.

Il convient d'être prudent lorsqu'on renonce à la forme normale. Il n'est pas garanti qu'une forme dénormalisée améliore les temps d'accès. En effet, la redondance peut entraîner une explosion des volumes de données qui peuvent écrouler les performances ou saturer les disques durs. La normalisation des modèles de données a été popularisée principalement par la méthode [Merise](http://fr.wikipedia.org/wiki/Merise_(informatique)). La principale limite de la normalisation est que les données doivent se trouver dans une même base de données (dans un seul schéma).