

TP 2 : Relations entre tables, normalisation et SQL

Il existe trois types de relations :

- One-to-one
- One-to-many
- Many-to-many

La relation one to one est utilisée seulement pour optimiser des performances lors de requête. Nous ne nous attarderons pas sur cette relation dans ce TP. Nous allons aborder les deux relations restantes.

Pour toutes commandes SQL, se référer à <http://www.w3schools.com/sql/default.asp>.

1. Relation one-to-many

Imaginons la configuration suivante : nous aimerions créer une librairie numérique à l'aide d'une base de données. Les informations devant apparaître dans la base de données seront les suivantes :

- Titre du livre
- Nom de l'auteur
- Prénom de l'auteur
- Maison d'édition
- Année de publication

Tâche #1 : Sur papier, créez un design de votre base de données à l'aide de la représentation UML. Identifiez les relations one-to-multi.

Tâche #2 : Sous Microsoft Access 2007, créer les différentes tables nécessaires précédemment définies dans la tâche #1.

Tâche #3 : Créez les relations comme définis dans la tâche #1 à l'aide du menu contextuel « Outils de base de données » - « Relations ».

Tâche #4 : Importez les données Excel afin de peupler les différentes tables créées. Observez les tables « editor » et « author ». Quel est l'intérêt d'avoir créé la relation entre les différentes tables ?

Tâche #5 : A l'aide du langage SQL, créez une requête pour obtenir les livres de Daniel Ward. Pour cela, créez une requête pour obtenir l'index correspondant à cet auteur. Par la suite, créez une seconde requête en utilisant l'index précédemment obtenu pour extraire les livres écrits par cet auteur.

Tâche #6 : Effectuez ces deux opérations en une seule requête

Tâche #7 : Répétez la tâche précédente en utilisant le principe de tableau join.

Tâche #8 : Joignez les tables book et editor afin d'obtenir le nom de l'éditeur au lieu de l'index et effectuez la même recherche que précédemment.

2. Relation many-to-many

Imaginez le même problème que dans la section ci-dessus. En revanche, chaque livre ne comportera un seul auteur mais plusieurs.

Tâche #9 : Répétez la tâche #1. Quel(s) est(sont) les relations multi-to-multi ?

Tâche #10 : Répétez les tâche #2 à #4.

Tâche #11 : Trouvez les auteurs de l'œuvre indexée 1300. Construire une seconde recherche afin de trouver le nom et prénom correspondant aux indices trouvés précédemment.

Tâche #12 : Créez une seule requête en utilisant le principe de jointure de table similairement à la tâche #8.

3. Normalisation

Tâche #13 : Importez dans une table le fichier « ugly-table.xlsx ». Répertoriez les différents problèmes avec ce design.

Tâche #14 : Sur papier, changez le design de la table pour obtenir une base de données 1NF. 1NF ne doit pas comporter de redondance au niveau des champs.

Tâche #15 : Toujours sur papier, changez le design de la table pour obtenir une base de données 2NF. 2NF ne doit pas comporter de redondance dans les valeurs des champs.

Tâche #16 : Toujours sur papier, changez le design de la table pour obtenir une base de données 3NF. 3NF implique de créer des séparations si les données dans une table ne sont pas connectées.

Tâche #17 : Convertissez la table d'origine aux design 3NF et copier les données. Il faudra pour cela :

- Créez des tables pour les auteurs, éditeurs, genre et reliure.
- Importez les données en utilisant des requêtes SQL.
- Créez des tableaux temporaires afin d'associer la clé primaire des livres et celui des tables précédemment créés.
- Joignez les tableaux pour obtenir le tableau livre avec seulement les IDs nécessaires.
- Créez les relations entre les différents tables.