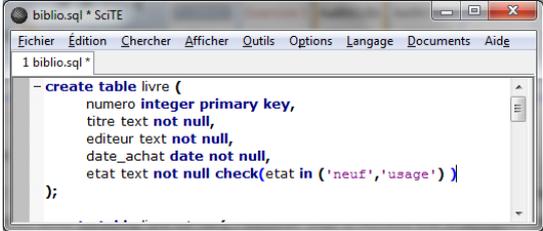


TP : Définition de Données sous Access

Objectif du TP

Dans les séances précédentes, nous avons vu comment concevoir le schéma relationnel d'une base de données *mais sur papier* ou comment construire la structure de la Base de Données avec Access en utilisant un assistant logiciel. L'objectif de ce TP est d'utiliser uniquement des *commandes SQL* (`create table`, `alter table`, `drop table` ...) afin de construire le schéma relationnel d'une BD ainsi que les contraintes d'intégrité (clés primaires et étrangères...). Dans un second temps, la modification de la structure de la BD ainsi que l'ajout de données seront abordés. L'aide en ligne des instructions SQL en Access se trouve à cet [endroit](#).

DÉFINITION DE LA STRUCTURE D'UNE BD

1. Créez une base de données vide intitulée LDD.
2. Access n'est pas spécialement prévu pour éditer du code SQL (puisque'il possède souvent des assistants graphiques pour faire ce dont vous avez besoin). De ce fait, l'éditeur SQL d'Access est vraiment rudimentaire. Nous vous conseillons d'utiliser un éditeur externe ([Scintilla Text Editor](#) par exemple qui est déjà installé sur les machines du département) afin de sauvegarder toutes les instructions SQL. Vous pourrez aisément copier-couper-coller (CTRL+C, CTRL+X, CTRL+V) vos instructions SQL d'une application à l'autre. Grâce à un éditeur externe, vous pourrez avoir une *coloration syntaxique* de vos instructions mais seulement si vous enregistrez le fichier avec l'extension `.sql`. De cette façon, vous pourrez savoir si l'instruction est au format SQL ou non. Vous pourrez également mieux *formater et structurer* vos documents. Enfin, vous pourrez conserver un historique des instructions SQL que vous aurez saisies et les charger dans un autre SGBD en chargeant simplement le fichier que vous aurez créé.

3. Pour saisir une instruction SQL avec Access, allez dans le menu `Créer > Création de requête` puis fermer la fenêtre graphique qui permet d'afficher des tables et passez l'affichage du mode `création` au mode `SQL`. Chaque requête devra être sauvegardée en lui donnant un nom.
4. Créez la requête SQL qui permet de créer une table (`create table...`) nommée `Telephone` qui comporte les attributs `numero` de type `nombre entier` et `prixForfait` de type `monétaire`.
 - a. Saisissez l'instruction dans votre éditeur externe ;
 - b. Copiez l'instruction dans Access ;
 - c. Exécutez l'instruction pour vérifier qu'elle est correcte. Dès qu'Access n'affiche aucune erreur, vous devriez voir apparaître la table `Telephone` dans les objets Access (il est possible que cela prenne beaucoup de temps...).
 - d. Fermez la fenêtre de création de la requête et enregistrez-la.
 - e. Double cliquez sur la requête que vous venez de créer. Que se passe-t-il ?
 - f. Affichez la table `Telephone` en mode création pour vérifier (cf. TP 3) puis fermez cette fenêtre.
5. Créez la requête SQL qui permet de créer la table `Contact` avec un attribut `num` qui est de type numérique auto, un attribut `nom` de type chaîne de 50 caractères maximum. Remarque : Contrairement à ce qui est annoncé sur le site MSDN, `UniqueIdentifier` n'est pas reconnu par Access 2007.
6. Créez la requête SQL qui permet de supprimer la table `Telephone`.

7. Modifiez la requête SQL de la question 4 pour que la saisie de l'attribut `numero` soit obligatoire (i.e. non NULL). La modification demandée correspond à une *contrainte d'intégrité*.

MODIFICATION DE LA STRUCTURE D'UNE BD

8. On souhaite altérer la table `Contact` de telle sorte que l'attribut `num` soit considéré comme clé primaire. Créez une nouvelle requête qui ajoute cette nouvelle contrainte d'intégrité sans modifier la requête de création de cette table.
9. Créez la requête SQL qui ajoute l'attribut `birthday` de type `Date` dans `Contact`.
10. Créez une requête qui ajoute l'attribut `numPers` de type `Entier` dans `Telephone`.
11. Créez la requête SQL qui définit `numPers` comme clé étrangère dans `Telephone` et qui référence `num` dans `Contact`. Pour vérifier que votre requête s'est correctement exécutée, affichez les relations entre les tables (cf. TP3, menu `Outils de base de données > Relations`).
12. Créez une requête qui supprime l'attribut `prixForfait` dans `Telephone`.

AJOUT DE DONNÉS

13. Créez les requêtes SQL qui permettent d'enregistrer les informations suivantes dans notre agenda téléphonique primaire :

Nom	Date de Naissance	Telephone
Secrétariat Info. Nice	02/10/1973	04 97 25 83 07
Secrétariat Info. Sophia		04 97 15 52 10

14. Affichez le contenu de la table `Telephone`. Que constatez-vous ?

MODIFICATION DES DONNÉES

15. Créez la requête qui modifie le type de données de l'attribut `numero` dans `Telephone` en chaîne de 10 caractères maximum. Modifiez à nouveau les numéros de téléphone pour qu'ils contiennent des 0.



Pour aller plus loin

16. Identifiez l'instruction SQL qui permet de préciser que la chaîne de caractères du téléphone doit obligatoirement commencer par un 0. Créez une requête SQL qui permet de définir cette nouvelle contrainte d'intégrité. Que constatez-vous ? Trouvez un autre moyen pour appliquer cette contrainte sous Access (cf. TP3).
17. Créez les requêtes SQL qui construisent le schéma relationnel gérant un grand magasin. Vous ajouterez au moins 2 enregistrements par table :
- ```

EMPLOYE (num_emp, nom_emp, prenom, adresse, salaire)
DEPARTEMENT (num_dep, nom_dep)
TRAVAILLE (num_emp#, num_dep#)
ARTICLE (num_art, nom_art, prix_vente)
FABRICANT (num_fab, nom_fab, adresse)
FABRIQUE (num_fab#, num_art#, prix_fournisseur)
VEND (num_dep#, num_art#)

```

## Sources pour ce TP

Base de données (Cours-TD-TP), Christiane MASSOUTIÉ - Max BUVRY, 2004-2005