Conception de Base de Données

Explication sur la Conception de Base de Données

Introduction

A la vue de plusieurs questions sur les bases de données, et surtout la conception du schéma de base.
En effet, bien souvent les problèmes de certaines personnes sont dus à des bases mal pensées à la base, et qui par la suite peuvent empêcher certaines possibilités qui peuvent être demandées dans la vie d'un projet.

Cet article n'est pas à but théorique mais va être développé à partir d'un exemple de façon à mieu comprendre l'importance de la réalisation du schéma de base, avant de s'attaquer au codage (que ce soit en ASP ou PHP).

Projet de Site Marchand

Je vais choisir un exemple simple que tout le monde pourra facilement comprendre, car tout le monde a déjà été confronté à un site marchand avec vente d'articles en ligne.

Définition du projet
Ce site va être très simple avec une demande d'identification de l'acheteur afin de mémoriser les articles déjà commandés pour chaque acheteur. On va faire un site simple avec une liste d'articles proposés très basique et des information basique pour les utilisateurs, on peut encore bien plus perfectionner la base.

Définition des besoins
On va simplement stocker les informations basiques concernant les **articles** (Nom, Prix Définition, Marque) et les informations nécessaires à l'envoie de ces produits à l'**acheteur** (Nom, Prénom, Adresse, Code Postal, Ville, Pays, Mot de passe). Donc d'aspect on peut déjà dire que les informations à stocker sont très simples et que la base ne va pas être très compliquée à monter.

NB : Dans tous les projets, il existe un cahier des charges. Il est soit implicite (cas d'un développement personnel), soit explicite (quand le projet est demandé par une autre personne). De même, il peut être écrit (cas des contrats entre client et vous) soit oral (cas où un supérieur vous demande de lui faire un site ou une page, ...). Quoi qu'il en soit, celui-ci devra toujours rester à l'esprit de la personne qui va réaliser le projet, car il sera le fil conducteur de celui-ci. Dans le cas contraire, on risque de déroger à la demande originale et de finir complètement hors sujet.

Donc, pour cet exemple, nous avons besoin :

- Une Liste des articles afin de présenter ceux-ci à l'acheteur potentiel
- Un stockage des informations sur le client pour son historique des achats
- Un enregistrement de l'utilisateur avec identification par mot de passe, afin que celui-ci puisse voir ce qu'il a déjà acheté (un peu comme Amazon)
- On voit ici que la même marque peut arriver plusieurs fois (donc on va scinder la table article en 2 avec une table article réelle et une table Marque, avec un lien entre les 2)
- De la même facon pour le Pays et la ville qui sont des données redondantes

On a donc la liste des besoins référencés, on va donc voir comment représenter les données et donc la base.

Données à stocker

Pour chaque table, on va lister les données à stocker dans cette base, table par table, en précisant le type de donnée qui va être choisi (non obligatoire).

ARTICLES

- id_article (Clé unique primaire et autoincrémentée)
- nom_article (Champ type texte)
- prix_article (Champ type numérique)
- definition_article (Champ type texte à taille variable - memo(sous Access) ou nvarchar(sous SQL Server))
- marque_article (Champ numérique en liaison avec la table MARQUE, correspondra à l'ID de la marque)

MARQUE

- id_marque (Clé unique primaire et autoincrémentée)
- nom_marque (Champ type texte)

ACHETEUR

- id_article (Clé unique primaire et autoincrémentée)
- nom_acheteur (Champ type texte)
- prenom_acheteur (Champ type texte)
- adresse_acheteur (Champ type texte)
- code_postal_acheteur (Champ type texte)
- ville_acheteur (Champ numérique en liaison avec la table VILLE, correspondra à l'ID de la ville)
- pays_acheteur (Champ numérique en liaison avec la table PAYS, correspondra à l'ID du pays)
- password_acheteur (Champ type texte)

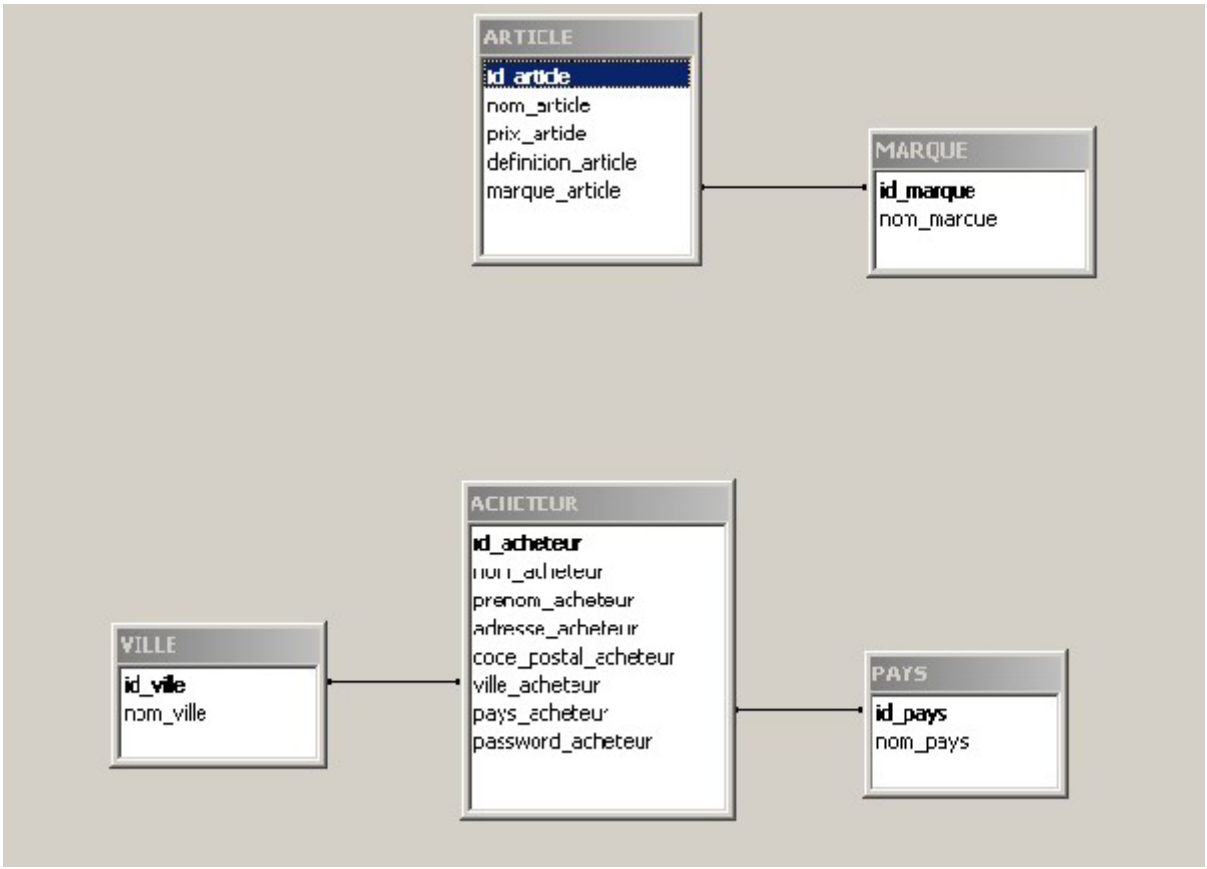
VILLE

- id_ville (Clé unique primaire et autoincrémentée)
- nom_ville (Champ type texte)

PAYS

- id_pays (Clé unique primaire et autoincrémentée)
- nom_pays (Champ type texte)

On voit ici le schéma avec les relations entre les tables :



Donc la nous avons une base qui nous permet d'avoir la liste des produits et la liste des utilisateurs en utilisant des jointures simples entre 2 tables (ARTICLE-MARQUE) de la même façon pour la table ACHETEUR.
Ainsi pour avoir les informations complètes d'un **article**, il suffit de faire une requette du type :

SELECT * FROM MARQUE m, ARTICLE a WHERE a.marque_article=m.id

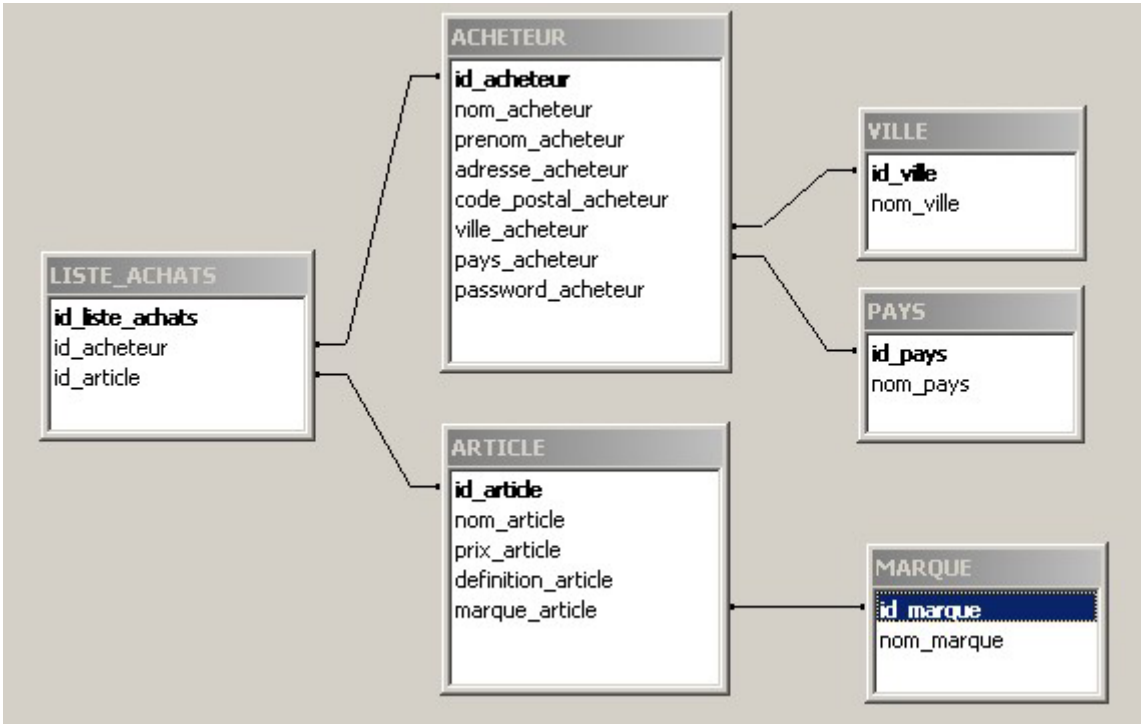
Je vous laisse imaginer la même pour les acheteurs (pour les requettes, voir ici)

Il nous manque maintenant un système afin de pouvoir avoir la liste des produits achetés par les visiteurs.
Pour ça il va falloir passer par une table intermédiaire qui va permettre d'avoir la liste des utilisateurs ayant acheté des produits. Cette table va donc contenir trois champs numériques simples.

LISTE_ACHATS

- id_liste_achats (Clé unique primaire et autoincrémentée)
- id_acheteur (Champ numérique en liaison avec la table ACHETEUR, correspondra à l'ID de l'acheteur)
- id_article (Champ numérique en liaison avec la table ARTICLE, correspondra à l'ID de l'article)

On a donc le schéma final avec les relations entre les tables :



Vous vous demandez certainement pourquoi mettre une clé unique dans cette table de liaison, c'est simplement pour la maintenance de cette table afin qu'il 'y ai jamais de doublons, car les doublons sont toujours problématiques dans les bases de données. Ainsi si on veut modifier l'id de l'utilisateur ou du produit sur un achat xxx, ça sera simple.

Téléchargement de l'exemple

Le téléchargement se fait ici :

- http://fromelard.free.fr/Scripts/Conception_Base/

Conclusion

J'espère que cette petite explication vous aura aidé dans la réalisation de vos projets.
Toutefois, je vous conseille de commencer par des bases simples.
Cette explication est applicable sur tous les types de base de données relationnelle (SGBDR), que ce soit Access ou MySQL ou encore ORACLE, c'est un principe général et on peut une fois la base comprise gérer des bases avec énormément de tables (pour info mon futur produit en cours de développement fait 26 Tables, mais ça sera un autre article :)))

F___