# Systèmes de gestion de bases de données

Présentation du cours

L’objectif de ce cours est d’assimiler, en 13 séances, les bases pour manipuler le logiciel Microsoft Access.

Ce logiciel est considéré comme un Sgbd, c’est-à-dire un outil de stockage de l’information, et permettant l’extraction d’informations grâce au langage Sql.

Ce cours se veut à la fois pratique et théorique ; fait appel aux connaissances suivantes : anglais, calcul matriciel, logique et ensembles, statistiques descriptives, notions de programmation, bureautique.

Plusieurs devoirs sont à rendre au fil des séances, avec un examen final.

Sommaire

1. qu’est ce qu’un SGBD ?
	1. introduction langage SQL
	2. les outils du marché
	3. les sgbd au service de la géographie
	4. présentation de microsoft Access : l’interface et ses onglets

2. concepts de base

1. tables
2. champs
3. requêtes
4. modules
5. macros
6. formulaire
7. la méthodologie Merise, le MCD et le MLD
	1. la syntaxe SQL
8. la grammaire du SQL
9. syntaxe SQL des principales requêtes
10. les jointures
11. mots-clefs et fonctions principales
	* 1. exemples et exercices
		2. résumé des cours 1 à 13
			1. qu’est ce qu’un SGBD ?

C’est un programme informatique, composé d’un outil de stockage des données, et des outils utilisés pour interroger ces données. Un SGBD est dit **relationnel** quand on peut relier les tables entre elles par des liens logiques.

1. le langage SQL

L’acronyme signifie « short query language », c’est la syntaxe utilisée universellement pour interroger les bases de données.

Cette syntaxe est formalisée, et peut se résumer à un pseudo-anglais ayant un lexique de mots-clefs. Ce lexique varie peu suivant les outils.

1. les outils du marché
	* + propriétaires : Oracle , Access, Sybase, etc
		+ open-source : mySql, postgreSQL , etc

Certains outils proposent des modules dits « spatiaux » (Oracle spatial, PostGis) dédiés au traitement de l’information géographique.

1. les sgbd au service de la géographie

Les bases de données peuvent se coupler aux logiciels cartographiques, voire eux mêmes être capables d’ interpréter des données géométriques ; les Sgbd se retrouvent donc aux cœur des SIG, pour le stockage et l’interrogation et la production de données, mais aussi pour l’analyse spatiale et la production de cartes.

1. Microsoft Access : présentation

A l’ouverture, créer un fichier vide de type .mdb (Microsoft data base), nommez-le et sauvegardez-le à un emplacement de votre choix; vous importerez ensuite vos tables dans ce fichier mdb ; Access peut importer beaucoup de types de fichiers, txt, xls, dbf, etc.

L’interface générale contient plusieurs onglets :

l’onglet **Tables** stocke vos objets tables

l’onglet **Requêtes** stocke vos requêtes de sélection, de mise à jour, de création de tables, les requêtes croisées et les requêtes de suppression de données, et les requêtes d’ajout de deux tables.

l’onglet **Formulaire** stocke vos formulaires, càd des interfaces que vous créez pour la saisie ou la consultation de vos données. Ils sont personnalisables, et permettent d’appeler des images, ou des contrôles activeX dans des fenêtres dédiées.

Les onglets **états** et **pages** ne seront pas abordés dans ce cours et ne sont pas indispensable pour nos besoins.

L’onglet **Macros** stocke vos scripts d’automatisation des tâches, à rédiger en langage Vba.

L’onglet **Modules** stocke des éléments de scripts et des fonctions, rédigés en Vba ;

des formules de mise en classe de séries continues, des fonctions de traitement de chaînes de caractères, etc

2. Outils et concepts de base

1. Tables

Chaque objet de cet onglet est une table, c’est à dire une **matrice n lignes X p colonnes** *(nxp).* Contrairement à Excel, le nombre d’enregistrements n’est pas limité.

Les n lignes sont appelés des **enregistrements**.

Les p colonnes sont des **champs**.

1. Champs

Ils ont un nom < ou = à 8 caractères en général, de préférence sans accentuation et sans ponctuation à part le caractère « \_ » .

La liste de tous les champs de toutes les tables est un document à produire pour toute conduite de projet, et se nomme généralement « dictionnaire des données ».

Généralement, les tables contiennent des données **identifiantes** (adresses, noms, codes,…), des données **attributaires** (des valeurs ou du texte), et de la géométrie dans le cas des SIG (mémo, objet OLE,…)

Une même table contient des champs de différents types, dont voici les principaux :

**texte** : stocke des caractères alphanumériques, du texte, des libellés, ou des variables qualitatives. Longueur max = 255 caractères.

**mémo**: idem, mais peut recueillir des chaînes de caractères plus longues (la géométrie d’objets par exemple.

**numérique/entier** : stocke des entiers naturels N uniquement, comme des variables quantitatives discrètes, des comptages, des identifiants.

**numérique/réel** : stocke des réels R ; séries quantitatives continues, taux, indicatrices …

**date/heure** : beaucoup de formats

**monétaire**

**clé primaire,** ou **identifiant**, ou **index**: champ texte ou numérique/entier stockant un identifiant unique, cad que les valeurs de ce champ sont **distinctes** **deux à deux**. Il n’y a pas de doublons dans ce champ. Généralement, ils s’agit de codes ; codes client, codes administratifs, clefs composées, numéro de commande, etc.

Ces champs particuliers sont très importants car ils permettent d’effectuer les liaisons entre tables.

Access propose une fonction de clé automatique, NumeroAuto.

1. Requêtes

Vous pouvez manipuler vos tables pour en les interroger simplement, pour extraire de l’information plus élaborée, pour les modifier, ou même pour les effacer.

Access possède un assistant à la composition de requêtes qui vous « traduira » votre opération en instruction SQL.(🡪 onglet « Requêtes », puis « Nouveau », puis « Mode création ». Il est cependant important d’aller vérifier soi-même le code généré.

L’avantage du stockage de requêtes est d’automatiser la production de données. L’autre avantage est de disposer de nouveaux objets assimilables à des tables dans vops futurs traitements, mais dynamiques, et très légers.

Access propose, dans l’onglet Requêtes, quelques requêtes assistées :

**Sélection** :

sélectionner un sous-ensemble de données issues des tables. Renvoie une nouvelle table stockée dans l’onglet requête.

**mise à jour** :

« remplit » un champ avec des valeurs que vous saisissez ou bien qui proviennent d’une autre table/requête.

**création** de table:

« transforme » le résultat d’une sélection en un nouvel objet table

requêtes **croisées** :

renvoie un tableau croisé de variables qualitatives ou quantitatives discrètes

requêtes de **suppression** de données

**ajout** de deux tables

requêtes **paramétrées**

1. Modules

C’est une fonction rédigée en VBA.

Exemple à venir MJ, module de mise en classe.

1. Macros

Suite d’instructions visant à automatiser des taches.

scripts en VBA, la fonction DoRunSQL est très utile.

1. formulaire

Exemple à venir NB

1. la méthodologie MERISE

On peut analyser la structure d’une base grâce à certains outils, en la modélisant, notamment avec la méthodologie MERISE.

« Cette méthode a eu comme objectif premier de jeter un pont entre les besoins des utilisateurs et les solutions des informaticiens. Certes sa finalité est quand même de faciliter la conception des projets informatiques en permettant d'analyser et de formaliser très tôt les « besoins » des utilisateurs. » *wikipedia*

- Le MCD

Le Modèle Conceptuel de Données est la formalisation de la structure et de la signification des informations décrivant des objets et des associations.

Le MCD comporte les concepts basiques suivants, à représenter sous forme de diagramme composé de boites contenant des listes de champs ; les champs **clefs primaires** et **les clefs étrangères** sont signalés respectivement en gras et souligné, et en gras.

**Entité** : table

**Relation** : modélisation d'une association entre deux ou plusieurs entités ;appartient à, possède, vend, achète, réside,…

**Cardinalités** : modélisation des participations mini et maxi d'une entité à une relation ;

<1,n> , <1,1> , <n,m>, …

**Propriétés** : modélisation des informations descriptives rattachées à une entité ou une relation

**Identifiant**: modélisation des propriétés contribuant à la détermination unique d'une occurrence d'un entité.

- Le MLD

Le MLD est la  « traduction technique » du MCD

Access gère le MLD dans l’option « Relations » interface dans laquelle vous précisez les liens entre tables.

NB

* 1. La syntaxe SQL

La formule pour générer une instruction SQL est toujours la même, et le langage possède une grammaire très simple.

1. La grammaire du SQL

pour créer vos instructions, vous assemblerez toujours, dans l’ordre, les éléments suivants :

Une instruction de base obligatoire, définissant le type d’opération (SELECT, INSERT INTO, DELETE…)

Vient ensuite la liste des champs que vous voulez restituer : champs d’origine, champs calculés ou des fonctions d’agrégat, ou bien la totalité des champs (caractère « \* »).

L’option AS permet de donner un nouveau nom aux champs calculés.

Un champ est nommé <nomtable.nomchamp> dans votre formule.

L’instruction obligatoire FROM, après laquelle vous précisez les tables dont sont issues les champs à extraire

Peuvent suivre une ou plusieurs clauses facultatives :

Clause de sélection logique WHERE ( =, <,>,<>,etc)

Clause de jointure : précision du lien logique entre les tables ( WHERE, INNER JOIN ON, LEFT JOIN ON, RIGHT JOIN ON)

Clause d’agrégat GROUP BY

Clause de restriction de la fonction d’agrégat HAVING, remplaçant la clause « WHERE » si la requête contient un GROUP BY

La formule générale d’une requête SQL se résume donc à ceci :

Type d’opération, liste de champs à restituer, liste des tables à requêter, clauses facultatives.

Ou encore : TypeRequête, <champs> FROM <tables>, critères et clauses.

1. Syntaxe SQL des principales requêtes

Requête de suppression de table :

DROP TABLE <NomTable>

Requête de suppression de champs ou d’enregistrements :

DELETE <NomTableSource.\*> FROM <NomTable> WHERE <critère>

NomTableSource.\* désigne la table dans de laquelle s'effectue la suppression dans le cas d'une jointure avec une autre table.

Requête de Mise à jour à partir d’une autre table:

INSERT INTO <NomTableDestination> SELECT <Champ1>, <Champ2>….FROM <NomTableSource> WHERE <condition>

Pour que l'insertion soit valide il faut que les champs de la table source aient le même type et la même longueur que ceux de la table destination.

Mise à jour d’enregistrements à l’aide d’une valeur choisie par l’utilisateur :

UPDATE <NomTable> SET <Nouvelle\_valeur> WHERE critère

Requête de sélection :

SELECT <Champ>, <Champ2>,...,... FROM <Relation> WHERE <Condition>

A noter que l’on peut compléter l’instruction select :

SELECT ALL (par défaut)

SELECT DISTINCT sélectionne sans doublons.

### Création d’une table :

### CREATE TABLE NomTable (<Champ1 Type (taille)> CONSTRAINT <Index>, <Champ2 Type (taille)> CONSTRAINT <Index2>)...CONSTRAINT <Index\_multichamp>

1. Les jointures

Soient une table A et une table B contenant chacune deux champs que l’on peut apparier.

Les jointures seront de 3 types, correspondant aux 3 résultats obtenus en intersectant les 2 ensembles A et B.

La jointure se déclare après la clause FROM <tables>.

LEFT JOIN : tous les enregistrements de A et seulement ceux de B pour lesquels le critère d’appariement est égal

RIGHT JOIN : : tous les enregistrements de B et seulement ceux de A pour lesquels le critère d’appariement est égal

INNER JOIN : tous les enregistrements pour lesquels la clé d’appariement est égale.

1. mots-clé et fonctions principales

instructions SQL, type de requête :

sélectionner : **SELECT, SELECT ALL**

sélectionner sans doublons : **SELECT DISTINCT**

mettre à jour des valeur : **UPDATE**

ajouter des enregistrements à une table depuis une autre table: **INSERT INTO**

effacer une table : **DROP**

effacer des enregistrements : **DELETE**

clauses :

indiquer les tables desquelles sont issues les données : **FROM** <liste des champs>

renommer un champ/donner un nouveau nom à un champ : <ancien> **AS** <nouveau>

regrouper/agréger les données : **GROUP BY** <champs>

clause de restriction pour l’agrégat : **HAVING**

clause de sélection : **WHERE**

jointure : … **JOIN** <Table> **ON** (<Champ1> = <Champ2>)

fonctions d’agrégat ou expressions de regroupement :

moyenne : **AVG**

écart-type : **STDEVP**

maximum, minimum :**MAX, MIN**

premier, dernier : **FIRST, LAST**

compte : **COUNT**

somme : **SUM**

opérateurs de la clause **WHERE**(non-exhaustif):

et : **AND**

ou inclusif: **OR**

ou exclusif : **XOR**

inférieur à :

supérieur à :

tri croissant : **ORDER** **BY** < champs> **ASC**

tri décroissant : **ORDER** **BY** < champs> **DESC**

équivalence : **EQV**

est vide : **IS NULL**

n’est pas vide : **IS NOT NULL**

entre : **BETWEEN**

Access propose, notamment grâce au générateur d’expressions de l’assistant requêtes, un vaste choix de fonctions de traitement de chaînes de caractères, et d’opérations logiques et mathématiques.

Dans d’autres systèmes, il existe des clauses WHERE topographiques, (intersect, inside,..), ainsi que des fonctions topographiques (distance, surface, centroide,..).

* 1. exemples et exercices

Exemple1 :

soit une table A n clients x p champs, avec :

p1 = identifiant client

p2 = CA réalisé

p3 = nb de commandes

p4 =code zone géographique

- restituer la table A à l’identique s’écrira :

- quelle est la clé ? qualifiez les autres champs :

- restituer une table renvoyant le CA moyen par commande et la dispersion de la série :

- restituer une table renvoyant le CA moyen par client et la dispersion de la série :

* + - restituer une table renvoyant le nb de clients, le CA moyen et total, le nb de commandes moyen et total, par zone, la table de résultat sera triée par le nb de clients décroissant :

- Idem que la précédente, mais que pour les secteurs « ouest » et « sud-est »

Exemple 2 :

on dispose d’une nouvelle table d’informations sur nos clients, la table B, contenant l’identifiant client et leurs adresses postales . Les données de cette table contiennent des erreurs, certains clients sont en double, et certaines adresses sont manquantes.

- comment dé-doublonner la table B ? (deux manières)

- comment incorporer les données de B dans A ? quel est le pré-requis avant cette manipulation ?

 - dans les tables A et B, sélectionner les clients dont l’adresse est renseignée.

Exemple 3 :

Rédiger des mini MLD (3 ou 4 tables) orientés vers la gestion marketing/produits.

rédiger les listes de tables et champs comme ci-dessous en évitant la redondance de l’information

imaginer quelques tables sur les thèmes suivants *: librairie, vente par correspondance, négoce de matières premières, devoir commun.*

Exemple d’une  librairie

*livres* (<**code livre**, **code\_auteur, code\_editeur,code\_genre,** titre**,** prix**,** date\_sortie**,** stock\_initial**,** date\_debut, date\_fin >)

*auteurs* (<**code auteur**, nom, prenom, age, **code pays**>)

*éditeurs* (<**code éditeur**, nom, adresse, fax, **code pays**>)

*tickets (<***code\_ticket, code\_livre**, quantite, date>)

*tables d’index (genres, pays)*

- caractérisez ces tables et ces champs (clés primaires, clefs étrangères, attributs et leurs types).

- Comment composeriez-vous le code livre ? pourquoi <code ticket de caisse> n’est pas une clef unique ?

- à partir de cet exemple , tracer un diagramme des entités-relations dans powerpoint ou autre.

- Quelles informations pourriez-vous extraire d’un tel système ? rédiger les requêtes correspondantes, et les représenter et les nommer sur le diagramme dans une deuxième partie du dessin.

- Ces 4 tables permettent une analyse des ventes et une gestion de stock ; quels autres aspects aurait-on pu incorporer dans cet outil de gestion ? quelles tables et champs rajouteriez-vous ?

- une fois que les tables, les relations et les requêtes pré-enregistrées sont formalisées, quels outils pourrait-on imaginer avec les autres outils d’Access ?