



Faculté de Sciences Économiques et de Gestion

Rappels de Visual Basic Connexion aux bases de données

Maîtrise de Sciences Économiques

Année 2000-2001

Jérôme Darmont

jerome.darmont@univ-lyon2.fr

Plan du cours

- I. Objectifs
- II. Rappels de Visual Basic
- III. Connexion VB-Access
- IV. Connexion Excel-Access

Plan du cours



I. Objectifs

II. Rappels de Visual Basic

III. Connexion VB-Access

IV. Connexion Excel-Access

Objectifs de l'année

- **Réaliser un projet informatique** en *Visual Basic* avec accès à une base de données *Access*
- **Organisation :**
 - Par groupes de 2
 - 5 séances de TD classiques
 - 5 séances consacrées uniquement au projet
- **Évaluation :**
 - Démo
 - Rapport écrit

Règles du jeu

- Le projet doit porter sur un *cas réel* concernant *personnellement* au moins un des étudiants
- Choix de sujet *motivé* à rendre le 20 octobre
- **Base de données :**
 - 5 tables minimum
 - 1 relation 1-N et 1 relation M-N minimum
 - Suffisamment de données pour effectuer des tests
 - 10 requêtes minimum (au moins 3 jointures, 3 groupements, 2 restrictions multicritères, 2 requêtes paramétrées)

Règles du jeu

- **Application Visual Basic :** Interface qui doit permettre de
 - Consulter
 - Traiter
 - Ajouter les données de la BD Access
 - Supprimer
 - Mettre à jour

Règles du jeu

- **Rapport écrit :**

- Présentation du sujet / Motivation
- Analyse, Schéma E/A de la base de données
- Organisation de l'application VB (feuilles, modules)
- Conclusion, Problèmes rencontrés
- Annexe : Code VB

Plan du cours

I. Objectifs



II. Rappels de Visual Basic

III. Connexion VB-Access

IV. Connexion Excel-Access

Qu'est-ce qu'une application VB ?

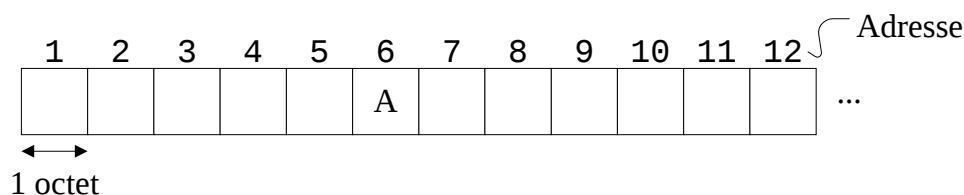
- Un ensemble de *procédures* (petits morceaux de programme)
- Une interface graphique composée de *contrôles* (zones de texte, boutons, listes, etc.)
- Une procédure est exécutée lorsqu'un *événement* survient (ex. si l'utilisateur clique sur un bouton)

⇒ On parle de programmation *visuelle* et *événementielle*

Stockage des données

- Un programme manipule des données qui doivent être stockées en mémoire centrale (mémoire vive, RAM en anglais)

- **Structure logique de la mémoire :**



- La taille mémoire se compte en méga-octets (1 Mo = 1024 Ko, 1 Ko = 1024 octets)

Stockage des données

- Manipulation d'adresses peu pratique $\Rightarrow$ association d'un nom à l'adresse d'une donnée
- Déclarer une variable ou une constante $\approx$ allouer une adresse à une donnée et lui associer un nom
- **Déclaration de variable VB :**
`dim nom_var as type`
- **Déclaration de constante VB :**
`const nom_const = valeur_const`

Types de données

- **Entiers :**
Byte (1 octet, [0, 255])
Integer (2 octets, [-32768, +32767])
Long (4 octets, [-2147483648, +2147483647])
- **Réels :**
Single (4 octets, [-3,402.10³⁸, + 3,402.10³⁸])
Long (8 octets, [- 1,797.10³⁰⁸, +1,797.10³⁰⁸])
Currency (8 octets, [-922337203685477,5808, +922337203685477,5807])

Types de données

- **Chaîne de caractères :**
String (jusqu'à environ 2 milliards d'octets/caractères)
- **Date :** Date (8 octets, [01/01/100, 31/12/9999])
- **Booléen :** Boolean (2 octets, True ou False)
- **Contrôle :** Object (4 octets, adresse mémoire du contrôle)
- **Variable :** Variant (16+ octets)
Le type change selon le contexte (à utiliser avec modération).

Types de données

- **Tableaux :** Dim *nom_tab(taille)* as *type*
- **Type structuré défini par l'utilisateur :**
Type *nom_type*
 champ1 as *type*
 champ2 as *type*
 ...
End Type

Ex. Type structuré Personne constitué de Nom (chaîne), Prénom (chaîne), Âge (octet)...

Exemples de déclarations

```
Const PI=3.1416
Dim i as Integer
Dim ch1, ch2, ch3 as String
Dim tab_de_reels(100) as Single
Dim matrice(10,10) as Double
Type Personne
    nom as String
    prenom as String
    age as Byte
End Type
Dim une_personne as Personne
```

Éléments de base du langage

- **Affectation : =**

```
Ex. i = 1
    ch1 = "Coucou !"
    tab_de_reels(5) = 3.14
    matrice(i,4) = PI*2
    une_personne.prenom = "Jérôme"
```

- **Commentaire : '**

```
Ex. ' Ceci est une ligne de commentaire
```

Éléments de base du langage

- **Conversions de type :**

Fonction / Convertit une expression en

CByte / Byte

CInt / Integer

CLng / Long

CSng / Single

CDb1 / Double

CCur / Currency

CStr / String

CDate / Date

CBool / Boolean

CVar / Variant

Ex. i = CInt(PI)

' i=3

```
msg_affiche = CStr(i)
```

Éléments de base du langage

- **Opérateurs de comparaison :**

Égal =

Différent <>

Inférieur <

Inférieur ou égal $\leq$

Supérieur >

Supérieur ou égal $\geq$

- **Opérateurs logiques :**

- *Ou* : or . La condition (*cond1* or *cond2*) est vraie si *cond1* est vraie ou *cond2* est vraie.

- *Et* : and . La condition (*cond1* and *cond2*) est vraie si *cond1* et *cond2* sont vraies.

- *Non* : not . La condition (not *cond1*) est vraie si *cond1* est fausse.

Éléments de base du langage

- **Composition des opérateurs logiques :**

not(*cond1* **or** *cond2*)
⇔ (**not** *cond1*) **and** (**not** *cond2*)

not(*cond1* **and** *cond2*)
⇔ (**not** *cond1*) **or** (**not** *cond2*)

Ex. **not**(*a*=1 **or** *b*>10)
⇔ (**not**(*a*=1)) **and** (**not**(*b*>10))
⇔ (*a*<>1) **and** (*b*<=10)

Structures de contrôle

- **Test :** *If condition then*
 ' Instructions si condition vraie
 Else
 ' Instructions si condition fausse
 End If

Ex. Calcul du maximum entre deux nombres

```
If n1<n2 then
    max = n2
Else
    max = n1
End If
```

NB : La clause else est optionnelle dans le cas général.

Structures de contrôle

- **Sélection :** `Select case variable`
 `Case valeur`
 `' Instructions`
 `...`
 `Case Else`
 `' Traitement par défaut (optionnel)`
 `End Select`

Ex. `Select case num_mois`
 `Case 1`
 `nom_mois = "janvier"`
 `Case 3, 4, 5`
 `saison = "printemps"`
 `Case Else`
 `message = "cas non prévu"`
 `End Select`

Structures de contrôle

- **Test « court » :** `Iif condition, si_vrai, si_faux`

Ex. Calcul du maximum entre deux nombres

`Iif n1<n2, max = n2, max = n1`

- **Choix d'index :** `var = Choose (index, expr1, expr2...)`

Ex. Calcul de taux

`taux = Choose (i, 0.05, 0.07, 0.1, 0.15)`

Si $i=1$ alors $\text{taux} = 5 \%$

Si $i=2$ alors $\text{taux} = 7 \%$

Si $i=3$ alors $\text{taux} = 10 \%$

Si $i=4$ alors $\text{taux} = 15 \%$

Structures de contrôle

- **Boucle « pour » :**

```
For var=min to max step pas
    ' Instructions
Next var
```

Ex. Calcul de factorielle ($10! = 1*2*3*4*5*6*7*8*9*10$)

```
fact = 1
For i = 1 to 10      ' step optionnel si 1
    fact = fact*i
Next i
```

Ex. Initialisation à rebours d'un tableau

```
For i = 100 to 1 step -1
    tab_de_reels(i) = i*PI
Next i
```

Structures de contrôle

- **Boucle « pour chaque élément de tableau » :**

```
For each var in tableau
    ' Instructions
Next var
```

Ex.

```
Dim nom_mois(12) as String
```

```
Dim mois as String
nom_mois(1) = "janvier"
...
```

```
For each mois in nom_mois
    ' Traitement
Next
```

Structures de contrôle

- **Boucle « tant que » :** While *condition*
 ' Instructions
 Wend

Ex. Arrêt du calcul de factorielle

```
fact = 1
i = 1
While (i<=10) and (fact<100)
    fact = fact*i
    i = i+1
Wend
```

NB : Le test sur la condition étant placé en début de boucle, on peut ne pas « entrer » dans la boucle.

Structures de contrôle

- **Boucle « répéter jusqu'à » :**

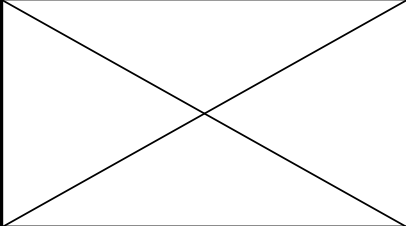
```
Do
    ' Instructions
Loop until condition
```

Ex. Arrêt du calcul de factorielle

```
fact = 1
i = 1
Do
    fact = fact*i
    i = i+1
Loop until (i>10) or (fact>=100)
```

NB : Le test sur la condition étant placé en fin de boucle, les instructions sont au moins exécutées une fois.

Comment choisir la « bonne » boucle ?

	Nombre d'itérations connu	Nombre d'itérations inconnu
La boucle doit être exécutée au moins une fois	FOR NEXT	DO LOOP UNTIL
La boucle peut ne pas être exécutée		WHILE WEND

Structures imbriquées

Ex. 1 : Condition « complexe » pour calculer une remise

```
If bon_client then ' ⇔ bon_client=true
    If montant>1000 then
        taux_remise = 0.1 ' 10 %
    Else
        taux_remise = 0.05 ' 5 %
    End If
Else ' bon_client=false
    If montant>2000 then
        taux_remise = 0.05 ' 5 %
    Else
        taux_remise = 0.025 ' 2.5 %
    End If
End If
```

Structures imbriquées

Ex. 2 : Initialisation d'une matrice

```
For i = 1 to 10
    For j = 1 to 10
        matrice(i,j) = i*j
    Next j
Next i
```

Ex. 3 : Comptage d'une valeur dans un tableau

```
c = 0
For i = 1 to 100
    If tab_de_reels(i) = PI then c = c+1
Next i
```

Sous-programmes

- **Sous-programme :** Programme autonome dédié à une tâche précise, de préférence de taille réduite. Un sous-programme peut recevoir des paramètres.
- **Deux types de sous-programmes :**
 - *Procédures* : Sous-programmes proprement dits
 - *Fonctions* : Sous-programmes qui renvoient un résultat (ex. fonction de calcul du sinus d'un angle)

Sous-programmes

- **Structure d'une procédure :**

```
Public|Private Sub nom_proc (paramètres)  
    ' Déclarations  
    ' Instructions  
End Sub
```

- **Structure d'une fonction :**

```
Public|Private Function nom_fn ( ) as type  
    ' Déclarations  
    ' Instructions  
    nom_fn = val_retour  
End Function
```

- *Private* : Appel possible uniquement depuis la même feuille ou le même module / *Public* : depuis toute l'application

Sous-programmes

- **Définition des paramètres :** séparés par des virgules, en spécifiant le type de chacun

Ex. *poct* as Byte, *pch* as String, *preel* as Single

- **Modes de passage :**

- *Par valeur* (ByVal) : Si le paramètre est une variable existante, son contenu est recopié. Il ne sera pas modifié en sortie de sous-programme.

Mode
par
défaut

- *Par référence* (ByRef) : Si le paramètre est une variable existante, son adresse est utilisée. Il pourra être modifié en sortie de sous-programme.

Ex. ByVal *poct* as Byte, ByRef *preel* as Single

Sous-programmes

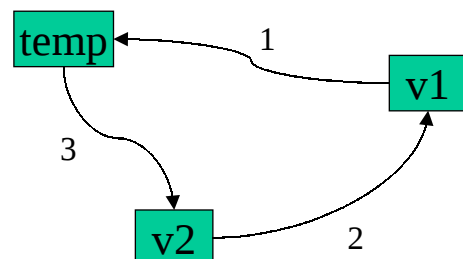
- **Exemple 1 : Calcul de maximum**

```
Public Function Max(ByVal n1 as Single,
ByVal n2 as Single) as Single
    If n1<n2 then
        Max = n2 ' Valeur de retour : n2
    Else
        Max = n1 ' Valeur de retour : n1
    End if
End Function
```

Sous-programmes

- **Exemple 2 : Échange de deux variables**

```
Public Sub Swap(v1 as String, v2 as String)
' v1 et v2 sont passées par référence
    Dim temp as String ' Variable temporaire
    temp = v1
    v1 = v2
    v2 = temp
End Sub
```



Sous-programmes

- **Appel de fonction** depuis un autre sous-programme :
Ex. `un_reel = Max (var1, 32767)`
- **Appel de procédure** depuis un autre sous-programme :
Ex. 1 : `Call Swap (chaine1, chaine2)`
Ex. 2 : `Swap chaine1, chaine2`
- **Fonctions intégrées** : Menu Aide / Sommaire de l'aide / Référence du langage / Fonctions

Portée des variables

- Variable déclarée dans un sous-programme ⇒ **variable locale au sous-programme** accessible uniquement dans ce sous-programme
 - Initialisée à chaque appel
 - Détruite en fin de sous programme
 - Sauf si déclarée en temps que **variable statique** (valeur conservée entre deux appels au sous-programme)
Ex. `Static compteur as Integer`

Portée des variables

- Variable déclarée dans la section *Général / Déclarations* d'une feuille ou d'un module ⇒ **variable locale à la feuille / au module** accessible par tous les sous-programmes de la feuille / du module, mais pas à l'extérieur
- Variable déclarée publique dans la section *Général / Déclarations* d'un module ⇒ **variable globale** accessible par toutes les feuilles et tous les modules de l'application
Ex. `Public Score as Integer`

Éléments visuels

- **Contrôle** : Objet prédéfini (type `Object`) au sens de la programmation orientée-objet
 - *Propriétés* : Définition de son aspect
 - *Événements* : Définition de son comportement
- **Contrôles courants** : Étiquette (*Label*), Zone de Texte (*TextBox*), Cadre (*Frame*), Bouton de commande (*CommandButton*), Case à cocher (*CheckBox*), Bouton d'option (*OptionButton*), Zone de liste (*ListBox*), Zone de liste modifiable (*ComboBox*)...

Éléments visuels

- **Quelques propriétés communes :**

- Name Nom interne de l'objet qui peut être utilisé dans un programme
- Appearance Apparence du contrôle (3D ou plat)
- BackColor Couleur du fond
- Caption Texte affiché à l'écran (sauf pour les champs de saisie ⇒ Text)
- Enabled Contrôle activé ou non (True/False)
- Font Police de caractères
- ForeColor Couleur de l'écriture
- TabIndex Ordre d'accès avec la touche TAB
- Visible Contrôle visible ou non (True/False)

Éléments visuels

- **Quelques événements commun :**

- Change Modification du contenu du contrôle
- Click Clic de souris sur le contrôle
- DoubleClick Double clic de souris sur le contrôle
- Drag... Glisser/déposer
- GotFocus Gain du focus
- Key... Événements provenant du clavier
- LostFocus Perte du focus
- Mouse... Événements provenant de la souris autres que le clic

Éléments visuels

- **Manipulation de contrôle par programme**

Ex. Procédure générique d’affichage d’un texte dans une étiquette (*label*).

```
Public Sub Etiq_Affiche (etiq as Object,  
    ByVal msg as String)  
    etiq.Caption = msg  
End Sub
```

Ex. d’appels : Etiq_Affiche label1, "coucou"
 Etiq_Affiche label2, message

Plan du cours

I. Objectifs

II. Rappels de Visual Basic

 **III. Connexion VB-Access**

IV. Connexion Excel-Access

Contrôle Donnée

- Un contrôle Donnée (*Data*) permet de connecter une feuille VB donnée à une table ou une requête d'une base de données Access.
- **Propriétés principales :**
 - *Connect* : Type de fichier base de données (Access)
 - *DatabaseName* : Nom du fichier .mdb contenant la base de données
 - *RecordSource* : Nom de la table/requête source
 - *ReadOnly* : Empêche la modification des données si mis à True (fortement conseillé pour les requêtes)

Contrôle Donnée

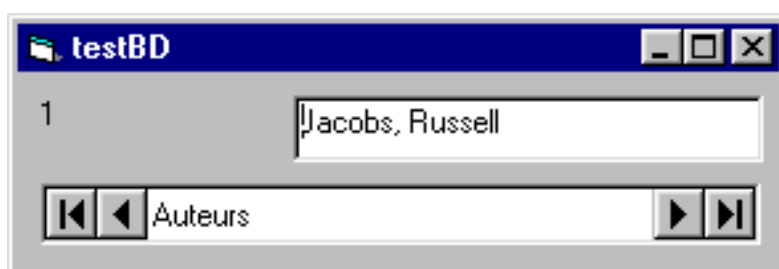
- **Affichage des données associées à un contrôle Donnée :**
 - Étiquettes (texte)
 - Zones de texte (texte modifiable)
 - Zones de liste et zones de listes modifiables
 - Images et dessins
 - Contrôles spécialisés (voir plus loin)
- **Propriétés utilisées :**
 - *DataSource* : Nom du contrôle Donnée à utiliser
 - *DataField* : Nom de l'attribut à afficher

Contrôle Donnée

- **Exemple :**

Soit la table Authors (Au_ID, Author, YearBorn) de la base de données biblio.mdb.

On souhaite afficher les numéros identifiants des auteurs et permettre la modification de leurs noms.



Contrôle Donnée

- Contrôle Donnée *Auteurs* :
 - *Connect* = Access
 - *DatabaseName* = E:\VB5\biblio.mdb
 - *RecordSource* = Authors
- Contrôle Étiquette *Numéro* :
 - *DataSource* = Auteurs
 - *DataField* = Au_ID
- Contrôle Zone de texte *Nom* :
 - *DataSource* = Auteurs
 - *DataField* = Author

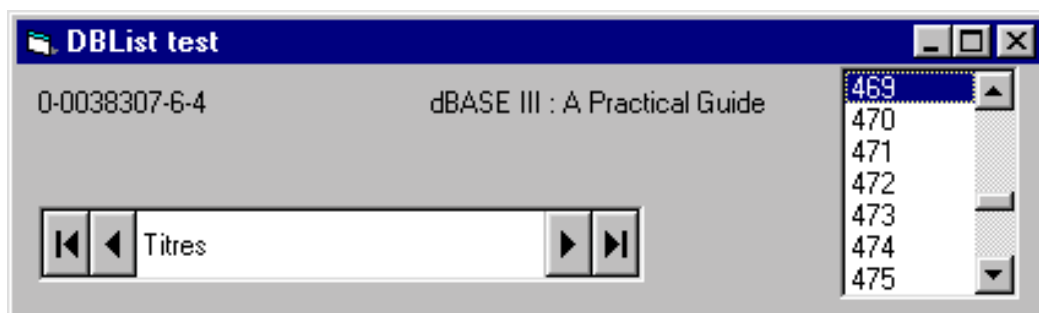
Contrôles Zones de liste BD

- Contrôles **Zone de liste BD** (*DBList*) et **Zone de liste BD modifiable** (*DBCombo*) $\approx$ Zones de liste (*ListBox*) et Zones de liste modifiables (*ComboBox*) standard
- Affichage d'informations provenant d'une base de données
- Permettent de travailler sur deux tables mises en relation (au sens Access du terme)
- **Différence *DBList* / *DBCombo*** : *DBCombo* $\Rightarrow$ choix dans la liste + saisie au clavier possible

Contrôles Zones de liste BD

- **Propriétés principales :**
 - *DataSource* : Contrôle Donnée primaire
 - *DataField* : Clé étrangère dans Donnée primaire
 - *RowSource* : Contrôle Donnée secondaire (liste)
 - *ListField* : Clé primaire dans Donnée secondaire
 - **Exemple :** Soient les tables
Titles (ISBN, Title, *PubID*...)
Publishers (PubID, Name, Address...).
- On souhaite, pour chaque titre, pouvoir sélectionner le numéro de l'éditeur (*publisher*) parmi la liste de tous les numéros d'éditeurs.

Contrôles Zones de liste BD



- Contrôle Donnée *Titres* :
 - *Connect* = Access
 - *DatabaseName* = E:\VB5\biblio.mdb
 - *RecordSource* = Titles

Contrôles Zones de liste BD

- Contrôle Donnée *Éditeurs* :
 - *Connect* = Access
 - *DatabaseName* = E:\VB5\biblio.mdb
 - *RecordSource* = Publishers
 - *Visible* = False
- Contrôle Étiquette *Numéro* :
 - *DataSource* = Titres
 - *DataField* = ISBN

Contrôles Zones de liste BD

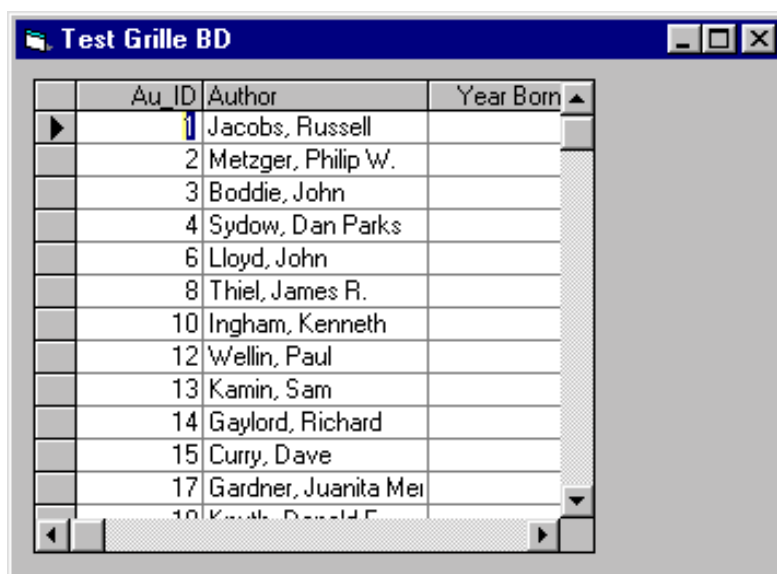
- Contrôle Étiquette *Titre* :
 - *DataSource* = Titres
 - *DataField* = Title
- Contrôle Zone de liste BD *Num Eds* :
 - *DataSource* = Titres
 - *DataField* = PubID
 - *RowSource* = Éditeurs
 - *ListField* = PubID

Contrôle Grille BD

- Contrôle **Grille BD** (*DBGrid*) $\approx$ Grille (*MSFlexGrid*)
- Affichage d'informations provenant d'une base de données
- Permet d'afficher sous forme tabulaire (façon grille Excel) et de modifier les données d'une table
- **Propriété principale** :
 - *DataSource* : Nom du contrôle Donnée à utiliser

Contrôle Grille BD

- **Exemple** : La table Authors



Au_ID	Author	Year Born
1	Jacobs, Russell	
2	Metzger, Philip W.	
3	Boddie, John	
4	Sydow, Dan Parks	
6	Lloyd, John	
8	Thiel, James R.	
10	Ingham, Kenneth	
12	Wellin, Paul	
13	Kamin, Sam	
14	Gaylord, Richard	
15	Curry, Dave	
17	Gardner, Juanita Mei	
18	K... ..	

Contrôle Grille BD

- Contrôle Donnée Auteurs :
 - *Connect* = Access
 - *DatabaseName* = E:\VB5\biblio.mdb
 - *RecordSource* = Authors
 - *Visible* = False
- Contrôle Grille BD Grille Auteurs :
 - *DataSource* = Auteurs

Assistant Feuilles de Données

- Permet, de manière interactive, de créer automatiquement des feuilles VB connectées à des bases de données
- Accès par le **menu Compléments**
Si l'Assistant Feuilles de données n'apparaît pas, il est possible de l'ajouter :
 - 1) Menu Compléments
 - 2) Gestionnaire de Compléments
 - 3) VB Data Form Wizard
 - 4) OK.

Assistant Feuilles de Données

- **Étapes de la création d'une feuille de données**
 - Sélection du type de base de données
 - Sélection du fichier contenant la base de données
 - Sélection de la présentation
 - Sélection de la ou des tables/requêtes sources
 - Sélection des attributs à afficher
 - Sélections des boutons de commande pré-programmés (Ajouter, Supprimer, Rafraîchir, Mettre à jour, Fermer)
 - Choix du nom de la feuille

Assistant Feuilles de Données

- **Appel d'une feuille de données :**

Dans l'événement *Click* d'un Bouton de commande ou *Form_Load* de la feuille principale d'un projet :

nom_feuille_données.Show

Manipulation de données par programme

- Utilisation de la propriété *RecordSet* (table) d'un contrôle Donnée et de ses « sous-propriétés » et méthodes (procédures)
- **Positionnement dans la table :**
 - Propriétés :

<i>BOF</i>	True si début de table
<i>EOF</i>	True si fin de table
<i>AbsolutePosition</i>	Position du tuple dans la table

Manipulation de données par programme

– Méthodes :

MoveFirst Déplacement en 1^{ère} position
MoveLast Déplacement en dernière position
MoveNext Déplacement à la position suivante
MovePrevious Déplacement à la position précédente

– Exemple :

```
Auteurs.RecordSet.MoveFirst  
While Not (Auteurs.RecordSet.EOF)  
  Auteurs.RecordSet.MoveNext  
Wend
```

Manipulation de données par programme

• **Accès aux données :**

Propriété *Fields*("attribut").*Value*

EX. Nom = Auteurs.RecordSet.Fields("Author").Value

• **Comptage des tuples :**

Propriété *RecordCount* (nombre de tuples accédés dans un objet *Recordset*)

Ex. Auteurs.RecordSet.MoveLast
 c = Auteurs.RecordSet.RecordCount

Manipulation de données par programme

- **Recherche dans la table :**

- Propriété :

Bookmark Signet (chaîne) pointant chaque enregistrement

Si le signet est mémorisé dans une position donnée, il est possible par la suite de « sauter » directement au tuple correspondant.

```
Ex.    Dim b as String
        b = Auteurs.RecordSet.Bookmark
        Auteurs.RecordSet.MoveLast
        Auteurs.RecordSet.Bookmark = b
        ' Retour à la position initiale
```

Manipulation de données par programme

- Méthodes :

FindFirst("condition_SQL")

Trouve le 1^{er} tuple correspondant à la condition

FindLast("condition_SQL")

Trouve le dernier correspondant à la condition

FindNext("condition_SQL")

Trouve le tuple suivant correspondant à la condition

FindPrevious("condition_SQL")

Trouve le tuple précédent correspondant à la condition

```
EX. Auteurs.RecordSet.FindFirst ("Author = 'Metzger'")
```

Manipulation de données par programme

- **Mises à jour :**

- Méthode *AddNew* (ajout de tuple en fin de table)

- Ex. `Auteurs.RecordSet.AddNew`

- Méthode *Delete* (suppression du tuple courant)

- Ex. `Auteurs.RecordSet.Delete`

NB : Les modifications de tuples sont automatiquement prises en compte si le contrôle Donnée n'est pas en lecture seule.

Plan du cours

I. Objectifs

II. Rappels de Visual Basic

III. Connexion VB-Access

 **IV. Connexion Excel-Access**

Importations de données externes

- **Principe** : Copie du résultat d'une requête dans une feuille Excel (utilisation de MS Query)
- **Marche à suivre** :
 - Menu Données / Données externes / Créer une req.
 - Choix de la source de données : Type de BD
 - Sélection de la base de données : Fichier BD
 - Sélection table(s)/requête(s) et attribut(s)
 - Filtrage des données (optionnel)
 - Tri des données (optionnel)
 - Envoi des données dans une feuille Excel



Importations de données externes

- **Exemple** :
 - Source de données : MS Access Database
 - Base de données : E:\VB5\biblio.mdb
 - Table : Authors
 - Attributs : Au_ID, Author, YearBorn
 - Filtrage : YearBorn n'est pas nul
 - Tri : 1^{ère} clé = Author, ordre croissant

Importations de données externes

	A	B	C	
1	Au_ID	Author	Year Born	
2	7358	Ageloff, Roy	1943	
3	611	Bard, Dick	1941	
4	8081	Bronzite, Michael	1936	
5	5650	Chou, George Tsu-Der	1939	
6	1410	Davis, Steve	1953	
7	8139	Gane, Chris	1938	
8	3915	Grommes, Bob	1957	
9				

Propriétés de la plage de données externes

- Accessibles par le bouton droit ou le menu Données / Données externes
- **Définition de la requête** : Ne pas modifier
- **Contrôle de l'actualisation** :
 - Actualisation en arrière-plan
 - Actualisation périodique (toutes les X minutes)
 - Actualisation à l'ouverture du fichier
- **Format et disposition des données** :
 - Afficher les noms d'attributs
 - Afficher les numéros de ligne ...

Actualisation / Modification de la requête

- Fonctions accessibles par le bouton droit ou le menu Données / Données externes
- **Actualisation** : Permet de prendre en compte les modifications éventuellement survenues dans la source de données Access (table/requête)
NB : Les modifications effectuées sur les données dans Excel ne sont pas prises en compte dans Access.
- **Modification de la requête MS Query** :
Changement des paramètres de la requête (table, champs, filtres, tri...)

Fonctions du menu Données

- **Trier** (fonction standard d'Excel)
- **Filtre** (fonction standard d'Excel)
- **Grille** : Accès aux données par formulaire
 - Navigation parmi les tuples (suivant/précédent)
 - Recherche par valeur d'attribut(s)
 - Ajout de tuple
 - Suppression de tuple
- **Sous-totaux** : Calcul de sous-totaux (somme, produit, compte, moyenne...) par groupes de données
Ex. Montant total de commandes pour une date donnée