RECHERCHE MULTI-CRITERES

par Charles A.

Date de publication : 01/10/2002

Dernière mise à jour : 01/09/2005

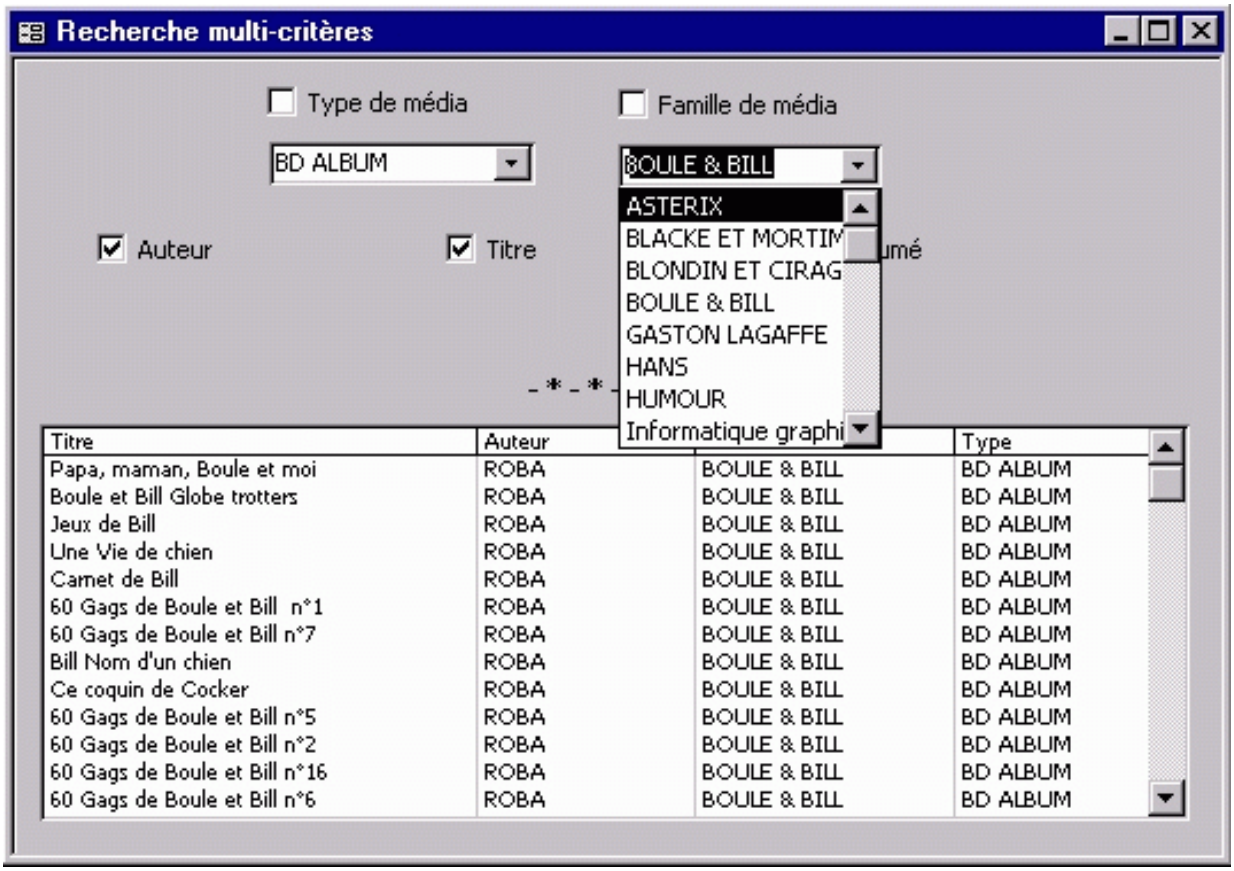
Réalisation pas à pas d'un formulaire de recherche dynamique en fonction de plusieurs critères

I - Accueil

Ce document vise à une meilleure compréhension du fonctionnement de l'application Microsoft ACCESS

Le projet exemple a été réalisé sous ACCESS 2000 avec pour OS Windows NT SP5

Voici une copie d'écran du résultat du tutoriel :



Vous pouvez télécharger la base access pour suivre en même temps le tutoriel.

II - Objectif

L'objectif de ce tutoriel est de permettre à un utilisateur de réaliser un formulaire de recherche multi-critères sur une table.

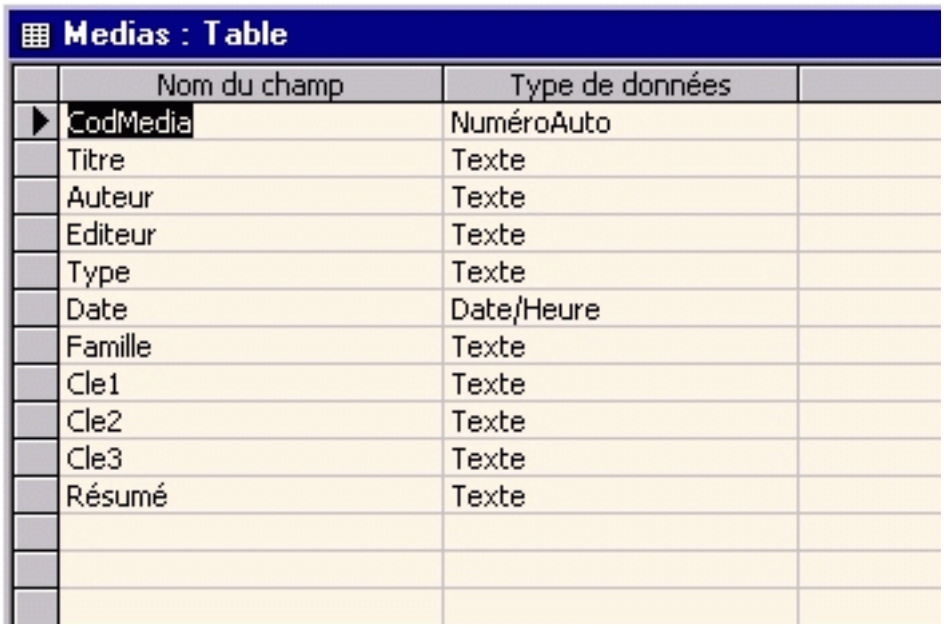
C'est à dire d'afficher une liste à partir de sélection de critères de recherche sur une table déterminée.

Les compétences requises sont de niveau débutant :	Les compétences acquises seront :
. connaissance de la structure d'une table	. principe de requête SQL
. notion de SQL	. codage VBA de l'interface utilisateur
. connaissance des formulaires	. codage VBA d'un SQL
. notion de code VBA	. compréhension des événements liés aux objets de données

III - Table du projet

Suite à une recherche sur le net, j'ai pu trouver une table contenant 100 enregistrements de médias

dont voici la structure.



Medias : Table

Nom du champ	Type de données
CodMedia	NuméroAuto
Titre	Texte
Auteur	Texte
Editeur	Texte
Type	Texte
Date	Date/Heure
Famille	Texte
Cle1	Texte
Cle2	Texte
Cle3	Texte
Résumé	Texte

et quelques données

CodMedia	Titre	Auteur	Editeur	Type
1	Manuel du développeur Access 2000	Rick DOBSON	Microsoft Pres:	Livre
2	VBA pour access 2000	Robert SMITH	Eyrolles	Livre
3	Guide du programmeur office 2000 visual basic	Microsoft	Microsoft Pres:	Livre
4	Visual Basic 6 Ressources d'experts	Rob THAYER	SSM	Livre
5	Grand Livre Access 2000	Spona HELMA	Micro applicatic	Livre
6	Le Platinium Access 2000	Ken GETZ	Sybex	Livre
7	Visual Basic 6 Atelier	John CLARK CRAIG	Microsoft Pres:	Livre
8	Visual Basic 6 les bases de données et SQL	Rémy LENTZNER	Eyrolles	Livre
9	Clickart	Photodex	Photodex	Livre
10	GoLive 5.0	Adobe	Adobe	Livre
11	Windows 98 Poche visuel	marangraphics	IDG Books	Livre de poche
12	Excel 2000 Poche visuel	marangraphics	IDG Books	Livre de poche
13	Word 2000 l'Essentiel	Jerry JOYCE	Microsoft Pres:	Livre de poche
14	Les Bijoux de la Castafiore	HERGE	CASTERMAN	BD ALBUM
15	L'Oreille cassée	HERGE	CASTERMAN	BD ALBUM
16	le Sceptre d'Ottokar	HERGE	CASTERMAN	BD ALBUM

L'objectif n'est pas ici d'améliorer tel ou tel type de données, mais d'effectuer une recherche performante et rapide sur cette table, au moyen d'un formulaire.

Ce formulaire ne sera pas dédié à la saisie mais à la consultation.

IV - Formulaire

Nous allons utiliser un formulaire indépendant, c'est à dire dont les contrôles ne sont pas liés directement à une source de données.

Dans un formulaire dépendant, un changement de valeur d'un contrôle dépendant implique un changement de valeur dans une table.

L'idée est de rechercher sur cinq critères :

*** Type de Média (BD, DVD ...)**

*** Famille (Humour, Informatique ...)**

*** Auteur**

*** Titre**

*** Résumé**

Nous ferons des recherches sur une combinaison des cinq éléments avec les spécifications suivantes :

* type et famille : **critère exact**, c'est à dire choix parmi une liste

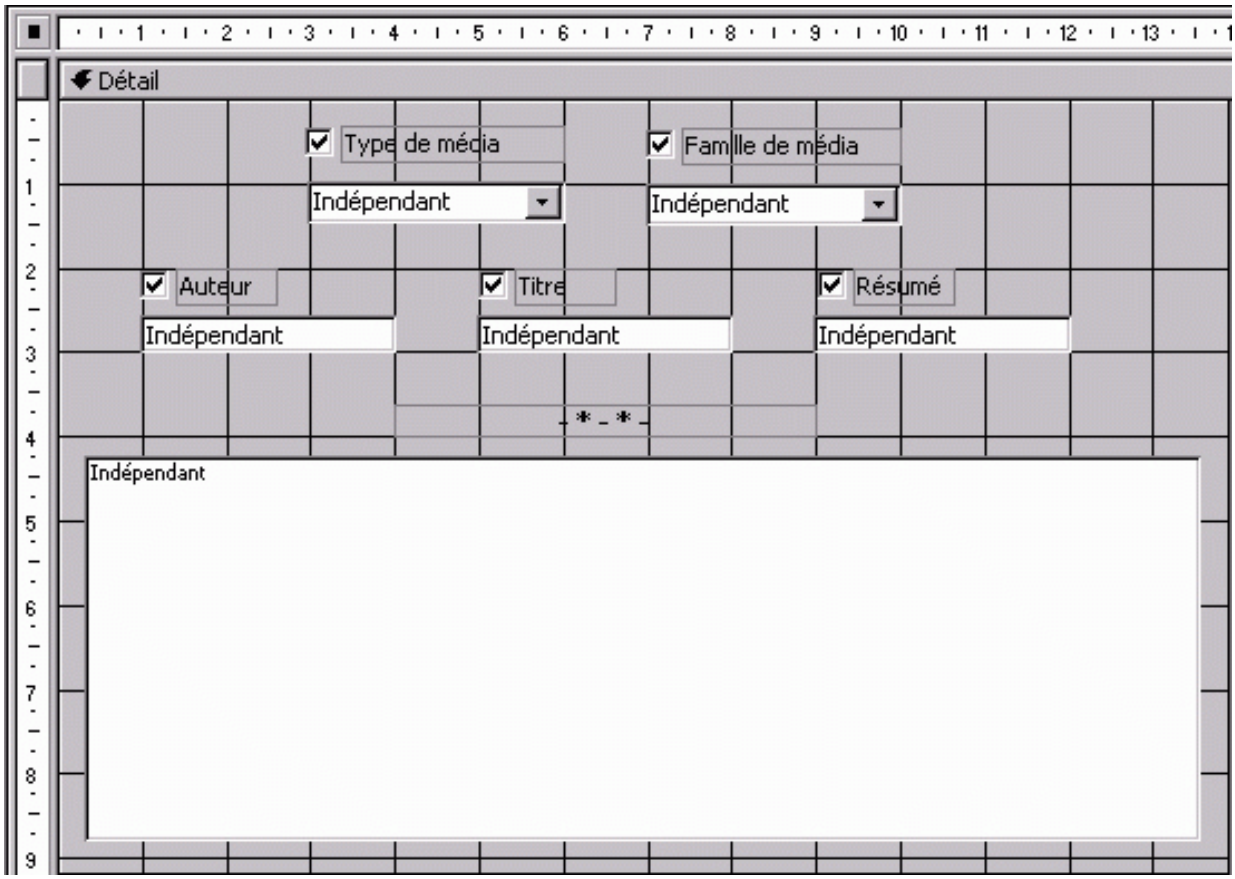
* Auteur, Titre et Résumé : **critère contenu**, c'est à dire que la réponse doit contenir le critère (par exemple : "HER" peut représenter HERGE ou HERNANDEZ).

Pour les critères exacts nous utiliseront des listes déroulantes ou Combo Box, et pour les autres des boites de saisie Text Box.

IV-A - Design et Contrôles

Nous allons créer un formulaire vierge indépendant sans assistant

dans lequel nous allons créer :



TextBox	ComboBox	CheckBox	Label	ListBox
txtRechAuteur		chkAuteur		
txtRechTitre		chkTitre		
txtRechResume		chkResume		
	cmbRechType	chkType		
	cmbRechFamille	chkFamille		
			lblStats	
				lstResults

Le choix des noms est toujours très important

J'utilise les trois premières lettres pour rappeller le type de contrôle : **txt** pour textbox, **cmb** pour combobox, **chk** pour checkbox, **lbl** pour Label et **lst** pour ListBox

Nous utilisons les check box pour déterminer si la sélection utilise ou non le critère.

ex : si la case chkTitre est cochée, la zone de saisie relative est affichée et l'utilisateur doit saisir une partie du titre qu'il recherche.

ex : si la case chkAuteur est décochée, l'utilisateur n'effectue pas de sélection sur l'Auteur.

Nous distinguerons deux types de codes : celui de l'interface utilisateur et celui de l'accès au données.

IV-B - Code d'interface

*** Masquer ou afficher la saisie du critère selon la case à cocher qui s'y rapporte**

Si l'utilisateur coche la case, valeur récupérée par Me.chkAuteur, le textbox de recherche est affiché pour permettre une saisie.

Nous reviendrons plus tard sur la Sub **RefreshQuery()**.

événement clic d'une case à cocher

```
Private Sub chkAuteur_Click()
   Me.txtRechAuteur.Visible = Not Me.txtRechAuteur.Visible
   RefreshQuery
End Sub
```

*** Remplir les combo box de sélection**

Ces combos sont remplies par la table elle même, on pourrait tout aussi bien utiliser des tables auxiliaires avec par exemple un code Type et un libellé Type.

*** Gestion des événements mise à jour**

Pour que le formulaire de recherche soit dynamique, nous n'utiliserons pas de bouton rechercher ici, toute modification des contrôles de recherche se répercute immédiatement sur le résultat de la recherche elle-même.

Il faut donc, pour chaque événement de ces contrôles mettre à jour nos résultats.

Pour éviter de répéter inutilement du code, nous allons créer une sub RefreshQuery qui se chargera de cette tâche.

Nous allons associer cette sub à chaque événement : pour un combo ou un text box l'événement **BeforeUpdate** et pour les check box dans l'événement **Click**

Pour appeler cette sub : il suffit de mettre son nom sur une ligne.

événements clic et mise à jour
```
   Private Sub chkAuteur_Click()
      Me.txtRechAuteur.Visible = Not  Me.txtRechAuteur.Visible
      RefreshQuery
   End Sub

   Private Sub cmbRechFamille_BeforeUpdate(Cancel As Integer)
      RefreshQuery
   End Sub

   Private Sub txtRechResume_BeforeUpdate(Cancel As Integer)
      RefreshQuery
   End Sub
```

IV-C - Code d'accès aux données

*** Création du SQL de résultat**

Comme vu précédemment nous traitons cette mise à jour par la sub suivante :

Le code commence par poser la variable String "SQL"

L'utilisation de cette Instruction Where est ici sans intérêt pour les résultats mais permet de mettre dans la chaine "SQL" le Where

La suite du code inspecte les valeurs des check box, si la check box est décochée, le code rajoute au SQL une condition.

Nous utilisons Like "*valeur*" pour les critères de recherches non exacts

et = "valeur" pour les critères exacts.

SQLWhere récupère ce qui est écrit dans le SQL après le mot Where pour l'utiliser dans les fonctions DCount().

DCount() permet de compteur le nombre d'enregistrements d'une table en fonction d'un critère, mais sans le mot clé "WHERE".

rappelons la syntaxe de DCount("[Champ]", "Table", [Champ1] = '" & Variable & "'")

Me.lblStats.Caption = DCount("*", "Medias", SQLWhere) & " / " & DCount("*", "Medias") : met à jour l'affichage des statistiques

Me.lstResults.RowSource = SQL : assigne l'instruction SQL fraichement créée de manière dynamique comme source des lignes de la liste des résultats.

Me.lstResults.Requery : permet d'exécuter la requête.

procédure : moteur de recherche

```
Private Sub RefreshQuery()
 Dim SQL As String
 Dim SQLWhere As String

 SQL = "SELECT CodMedia, Titre, Auteur, Famille, Type FROM Medias Where Medias!CodMedia <> 0 "
 If Me.chkAuteur Then
    SQL = SQL & "And Medias!Auteur like '*" & Me.txtRechAuteur & "*' "
 End If
 If Me.chkFamille Then
    SQL = SQL & "And Medias!Famille = '" & Me.cmbRechFamille & "' "
 End If
 If Me.chkResume Then
    SQL = SQL & "And Medias!Résumé like '*" & Me.txtRechResume & "*' "
 End If
 If Me.chkTitre Then
    SQL = SQL & "And Medias!Titre like '*" & Me.txtRechTitre & "*' "
 End If
 If Me.chkType Then
    SQL = SQL & "And Medias!Type = '" & Me.cmbRechType & "' "
 End If

 SQLWhere = Trim(Right(SQL, Len(SQL) - InStr(SQL, "Where ") - Len("Where ") + 1))
 SQL = SQL & ";"

 Me.lblStats.Caption = DCount("*", "Medias", SQLWhere) & " / " & DCount("*", "Medias")
 Me.lstResults.RowSource = SQL
 Me.lstResults.Requery

End Sub
```

*** Gestion des paramètres d'ouverture** Nous souhaitons qu'à l'ouverture du formulaire aucun critère ne vienne filtrer la table, tout se gère sur l'événement **Load** du formulaire.

Pour plus d'efficacité nous allons utiliser la collection **Controls** du formulaire et nous servir du nom des contrôles.

Select Case Left(ctl.Name, 3) : permet de gérer les 3 premières lettres du nom du contrôle que nous avons choisies astucieusment.

Si le contrôle est une check box : nous cochons la case (ctl.value = -1)

Si le contrôle est une text box : nous vidons le contenu (ctl.value = "") et nous la masquons (ctl.visible=False)

Si le contrôle est un label : nous réinitialisons son étiquette (ctl.caption = ""), dans notre application nous n'en avons qu'un lblStats

Si le contrôle est une combo box : nous la masquons (ctl.visible = False)

Le code réinitialise la liste des résultats en assignant un SQL sans condition Where, et sans oublier la mise à jour avec la méthode **.Requery**

initialisation des contrôles sur le chargement du formulaire

```
Private Sub Form_Load()
Dim ctl As Control

For Each ctl In Me.Controls
    Select Case Left(ctl.Name, 3)
        Case "chk"
            ctl.Value = -1
        Case "lbl"
            ctl.Caption = "- * - * -"
        Case "txt"
            ctl.Visible = False
            ctl.Value = ""
        Case "cmb"
            ctl.Visible = False
    End Select
Next ctl

Me.lstResults.RowSource = "SELECT CodMedia, Titre, Auteur, Famille, Type FROM Medias;"
Me.lstResults.Requery

End Sub
```

V - Complément

Pour perfectionner l'application nous allons lui ajouter une fonctionnalité :

la possibilité de modifier un enregistrement par un double clic sur la liste des résultats.

Notre but :

en fonction du choix de l'utilisateur nous allons ouvrir un formulaire de saisie/modification de l'enregistrement qu'il aura pointé.

Les moyens :

Création d'un formulaire instantanné à partir de la table "Medias"

Gestion de l'événement Double Clic [DblClick] du contrôle lstResults

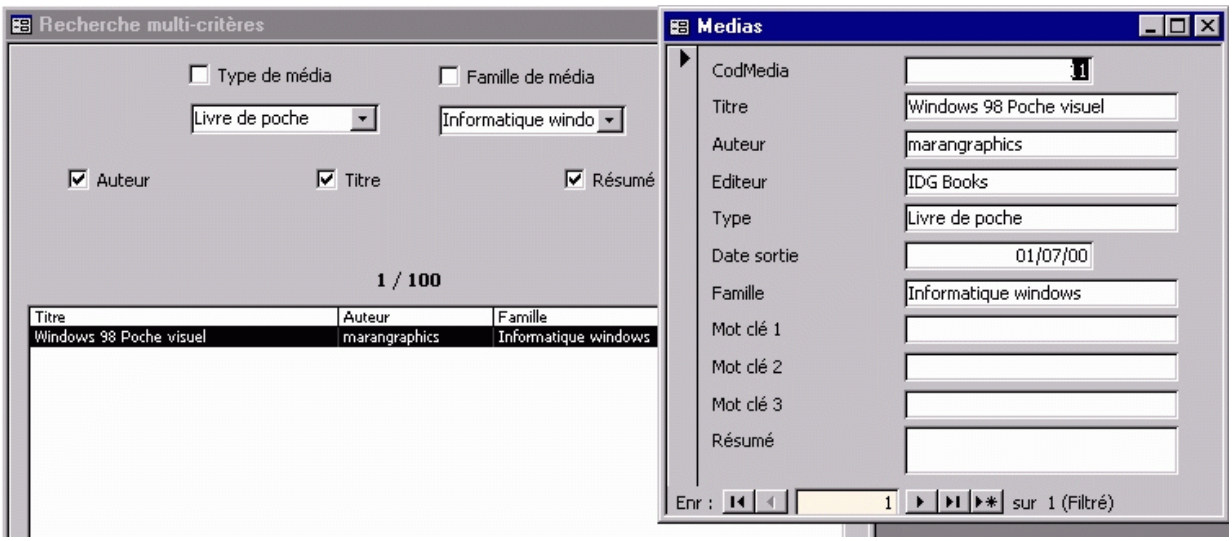
événement double-clic de la liste des résultats
```
Private Sub lstResults_DblClick(Cancel As Integer)
   DoCmd.OpenForm "frmAutoMedias", acNormal, , "[CodMedia] = " & Me.lstResults
End Sub
```

Pour que cette fonctionnalité marche, il faut que la propriété "Colonne Liée" de lstResults soit 1, c'est à dire le numéro de la colonne qui contient le code du média. Ainsi Me.lstResults renverra le code choisi.

lorsque l'utilisateur double clique sur une ligne de la liste des réponses nous ouvrons le formulaire automatique avec une condition Where.

"[CodMedia] = " & Me.lstResults : permet de positionner le formulaire sur l'enregistrement cliqué.

Résultat :



VI - Rappel bref de notions SQL

Ce projet utilise quelques notions de SQL.

Ce tutoriel n'a pas la prétention d'apprendre le SQL, mais il cherche simplement à donner quelques notions qui peuvent être utiles à la compréhension du formulaire.

Pour en savoir plus sur le SQL : http://sqlpro.developpez.com/

Les requêtes qui nous intéressent ici sont les requêtes sélections ici sur une seule table, dont la structure est du type suivant :

SELECT [obligatoire]

liste des champs séparés par une virgule, au besoin renommés par un alias grâce à l'instruction "As"

les champs choisis vont être les colonnes de la requête.

pour sélectionner tous les champs dans une requête sur une seule table : SELECT *

pour sélectionner tous les champs dans une requête sur plusieurs tables : SELECT Table1.*

ex : SELECT NomFamille As NomF, PrenomUsuel As Prenom, DateNaissance

3 colonnes NomF, Prenom et DateNaissance

FROM [obligatoire]

table sur laquelle porte la requête, au besoin elle aussi renommée par un alias grâce à l'instruction "As"

ex : FROM tblPersonnel

la requête porte sur la table "tblPersonnel"

WHERE [Facultatif]

liste des conditions séparées par un opérateur logique "And" ou "Or"

les conditions sont exprimées : Champ = Valeur

ex : WHERE tblPersonnel.NomFamille = 'MARTIN'

ici nous sélectionnons les personnes qui portent le nom de Martin (Claude Martin et Jean Martin)

ex : WHERE tblPersonnel.NomFamille = 'MARTIN' And tblPersonnel.PrenomUsuel Like 'C*'

ici nous sélectionnons les personnes qui portent le nom de Martin ET dont le prénom comment par un C (Claude Martin, mais pas Jean Martin)

ex : WHERE tblPersonnel.NomFamille = 'MARTIN' Or tblPersonnel.PrenomUsuel Like 'C*'

ici nous sélectionnons les personnes qui portent le nom de Martin OU dont le prénom comment par un C (Claude Martin et Jean Martin mais aussi Christian Janvier)

Le SQL se termine par un ";"

NB : le point virgule est requis dans Access mais pas nécessairement pour toutes les implémentations de base de données.

SQL avec '=' et avec 'Like'

```
SELECT NomFamille as NomF, PrenomUsuel As Prenom, DateNaissance
FROM tblPersonnel
WHERE tblPersonnel.NomFamille = 'MARTIN' Or tblPersonnel.PrenomUsuel Like 'C*';
```

NomF	Prenom	DateNaissance
Martin	Claude	30/06/1960
Martin	Jean	31/01/1959
Janvier	Christian	28/02/1964

VI - Fonctions de domaine

Dans ce tutoriel nous avons utilisé des fonctions de domaine pour afficher les statistiques de la requête.

affichage des statistiques de la requête
```
Me.lblStats.Caption = DCount("*", "Medias", SQLWhere) & " / " & DCount("*", "Medias")
```

Nous verrons ici les fonctions **DCount()**, **DLookUp()** et **DSum()**

Elles fonctionnent toutes selon le même modèle d'arguments :

DLookup(*expr As **String***, *domaine As **String***[, *critère As **String***])

cette fonction permet de trouver la première correspondance à l'intérieur d'une table ou d'une requête selon une condition.

DCount(*expr As **String***, *domaine As **String***[, *critère As **String***])

cette fonction bâtie sur le même modèle compte le nombre de réponses satisfaisant la condition.

Dans notre exemple nous l'avons utilisée deux fois :

DCount("*", "Medias") : pas de condition, la fonction renvoie le nombre total d'enregistrement de la table Medias.

DCount("*", "Medias", SQLWhere) : on applique la condition formulée par les contrôles du formulaire, la fonction renvoie le nombre d'enregistrement correspondant.

DSum(*expr As **String***, *domaine As **String***[, *critère As **String***])

cette fonction bâtie sur le même modèle fait la somme des réponses satisfaisant la condition

VIII - Telechargement

Vous pouvez télécharger la base exemple en cliquant ici.