

Création et paramétrage dynamique de requêtes

par [Christophe WARIN](#)

Date de publication : 19/05/2005

Dernière mise à jour :


Ce tutoriel est destiné aux lecteurs souhaitant créer dynamiquement des requêtes paramétrables et ainsi, améliorer leurs systèmes de sélection de critères dans des requêtes Microsoft Access.

- 1 - Introduction
- 2 - Pourquoi un tel besoin ?
- 3 - Principe de réalisation
- 4 - Création de la requête modèle
- 5 - Création du formulaire
- 6 - Création de la nouvelle requête
 - 6.1 - Déclaration de la référence DAO
 - 6.2 - Détails du fonctionnement
 - 6.3 - Comment tester si une requête existe déjà
 - 6.4 - Modification de la requête modèle
 - 6.5 - Sauvegarde de la nouvelle requête
 - 6.6 - Affichage des données
- 7 - Visualisation
- 8 - Aller plus loin ...
 - 8.1 - Code du bouton Valider
 - 8.2 - Exemple d'utilisation
- 9 - Conclusion

1 - Introduction

Ce petit tutoriel a pour objectif de vous apprendre à réaliser une interface simple permettant de saisir le critère d'une requête paramétrée depuis un formulaire dans lequel une zone de liste modifiable fera office de paramètre.

Il ne nécessite pas de réelles connaissances en VBA et les points difficiles abordés ne demandent pas à être maîtrisés mais simplement compris.

 *La table utilisée en guise d'exemple est issue de la base de données Comptoir.mdb disponible dans le répertoire Samples d'Office*

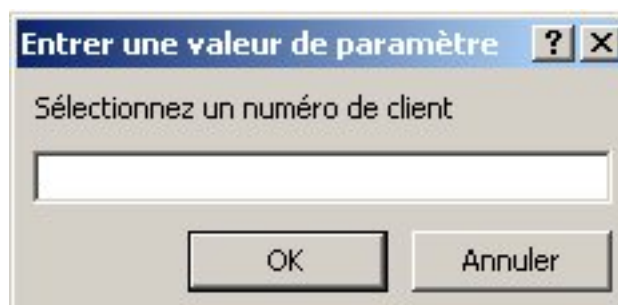
2 - Pourquoi un tel besoin ?

Access permet en effet à l'utilisateur de saisir des critères dans une requête. Toutefois, il est impossible de formater la saisie, ni même de la contrôler. Prenons le cas d'une requête sélectionnant un client en fonction de son numéro.

Cette requête serait :

```
SELECT * FROM Clients
WHERE Code_Client=[Sélectionnez un numéro de client];
```

Access traduit ce qu'il y a entre crochet en affichant une fenêtre de saisie, c'est ce que l'on appelle le *paramètre* :



Toutefois si le numéro à saisir est de type numérique, rien n'empêche l'utilisateur de saisir des caractères. Evidemment dans ce cas, la requête ne retournera aucun enregistrement et dans le pire des cas, un message d'erreur apparaîtra

3 - Principe de réalisation

Pour contourner ce problème, il nous faut créer notre propre fenêtre de saisie.

Dans un premier temps, nous allons créer une requête paramétrée classique qui va nous servir de modèle, le nom du critère étant quant à lui, écrit entre crochets.

Ensuite, il nous faut créer un formulaire de saisie. Ainsi, nous pouvons imaginer placer une zone de liste pour le choix d'un pays, un calendrier pour le choix d'une date, etc # les possibilités sont multiples.

Enfin, nous rédigerons une nouvelle requête temporaire basée sur la requête modèle mais en modifiant les critères entre crochets par leur valeur.

4 - Création de la requête modèle

Soit la table **Clients** suivante :

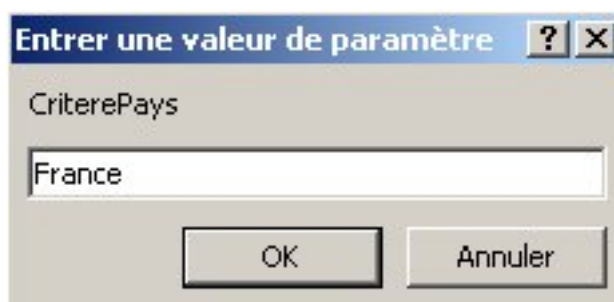
Clients : Table			
	Nom du champ	Type de données	Description
🔑	Code_client	Texte	Code unique de cinq caractères basé sur le nom du client.
	Société	Texte	
	Contact	Texte	
	Fonction	Texte	
	Adresse	Texte	Rue ou boîte postale.
	Ville	Texte	
	Région	Texte	État ou province.
	Code_postal	Texte	
	Pays	Texte	
	Téléphone	Texte	Le numéro de téléphone inclut le code du pays ou de la zone.
▶	Fax	Texte	Le numéro de téléphone inclut le code du pays ou de la zone.

Tout d'abord, créez une requête permettant d'afficher tous les clients habitant dans un pays sélectionné.

La requête SQL est donc :

```
SELECT * FROM Clients
WHERE Pays=[CriterePays];
```

Testez la requête avec le pays *France* :



Le résultat est le suivant :

Requête1 : Requête Sélection				
	Code client	Société	Contact	Fonct
▶	BLONP	Blondel père et fils	Frédérique Citeaux	Directeur du r
	BONAP	Bon app'	Laurence Lebihan	Propriétaire
	DUMON	Du monde entier	Janine Labrune	Propriétaire
	FOLIG	Folies gourmandes	Martine Rancé	Représentant(
	FRANR	France restauration	Carine Schmitt	Directeur du r
	LACOR	La corne d'abondance	Daniel Tonini	Représentant(
	LAMAI	La maison d'Asie	Annette Roulet	Chef des vente
	PARIS	Paris spécialités	Marie Bertrand	Propriétaire
	SPECD	Spécialités du monde	Dominique Perrier	Directeur du r
	VICTE	Victuailles en stock	Mary Saveley	Assistant(e) d
	VINET	Vins et alcools Chevalier	Paul Henriot	Chef comptabl
*				

Enr :  1  sur 11 

La requête fonctionne, enregistrez-la sous le nom de **qryClientParPays_modele**.

5 - Création du formulaire

Vous allez maintenant créer un nouveau formulaire en mode création où vous poserez une zone de liste modifiable nommée **cboPays** et un bouton nommé **btValider**.

La zone de liste va permettre de proposer le choix du pays à l'utilisateur. Il vous faut donc lui définir quelques propriétés :

Origine Source : Table/Requête

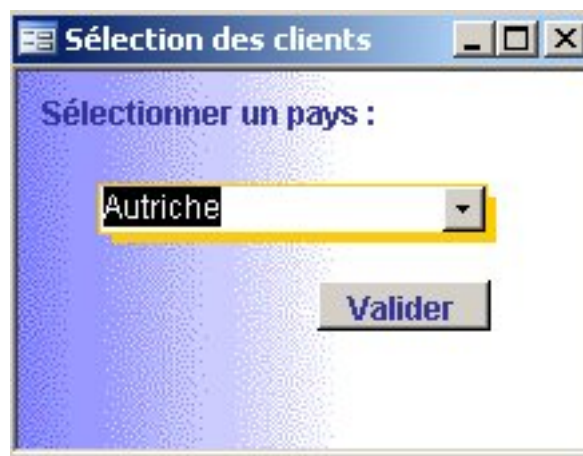
Contenu :

```
SELECT DISTINCT Pays FROM Clients ORDER BY Pays
```

Cette requête renvoie la liste des pays figurant dans la table Clients en les triant par ordre alphabétique. Le mot clé **DISTINCT** permet de retirer les doublons de la sélection.

Pour que le formulaire se comporte comme une boîte de dialogue Windows, il faut définir ses propriétés **Fenêtre Indépendante** et **Modale** à **Oui** (Onglet : autres).

Après avoir retiré les barres de défilement, le diviseur d'enregistrements, le sélecteur ainsi que les boutons de déplacements (Onglet Format) vous obtiendrez aisément quelque chose de comparable à cette fenêtre :



Sauvegardez ce formulaire et nommez-le **frmSelectionClientPays**.

6 - Création de la nouvelle requête

6.1 - Déclaration de la référence DAO

La création d'une requête sous Visual Basic implique la manipulation d'objets de la base de données. Pour cela nous allons utiliser la librairie **DAO** qui permet de telles acrobaties.

Pour utiliser cette librairie vous devez ajouter la référence Microsoft DAO 3.6 à notre projet. Dans l'éditeur **VBA**, cliquez sur **Outils**, puis **Références**.

Cochez la ligne intitulée **Microsoft DAO 3.6 Object Library**.

6.2 - Détails du fonctionnement

Vous disposez maintenant d'une requête nommée **qryClientParPays_modele** qui contient le modèle de la requête à créer. Sur le bouton **Valider** du formulaire, il vous faut récupérer son code SQL, modifier le critère entre crochets, puis sauvegarder le tout dans une nouvelle requête. Enfin, vous devrez exécuter cette requête nouvellement créée afin d'afficher les données voulues.

Un problème majeur se pose alors. A la première exécution, vous n'allez rencontrer aucune difficulté. La requête va bien s'enregistrer sous le nom de **qryClientParPays**. Mais à la prochaine exécution, lorsque le programme souhaitera enregistrer la requête, une erreur sera levée. En effet, le nom **qryClientParPays** sera déjà utilisé. Il vous faudra donc procéder à un test d'existence avant d'enregistrer la nouvelle requête sans quoi le programme ne pourra fonctionner. Soit elle n'existe pas, dans ce cas, il faut la créer. Soit elle existe déjà, et dans ce cas, il faut la modifier

6.3 - Comment tester si une requête existe déjà

Pour vérifier qu'une requête existe déjà, il est nécessaire de rédiger une petite fonction à placer dans un module. Inutile de s'attarder sur le code. L'essentiel est que vous sachiez l'utiliser.

```
Public Function TesteExistenceRequete(strNomRequete As String) As Boolean
On Error GoTo err
Dim Db As DAO.Database
Dim QryTest As DAO.QueryDef
'initialise un objet représentant la base de données
Set Db = CurrentDb
'tente d'accéder à la requête
Set QryTest = Db.QueryDefs(strNomRequete)
TesteExistenceRequete = True
err:
End Function
```

En fait, le programme tente d'accéder à la requête dont le nom est passé en paramètre. S'il y parvient, la fonction retourne **True** (Vrai), sinon la fonction retourne son état d'origine soit **False** (Faux).

6.4 - Modification de la requête modèle

Procédez de la même façon que dans la fonction ci-dessus pour atteindre la requête modèle et en extraire le code SQL à l'aide de la propriété SQL de l'objet **QueryDef**.

Ainsi, dans la propriété **Sur Clic** du bouton **Valider**, créez une procédure événementielle contenant le code suivant

:

```
Dim Db As DAO.Database
Dim QryModele As DAO.QueryDef
Dim strSQLModele As String
Set Db = CurrentDb
Set QryModele = Db.QueryDefs("qryClientParPays_modele")
strSQLModele = QryModele.SQL
```

A partir de là, il vous devez modifier le mot [criterepays] de la chaîne de caractères par la valeur contenue dans la zone de liste. Suivant le type de champ utilisé par le critère, il faut procéder de différentes manières.

- Le champ est de type texte (Le cas ici) :

Le code SQL devra se présenter de cette manière :

```
SELECT * FROM Clients
WHERE Pays="FRANCE";
```

Pour créer une telle chaîne sous VBA, il nous faut donc encadrer la valeur de la zone de liste par des guillemets, ce que fait la fonction **Chr(34)**.

- Le champ est une date :

Access stocke les dates au format américain (mm/dd/yyyy) et les dates sont à encadrer par des #.

Exemple :

```
SELECT * FROM Clients
WHERE DateNaissance=#10/22/1976#
```

- Le champ est d'un autre type :

Aucune précaution n'est alors nécessaire et vous pouvez écrire le critère directement.

Complétez le code de la façon suivante :

```
strSQLModele = Replace(strSQLModele, "[criterepays]", Chr(34) & Nz(cboPays) & Chr(34))
```

Décortiquons cette ligne en détail :

La fonction **Nz** permet de traiter le cas où la valeur de la zone de liste serait nulle. Dans ce cas, au lieu de renvoyer NULL, la fonction Nz retourne "" (une chaîne vide). Nous encadrons ensuite la valeur retournée par cette fonction avec les guillemets fournies par la fonction **Chr(34)**.

Enfin, la fonction **Replace** effectue le remplacement du critère par la nouvelle valeur obtenue

6.5 - Sauvegarde de la nouvelle requête

A ce stade le code SQL créé est prêt à être enregistré dans une nouvelle requête nommée **qryClientParPays**. Je vous rappelle toutefois que nous devons vérifier que la requête n'existe pas avant de l'enregistrer dans un nouvel objet **DAO**, sans quoi le programme générera un bug.

Le code devient alors :

```
'Si la requête existe déjà alors
If TesteExistenceRequete("qryClientParPays") Then
  'alors modifier le code de la requête
  Db.QueryDefs("qryClientParPays").SQL = strSQLModele
'Sinon
Else
  'Créer la nouvelle requête
  Db.CreateQueryDef "qryClientParPays", strSQLModele
End If
```

J'insiste sur le fait que la méthode **CreateQueryDef** possède comme premier paramètre le nom de la requête à créer et comme second le code représenté par la chaîne SQL correspondante. Cette méthode crée et enregistre une nouvelle requête dans la base de données.

6.6 - Affichage des données

Il ne reste plus qu'à ouvrir la nouvelle requête pour afficher le résultat. Pour cela, utilisez la méthode `DoCmd.OpenQuery`.

```
DoCmd.OpenQuery "qryClientParPays"
```

Le code devient donc :

```
Private Sub btValider_Click()
Dim Db As DAO.Database
Dim QryModele As DAO.QueryDef
Dim strSQLModele As String
Set Db = CurrentDb
Set QryModele = Db.QueryDefs("qryClientParPays_modele")
strSQLModele = QryModele.SQL
'Effectue le remplacement du critere par la valeur
strSQLModele = Replace(strSQLModele, "[criterepays]", Chr(34) & Nz(cboPays) & Chr(34))
'Si la requête existe déjà alors
If TesteExistenceRequete("qryClientParPays") Then
  'alors modifier le code de la requête
  Db.QueryDefs("qryClientParPays").SQL = strSQLModele
'Sinon
Else
  'Créer la nouvelle requête
  Db.CreateQueryDef "qryClientParPays", strSQLModele
End If
'Ouvre la requête
DoCmd.OpenQuery "qryClientParPays"
'Ferme le formulaire
DoCmd.Close acForm, Me.Name
End Sub
```

Afin de parfaire le fonctionnement de cet exemple, vous pouvez ajouter une routine de gestion d'erreur afin d'alerter l'utilisateur lors d'un dysfonctionnement.

```
Private Sub btValider_Click()
On Error GoTo err
Dim Db As DAO.Database
Dim QryModele As DAO.QueryDef
Dim strSQLModele As String
Set Db = CurrentDb
Set QryModele = Db.QueryDefs("qryClientParPays_modele")
```

```
strSQLModele = QryModele.SQL
'Effectue le remplacement du critere par la valeur
strSQLModele = Replace(strSQLModele, "[criterepays]", Chr(34) & Nz(cboPays) & Chr(34))
'Si la requête existe déjà alors
If TesteExistenceRequete("qryClientParPays") Then
  'alors modifier le code de la requête
  Db.QueryDefs("qryClientParPays").SQL = strSQLModele
'Sinon
Else
  'Créer la nouvelle requête
  Db.CreateQueryDef "qryClientParPays", strSQLModele
End If
DoCmd.OpenQuery "qryClientParPays"
'Ferme le formulaire
DoCmd.Close acForm, Me.Name
Exit Sub
err:
MsgBox "Une erreur est survenue", vbCritical, "Sélection d'un client"
End Sub
```

De même, les puristes préféreront utiliser des variables pour stocker le nom des requêtes. Ce qui pourrait donner :

```
Private Sub btValider_Click()
On Error GoTo err
Dim Db As DAO.Database
Dim QryModele As DAO.QueryDef
Dim strSQLModele As String
Dim strNomQueryModele As String
Dim strNomQuery As String
strNomQueryModele = "qryClientParPays_modele"
strNomQuery = "qryClientParPays"
Set Db = CurrentDb
Set QryModele = Db.QueryDefs(strNomQueryModele)
strSQLModele = QryModele.SQL
'Effectue le remplacement du critere par la valeur
strSQLModele = Replace(strSQLModele, "[criterepays]", Chr(34) & Nz(cboPays) & Chr(34))
'Si la requête existe déjà alors
If TesteExistenceRequete(strNomQuery) Then
  'alors modifier le code de la requête
  Db.QueryDefs(strNomQuery).SQL = strSQLModele
'Sinon
Else
  'Créer la nouvelle requête
  Db.CreateQueryDef strNomQuery, strSQLModele
End If
DoCmd.OpenQuery strNomQuery
'Ferme le formulaire
DoCmd.Close acForm, Me.Name
Exit Sub
err:
MsgBox "Une erreur est survenue", vbCritical, "Sélection d'un client"
End Sub
```

7 - Visualisation

Pour visualiser le résultat il vous suffit d'ouvrir le formulaire en mode normal, de sélectionner un pays et de valider.

Le résultat obtenu est bien le même que lorsque vous aviez saisi le pays en toute lettre au début de ce document.

Requête1 : Requête Sélection				
	Code client	Société	Contact	Fonct
▶	BLONP	Blondel père et fils	Frédérique Citeaux	Directeur du r
	BONAP	Bon app'	Laurence Lebihan	Propriétaire
	DUMON	Du monde entier	Janine Labrune	Propriétaire
	FOLIG	Folies gourmandes	Martine Rancé	Représentant(
	FRANR	France restauration	Carine Schmitt	Directeur du r
	LACOR	La corne d'abondance	Daniel Tonini	Représentant(
	LAMAI	La maison d'Asie	Annette Roulet	Chef des vente
	PARIS	Paris spécialités	Marie Bertrand	Propriétaire
	SPECD	Spécialités du monde	Dominique Perrier	Directeur du r
	VICTE	Victuailles en stock	Mary Saveley	Assistant(e) d
	VINET	Vins et alcools Chevalier	Paul Henriot	Chef comptabl
*				

Enr : sur 11


8 - Aller plus loin ...

Vous remarquerez que le mode opératoire ci-dessus vous oblige à créer autant de requêtes modèles que de requêtes à utiliser au final. Aussi, à un stade où l'application aura beaucoup été utilisée vous vous retrouverez avec le double de requêtes que nécessaire. Une solution consiste alors à stocker le code SQL des requêtes modèles dans une table locale et de lire cette table afin de créer de véritables requêtes enregistrées.

Ainsi par exemple nous pourrions créer la table suivante :

Champ	Type	Taille
NomRqt	Texte	50 (Clé primaire)
CodeRqt	Texte	255
DescriptionRqt	Texte	50

Vous enregistrerez cette table sous le nom : **tblRequete**.

 Dans le cas où vous auriez à stocker de grandes requêtes, vous pouvez utiliser le type **Mémo** pour le champ **CodeRqt**.

8.1 - Code du bouton Valider

Afin d'utiliser cette table pour lire le SQL des requêtes il est nécessaire de modifier le code du bouton **Valider** de la sorte :

```
Private Sub btValider_Click()
On Error GoTo err
Dim Db As DAO.Database
Dim QryModele As DAO.QueryDef
Dim RstTable As DAO.Recordset
Dim strSQLModele As String
Dim strNomRqt As String
'Instancie la base de données
Set Db = CurrentDb
'Ouvre la table des requetes sur l'enregistrement
'dont le nom de la requête est "qryClientParPays"
strNomRqt = "qryClientParPays"
Set RstTable = Db.OpenRecordset("SELECT * FROM tblRequete WHERE NomRqt=" & Chr(34) & strNomRqt & Chr(34))
'Si la requête est trouvée dans la table,
If Not RstTable.EOF Then
'Récupère le code correspondant
strSQLModele = RstTable.Fields("CodeRqt")
'Effectue le remplacement du critere par la valeur
strSQLModele = Replace(strSQLModele, "[criterepays]", Chr(34) & Nz(cboPays) & Chr(34))
'Si un objet requête existe déjà alors
If TesteExistenceRequete(strNomRqt) Then
'alors modifier le code de la requête
Db.QueryDefs(strNomRqt).SQL = strSQLModele
'Sinon
Else
'Créer la nouvelle requête
Db.CreateQueryDef strNomRqt, strSQLModele
End If
DoCmd.OpenQuery "qryClientParPays"
'Ferme le formulaire
DoCmd.Close acForm, Me.Name
'Sinon
Else
'Avertir que le code SQL n'est pas disponible
MsgBox "Impossible de trouver la requête : " & strNomRqt & " dans la table des requêtes"
End If


Exit Sub
err:
```

```
MsgBox "Une erreur est survenue" & vbCrLf & err.Description, vbCritical, "Sélection d'un client"
End Sub
```

Quelques explications :

Pour lire les données d'une table nous devons utiliser des objets de type *Recordset* (littéralement : jeu d'enregistrements). Il s'agit ni plus ni moins du résultat d'une requête et chaque enregistrement est stocké en mémoire. Lorsque le *Recordset* est vide à son ouverture, sa propriété **EOF** est égale à **True**. Cette propriété signifie en fait que l'on se trouve après le dernier enregistrement, ce qui est logique si le jeu est vide. Enfin, il est possible d'accéder à un champ du *Recordset* pour la ligne courante à l'aide de la syntaxe : `RstTable.Fields("Nomduchamp")`

Une fois le code SQL lu et stocké dans une variable (**strSQLModele**), le principe de création de la requête reste le même que celui utilisé pour les requêtes modèles tout en gardant à l'esprit que la requête à créer peut déjà exister.

 Vous trouverez un cours complet sur l'accès aux données via DAO à l'adresse suivante : <http://warin.developpez.com/TutorielDAO>

8.2 - Exemple d'utilisation

Pour que cela fonctionne il suffit maintenant de stocker notre requête modèle dans la table.

tblRequete : Table		
NomRqt	CodeRqt	DescriptionRqt
qryClientParPays	SELECT * FROM Clients WHERE (((Clients.Pays)=[CriterePays]));	Requête de sélection des clients par pays

Il ne reste plus qu'à tester le formulaire en saisissant de nouveau un pays...



... et constater que cela fonctionne.

Requête1 : Requête Sélection				
	Code client	Société	Contact	Fonct
▶	BLONP	Blondel père et fils	Frédérique Citeaux	Directeur du r
	BONAP	Bon app'	Laurence Lebihan	Propriétaire
	DUMON	Du monde entier	Janine Labrune	Propriétaire
	FOLIG	Folies gourmandes	Martine Rancé	Représentant(
	FRANR	France restauration	Carine Schmitt	Directeur du r
	LACOR	La corne d'abondance	Daniel Tonini	Représentant(
	LAMAI	La maison d'Asie	Annette Roulet	Chef des vente
	PARIS	Paris spécialités	Marie Bertrand	Propriétaire
	SPECD	Spécialités du monde	Dominique Perrier	Directeur du r
	VICTE	Victuailles en stock	Mary Saveley	Assistant(e) d
	VINET	Vins et alcools Chevalier	Paul Henriot	Chef comptabl
*				

Enr :  1  sur 11 

9 - Conclusion

Vous voilà maintenant paré à améliorer l'ergonomie de vos applications simples.

Notez que rien ne vous empêche d'étendre les possibilités de cet exemple et de réaliser une interface permettant de saisir plusieurs critères.

Je tiens à remercier l'ensemble de l'équipe Access notamment Argyronet et reste à votre disposition pour d'éventuelles remarques par MP.