

Comment créer des requêtes d'analyses croisées conditionnelles

par Fabrice CONSTANS ([autres articles](#))

Date de publication : 20/04/2005

Dernière mise à jour : 01/10/2009

Vous avez sûrement déjà utilisé un contrôle de formulaire pour paramétrer une requête Select. Mais avec une requête d'analyse croisée ?! Cet article se propose de vous guider dans la réalisation de cette tâche.

1 - Avertissement.....	3
2 - Pré requis - Les objets ACCESS nécessaires pour l'exercice.....	4
2.1 - Formulaire et zone liste modifiable.....	4
2.2 - La table.....	4
3 - La requête d'analyse croisée.....	6
3.1 - Création.....	6
3.2 - Le critère de sélection.....	6
4 - Méthode "Full Queries".....	8
4.1 - Les objets requêtes.....	8
4.2 - La procédure Visual Basic.....	9
5 - Méthode Visual Basic pour Application.....	11
5.1 - Création de la requête.....	11
5.2 - Création de la procédure Visual Basic.....	11
6 - Conclusion.....	13
7 - Remerciements.....	14

1 - Avertissement

Ce tutoriel n'ayant pas pour but l'apprentissage du Visual Basic pour Application ni du langage SQL, mais la démonstration que certaines impossibilités d'ACCESS peuvent être contournées, j'ai choisi de ne pas agrémenter le code du traitement des erreurs et autres vérifications d'usages, ni la zone liste d'une liste dynamique.



L'utilisation de la touche F1 est vivement conseillée à tous les stades de l'utilisation d'ACCESS. L'amélioration constante de l'aide en fait un partenaire de choix dans l'apprentissage permanent d'ACCESS. Personnellement je ne peux m'en passer, ne serait-ce que pour mémoire.

2 - Pré requis - Les objets ACCESS nécessaires pour l'exercice.

Pour réaliser cet exercice ACCESS nous aurons besoin de créer une petite IHM simple composée d'un formulaire contenant une zone de liste et deux boutons de commandes.

IHM : Interface Homme-Machine tout ce qui permet à l'utilisateur d'exploiter des données. Les formulaires et états sont les composants d'une IHM.

2.1 - Formulaire et zone liste modifiable

Pour commencer nous allons définir les objets nécessaires pour la réalisation de l'exercice.

Ouvrez un fichier de base de données Microsoft ACCESS vierge. Commencez par créer un formulaire que vous nommerez **Analyse Croisée**.

Dans ce formulaire ajoutez une zone de liste déroulante nommée **ZI_critere**. Celle-ci contiendra le critère pour la requête d'analyse croisée.

Les propriétés de **ZI_critere** sont les suivantes :

Propriété	Valeur
Nom	ZI_critere
Origine source	Liste de valeurs
Contenu	2001;2002
Nbre colonnes	1

Sauvegardez le formulaire, passez-le en mode utilisation et sélectionnez la valeur 2001 dans la zone liste. La valeur de cette zone liste constituera notre critère.



Voici ZI_critere !

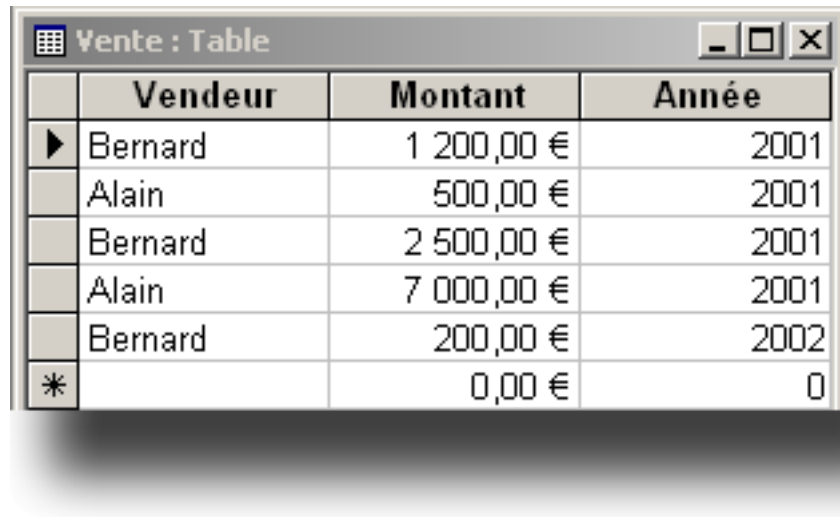
⚠ Ne fermez pas le formulaire avant la fin de l'exercice.

2.2 - La table

Créez une table nommée **Vente** avec les champs suivants :

Nom des champs	Type	longueur/type
Vendeur	Texte	50
Montant	Monétaire	
Année	Numérique	Entier

Et comme une table ACCESS n'est jamais vide, saisissez les enregistrements ci-dessous.



	Vendeur	Montant	Année
▶	Bernard	1 200,00 €	2001
	Alain	500,00 €	2001
	Bernard	2 500,00 €	2001
	Alain	7 000,00 €	2001
	Bernard	200,00 €	2002
*		0,00 €	0

Vu de la table

3 - La requête d'analyse croisée

3.1 - Création

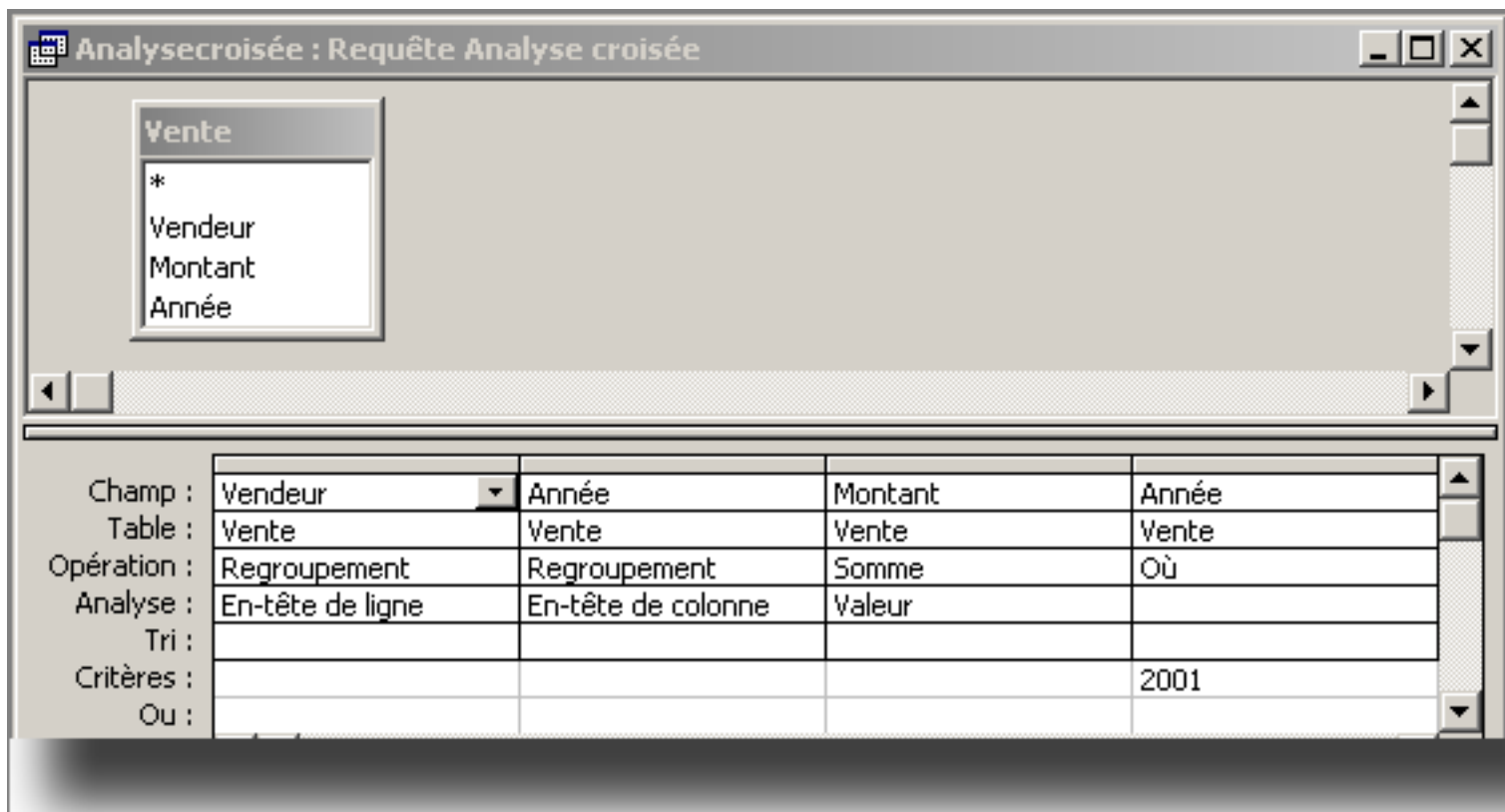
Pour la requête d'analyse croisée nous allons faire très simple. Ce n'est qu'un exercice.

Ouvrez une nouvelle requête et insérez la table **Vente**.

Fermez le panneau d'insertion de table.

Sélectionnez dans le menu *Requête* l'item *Analyse Croisée*.

Puis faites glisser les champs de la table un à un comme l'exemple présenté ci-dessous :



Vu de la requête en mode création.

Testez la requête et sauvegardez-la sans la fermer.

3.2 - Le critère de sélection

Nous allons tenter à présent de lier la zone liste contenant le critère à notre belle requête. Pour cela nous allons devoir rajouter du code dans la cellule *Critère* de la dernière colonne à la place du 2001.

Le lien vers notre critère

```
[formulaires]![Analyse Croisée].[Z1_Critere]
```

Exécutez la requête... Une erreur se produit dont le message est plus qu'explicite.



Le moteur de la base de données Microsoft Jet ne reconnaît pas '[formulaires]![Analyse Croisée].[Zl_Critere]' en tant que nom de champ ou expression correcte.

Le vilain message d'erreur !

Nous venons de démontrer que la requête d'analyse croisée dans Access ne supporte pas les critères liés. Si vous remplacez le critère "[forms]![Analyse Croisée].[zl_critere]" par "2001" vous verrez que cela fonctionne. Vous pouvez essayer avec une requête Select intermédiaire pour le critère, le résultat est malheureusement identique, à cause de la référence faite à un contrôle de formulaire.

Ce problème peut être contourné de trois manières.

La première consiste à utiliser le tableau des paramètres **clic droit / Paramètres** de la requête et d'y inscrire les liens vers les contrôles. [forms]![Analyse Croisée].[zl_critere] n'oubliez pas de mettre le type de données dans la 2ème colonne !

Il arrive que sur de grosses requêtes trop complexes, Access éprouve des problèmes d'exécution. Dans ce cas il faut utiliser d'autres méthodes.

La deuxième est à base de requêtes. En pleine contradiction avec notre démonstration ?!. Pas tout à fait puisqu'elle consiste à créer une table temporaire ne contenant que les enregistrements concernés pour l'analyse. Facile à mettre en oeuvre car sans utilisation du VBA, elle peut être automatisée dans une macro.

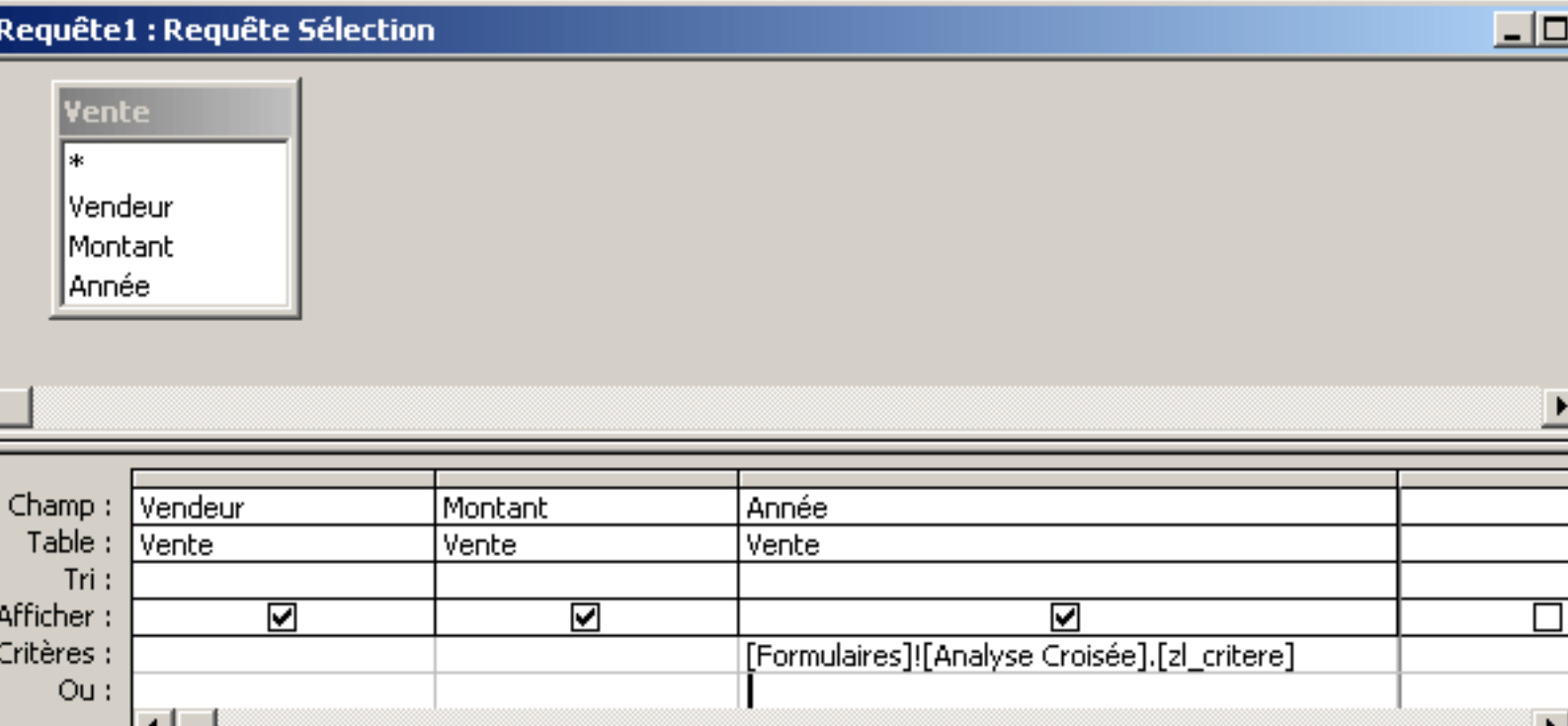
L'autre fait appel au VBA et à l'avantage d'être plus rapide à exécuter sur de gros volumes de données car ne mettant en oeuvre qu'une seule requête.

4 - Méthode "Full Queries"

Pour mettre en oeuvre cette méthode vous devez créer une requête création de table, une requête d'analyse croisée et une procédure en Visual Basic pour Application (possible par macro).

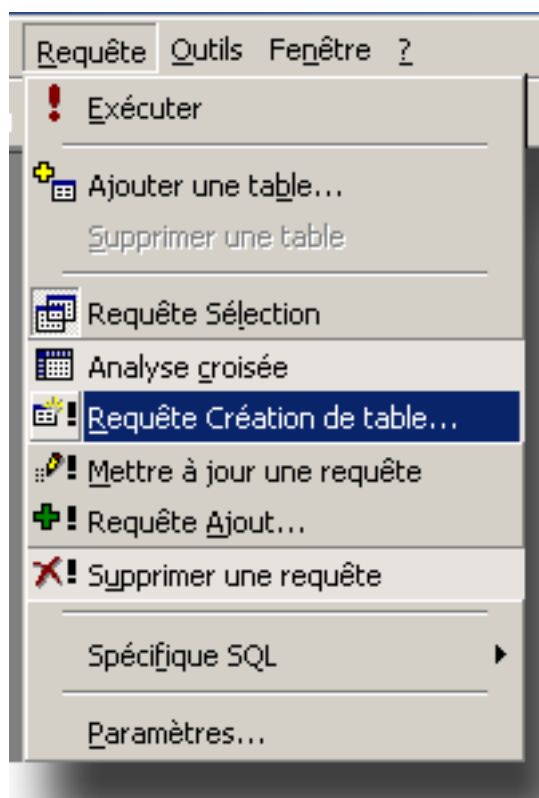
4.1 - Les objets requêtes

Pour créer la première requête, procédez de la même manière que pour la création d'une requête de type Select, suivant le modèle présenté ci-dessous.




La requête Select à transformer.

Cette requête permet de sélectionner les enregistrements à traiter dans la requête d'analyse croisée. A l'aide du menu *Requête* transformez-la en *Requête Création de table*...



Un menu précieux.

Sauvegardez la requête sous le nom suivant : **Gen_Temp_AC**

 En Choissant un nom explicite lors d'un développement multi-utilisateur, indiquant qu'il s'agit d'un objet temporaire, vous pourrez déterminer rapidement les tables à garder dans l'ihm et celles à inclure dans le fichier de données partagé.

La requête finale.

Sauvegardez celle-ci sous **Analyse Résultat par An**

Nos objets sont prêt :

Liste des objets	
Gen_Temp	Permet de copier les enregistrements de Vente à analyser.
Temp_AC	Table créée avec les enregistrements issues de Vente .
Analyse Résultat par An	Requête d'analyse croisée finale basée sur Temp_AC .

4.2 - La procédure Visual Basic

Bien que cette procédure puisse être réalisée dans une macro, nous allons plutôt la réaliser en VBA. Passez le formulaire **Analyse Croisée** en mode création. Créer un bouton de commande que vous nommez **Btn_FullQueries** et qui porte *Analyse Croisée Full Queries* comme légende. Sur l'évènement **Sur Clic** sélectionnez [**Procédure événementielle**] puis cliquez sur le bouton pour créer une procédure VBA.


Code à insérer dans la procédure Private Sub Btn_FullQueries_Click()

```
DoCmd.SetWarnings False           ' message off
DoCmd.DeleteObject acTable, "Temp_AC" ' supprime la table temporaire
DoCmd.OpenQuery "Gen_Temp_AC", acViewNormal ' création de la table temporaire
DoCmd.OpenQuery "Analyse Croisée", acViewNormal ' affiche l'analyse croisée
docmd.setWarnings True           ' message on
```

La méthode à base de requête est maintenant terminée. En sélectionnant le critère dans la zone liste et en cliquant sur le bouton vous aurez une belle requête d'analyse croisée basée sur le critère de la liste.

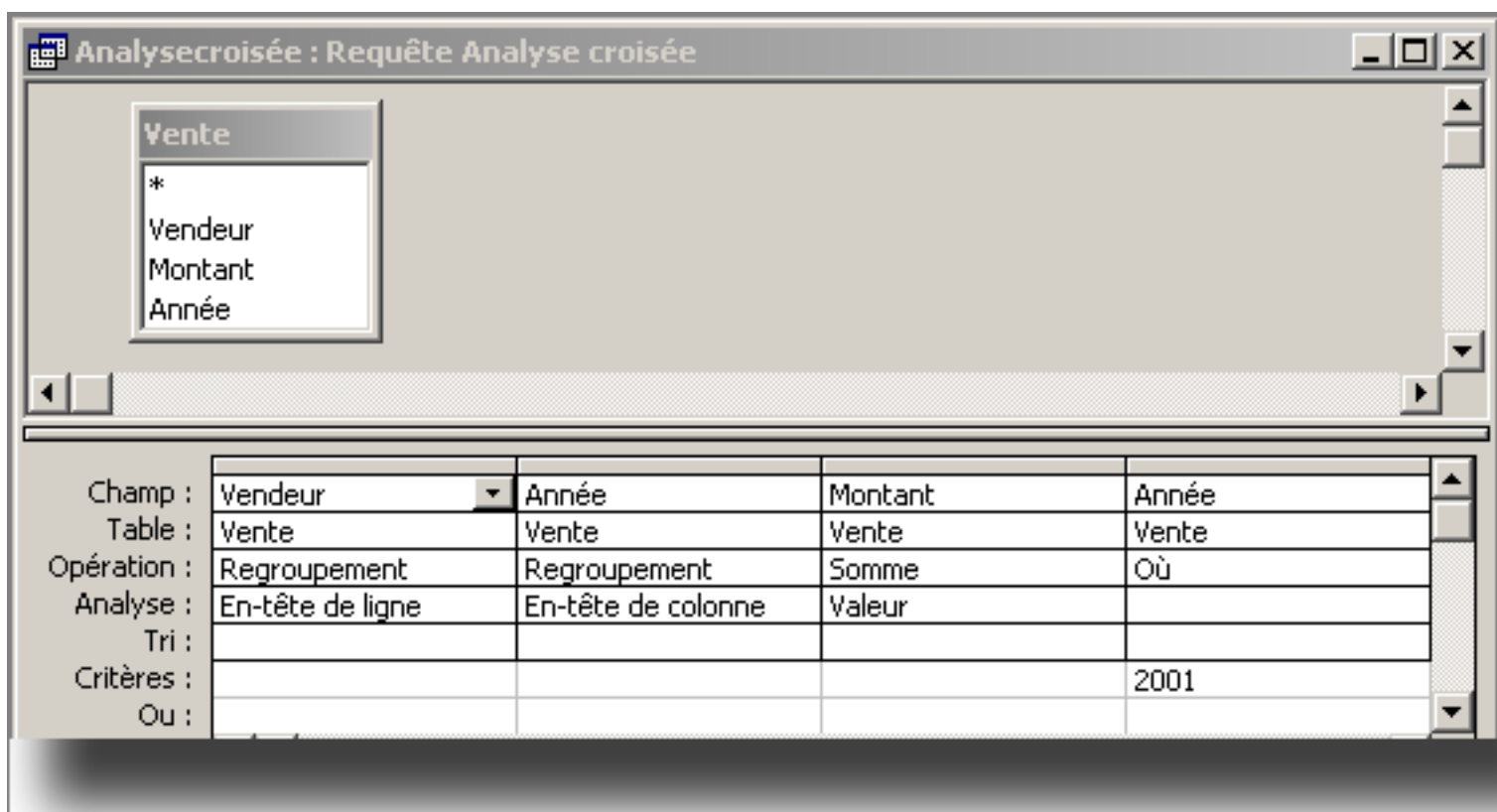
5 - Méthode Visual Basic pour Application

La méthode à base de Visual Basic pour Application est beaucoup plus simple à mettre en oeuvre pour un peu qu'on connaisse le langage VBA et les requêtes.

 Il est conseillé de créer une requête d'analyse croisée à l'aide du générateur, cela évite les erreurs potentielles.

5.1 - Création de la requête

Créez la requête d'analyse croisée identique à la première réalisée.



Tiens !? un air de déjà vu...

Passez en mode SQL à l'aide du bouton liste de la barre d'outils requête.

Le code SQL apparaît faites en une copie à l'aide de la combinaison de touches *Ctrl+C*. Enregistrez la requête sous le nom **AnalyseAnnée**.

5.2 - Création de la procédure Visual Basic

Dans le formulaire créez un deuxième bouton que vous nommez **Btn_FullVBA** avec la légende *Analyse Croisée Full VBA*. Créez de la même manière que pour le premier bouton une procédure VBA sur l'évènement **Sur Clic** Insérez la chaîne SQL précédemment copiée.

La chaîne SQL insérée.

```
TRANSFORM Sum(Vente.Montant) AS SommeDeMontant
SELECT Vente.Vendeur
FROM Vente
```

La chaîne SQL insérée.

```
WHERE ((Vente.Année)=2001))
GROUP BY Vente.Vendeur
PIVOT Vente.Année;
```

Modifiez le code comme suit en respectant bien les espaces après les double-côtes

Code modifié

```
Dim Sql As String ' declare la variable Sql

Sql = "TRANSFORM Sum(Vente.Montant) AS SommeDeMontant"
Sql = Sql & " SELECT Vente.Vendeur"
Sql = Sql & " FROM Vente"
Sql = Sql & " WHERE ((Vente.Année) = 2001))"
Sql = Sql & " GROUP BY Vente.Vendeur"
Sql = Sql & " PIVOT Vente.Année;" ' compose la chaîne SQL

DoCmd.DeleteObject acQuery, "AnalyseAnnée" ' supprime la requête AnalyseAnnée

CurrentDb.CreateQueryDef "AnalyseAnnée", Sql ' crée la nouvelle requete
DoCmd.OpenQuery "AnalyseAnnée" ' ouvre la requete
```

Dernière modification du code et la plus importante ; elle concerne le remplacement du critère statique (2001) par le critère dynamique Me.ZI_critere.

Cette instruction fait référence à la valeur de la zone liste **ZI_critere** du formulaire courant (**me**).

Elle peut également être écrite ; **Me.ZI_critere.value** ou bien **ZI_critere.value**.

Dernière modification à effectuer.

```
'Remplacez cette ligne
Sql = Sql & " WHERE ((Vente.Année) = 2001))"

'Par celle-ci
Sql = Sql & " WHERE ((Vente.Année) = " & Me.ZI_critere & ")")"
```

Notez que l'on peut également modifier le code SQL d'une requête en faisant référence à la propriété SQL de l'objet DAO.QueryDef dont un excellent **exemple** est présent dans la FAQ.

6 - Conclusion

Si la première méthode, "Full Queries", est réalisable sans aucune connaissance du VBA, elle pose quelques problèmes importants.

Avantages	Inconvénients
Peut être mise en oeuvre par un réfractaire au VBA.	Nécessite l'exécution de 2 requêtes (sélection et analyse croisée) qui peut être rédhibitoire sur un gros volumes de données.
Traçage d'erreurs plus simple puisque ne faisant appel qu'au générateur de requêtes.	Implique la création d'une table temporaire sur le poste client, toujours contraignante en terme d'occupation disque.
Je n'en vois pas d'autres...	Nécessite l'intervention sur 2 objets requêtes en cas de modification.

La deuxième méthode, "VBA", est de loin ma préférée.

Avantages	Inconvénients
Exécution d'une seule requête permettant un gain de temps.	Ne peut être mise en oeuvre par un réfractaire au VBA.
Pas de création de table temporaire.	Traçage d'erreurs nécessitant l'intervention à 2 niveaux ; VBA et Requête.
Modification plus simple ; remplacement de la requête dans le code.	Je n'en vois pas d'autres...

Si le traçage des erreurs dans le code SQL est pour vous un gros problème sachez que les points d'arrêts (**F9**) ainsi que le passage en mode création de la requête finale vous permettra de la déboguer par le générateur de requête.

7 - Remerciements

Je tiens à remercier : **GreyBird**, **Anomaly**, **Loulou24**, **srjd**, **ChristopheJ**, **FRED.G** et **Super.Nono40** pour les précieuses informations dispensées dans l'utilisation de l'éditeur XML et les différents outils et procédures de Developpez.com.

Maxence HUBICHE pour son accueil et ses conseils avisés ainsi que l'équipe du forum ACCESS.

A **FRED.G** (encore et toujours) pour la pertinence de ses corrections, sous oublier les relecteurs de l'ombre.

A l'équipe de **Developpez.com** pour la qualité du site.

Je présente mes plus plates excuses à ceux que j'aurais omis de remercier.

"Access ne nous rend que ce qu'on lui donne."