

Ch1. Nouveautés déjà intégrées...

1. Table PARAMETRES pour l'application

Il existe pour une application des **valeurs de référence** à stocker dans **une table** avec 1 seul enregistrement à plusieurs champs. Cette table n'est pas reliée aux autres tables.

Exemple : Cours de l'€ en \$, (nombre de points de base attribués à tous les représentants pour une visite effectuée à un client...)

- Ces valeurs ne sont **pas constantes** donc on n'écrit pas (valeur en \$= valeur en € * 1.21). On souhaite pouvoir **modifier** le cours de l'€ en \$ donc on range le cours de l'€ dans un **champ** de table et l'on souhaite pouvoir **l'atteindre** dans l'application.

- Ces valeurs ne concernent **pas un prix particulier** (prix du produit x, budget de l'équipe y,...) mais sont **communes** à toutes les valorisations de l'application donc elles sont stockées en **un seul exemplaire** (1 seul enregistrement) et ne sont **pas reliées** à tel ou tel élément.

1) Enregistrer des paramètres

Donc création de la **Tables Paramètres** avec 1 seul enregistrement et autant de champs que de paramètres (en vrac) à enregistrer:

PARAMETRES : Table			
Num paramètre	Cours Euro en Dollar	Cours Euro en Franc	BasePoints/Visite
1	1,21	6,559	3
(NuméroAuto)	0	0	0

2) Modifier des paramètres

La modification peut se faire directement dans la Table ou par un Formulaire.

3) Utiliser des paramètres

Pour accéder aux paramètres dans un formulaire, il faut que celui-ci ait pour Source une requête contenant les champs des paramètres à utiliser.

Remarque: la table Paramètres dans la requête n'est pas liée aux autres tables de la requête.

Remarque : *PARAMETRES.** dans la requête signifie tous les champs de la table.

Voir : depuis le menu, 'Représentant et ses rôles d'entraîneur' puis le sous formulaire et sa requête en source : '**R Représentant et PARAMETRES**' pour ce résultat $=[\text{BudgetEuro}] * [\text{Cours Euro en Dollar}]$

Equipe sportive.*	PARAMETRES.*	
Equipe sportive	PARAMETRES	

Utilisation de la Table PARAMETRES...

Budget total € : 25 000,00 €

Conversion en Francs : 163 975,00 F


Conversion en Dollars : 30 250,00 \$

2. Images dans des Formulaires ou Etats

1) Images de 'Décoration' non stockées dans une table

Exemple : Formulaire **Produit** dans l'En-tête

Il s'agit de gérer une **image fixe** pour **tous les enregistrements** d'un formulaire.

Cette image est **définie dans un contrôle** de type  **'Image'**. à gérer en mode assistant contrôle. Ensuite, définir la propriété *Mode affichage* : **Zoom** (pour remplir au mieux le cadre).




2) Images de 'Données' stockées dans un champ de table

Exemple : Formulaire **Produit** dans **Détail**

Il s'agit de gérer une **image changeant** à **chaque enregistrement** d'un formulaire.

Cette image est **enregistrée dans un champ** de type **Objet OLE** d'une table. (table Produit, champ Photo).



Cette image est affichée par **un contrôle** du formulaire de type  **'Cadre d'objet dépendant'** (dépendant de chaque enregistrement). La source du contrôle est le champ **Photo** de la table. Gérer la propriété *Mode affichage* : **Zoom**

La saisie peut se faire dans la table ou dans le formulaire, par clic droit sur la zone d'image puis soit insérer un objet à partir d'un fichier .BMP, soit par Coller.

Seul le format .BMP peut être utilisé et les fichiers .JPG (compressés) ne peuvent être gérés d'où un problème de taille de l'application.

Attention : 8 images .BMP de 500 Ko chacune = 4 Go de plus dans l'application !!!

Remarque : Pour minimiser la taille de l'application Access, on peut lier les images (donner l'adresse du fichier-image sur le disque C:\MesImages\image1.bmp) au lieu d'incorporer les images dans la base.

Attention dans ce cas si vous déplacez les images sur un autre disque avec un chemin de dossiers différent (C:\MesDocuments\Catalogue\image1.bmp)...

3. Champ de type Mémo pour du texte long

Le champ de type **Texte** peut enregistrer 255 caractères au plus.

Le champ de type **Mémo** peut enregistrer 65 535 caractères au plus.

Exemple : table **Produit**, champ **Commentaires libres**.

Un contrôle de type **'Zone de texte'** du formulaire Produit affiche le texte long.

4. Champ de type Lien hypertexte pour accéder à...

Le champ de type **Lien hypertexte** peut enregistrer **un lien** vers:

une page web, un fichier sur disque, une adresse mail, un objet de la base de données.

Exemple : table **Produit**, champ **LienWeb**.

Un contrôle de type **'Zone de texte'** du formulaire Produit affiche le lien.

Remarque : **Liens sur la formation à Access** présents sur le Formulaire **'Menu'**.
(liens non stockés dans une table)

Ch2. Travail à faire

1. Saisie par liste déroulante sans table-liste

Objectif

Saisir la **VILLE** des Client depuis une liste déroulante définie par les **villes déjà saisies** dans le formulaire **Client**.

Si la ville n'est pas présente dans la liste, il faut la saisir au clavier. La nouvelle ville saisie au clavier doit immédiatement être disponible dans la liste déroulante.

Procédure

1) Créer une liste Déroulante (LD)

a) Créer une requête avec :

Champ : *Ville* Table : *Client* Opération : *Regroupement* Tri : *Croissant*
Exécuter, Enregistrer sous : **R Liste des villes pour LD** et Fermer

b) Remplacer le contrôle zone de texte *Ville* du *Client* par un contrôle Liste Déroulante ajouté en mode *Assistant-contrôle* avec :

Rechercher les valeurs **dans une table ou requête**

Requête ; **R Liste des villes pour LD**

Sélectionner **Ville** pour Liste déroulante

Gérer la largeur de la liste

Stocker la valeur de la liste déroulante dans le champ **Ville**

Etiquette : *Ville* Terminer

Nommer ce contrôle : **LD Ville** et gérer la mise en page du formulaire si besoin.

c) Exécuter

1) modification d'une ville d'un Client par saisie dans la liste déroulante
2) puis modification, pour ce même Client, de sa ville par saisie directe au clavier d'une nouvelle ville (exemple DAX) car cette ville est non présente dans la liste déroulante

3) puis, pour un autre Client, impossibilité de retrouver dans la liste déroulante la ville de DAX saisie au clavier précédemment = Problème à gérer en actualisant la liste à partir de son nom (**LD Ville**).

2) Actualiser les valeurs d'une liste déroulante

Cette Actualisation sera faite dès que l'on **Entre** (clique) dans la liste déroulante.

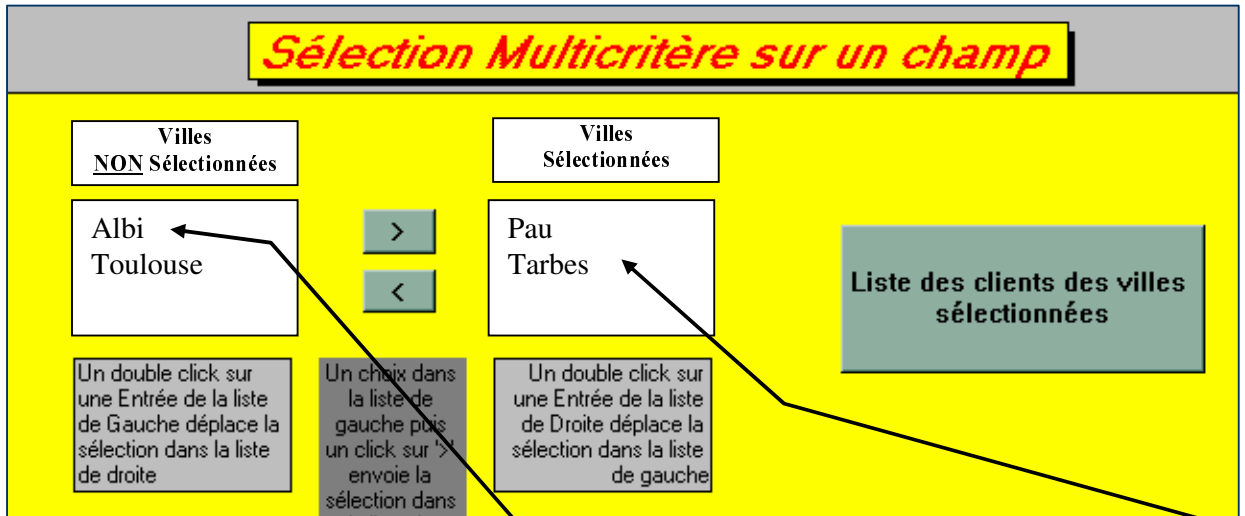
Créer puis Associer la Macro **Formulaire client, Ville, Actualiser LD** à l'événement **Sur Entrée** (le curseur entre dans le contrôle)

Actualiser

Nom contrôle : **LD Ville**

2. Contrôle 'Zone de liste' pour sélection multiple gérée avec des requêtes MiseAJour ou en VBA

A. Le formulaire



B. La Table de sélection

C. Chronologie d'exécution...

Formulaire Sélection ville



Ville pour sélectionner	Choix
Albi	Non
Pau	Oui
Tarbes	Oui
Toulouse	Non

Ville pour sélectionner	Choix
Albi	Non
Pau	Non
Tarbes	Non
Toulouse	Non

1) Macro 'BT sélection N,désélectionner TOUTES'

Objectif : Désélectionner toutes les villes et ouvrir le formulaire de sélection des villes

Liée à : Contrôle **BT selection N** du Formulaire **Sélection ville**

Événement : **Sur clic**

Contenu :

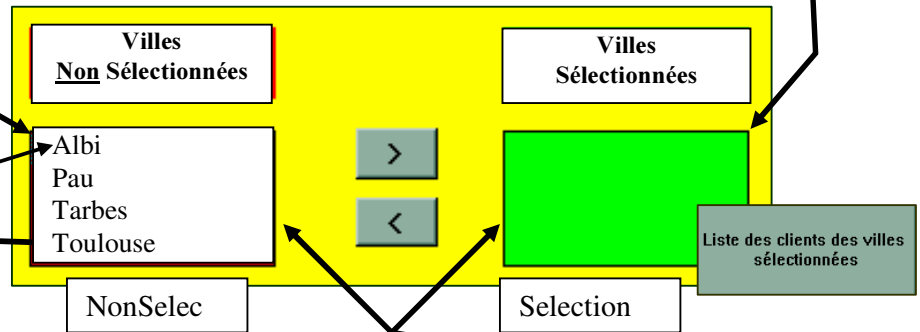
Avertissements	Avertissement actif: Oui	
OuvrirRequête	R ville 1 MAJ désélectionner toutes choix<=non	
Avertissements	Avertissement actif: Non	
OuvrirFormulaire	Selection villes multiples par Requetes MAJ	

Remarque : on peut paramétrer Access pour les confirmations d'opération : dans Outils/Options/ModifierRechercher **décocher** Confirmer : Modification/Suppression/ Requête action.

Champ:	Ville pour sélectionner	Choix
Table:	Ville pour sélection	Ville pour sélection
Tri:	Croissant	
Afficher:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Critères:		Non

Champ:	Ville pour sélectionner	Choix
Table:	Ville pour sélection	Ville pour sélection
Tri:	Croissant	
Afficher:	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Critères:		Oui

Propriétés	
Nom	NonSelec
Source contrôle	
Origine source	Table/requête
Contenu	R ville 4 Afficher les désélectionnées: choix=non



2) Macro 'ZL Non sélectionnéE, sélectionnER 1'

Objectif : Affecter **OUI** à **Choix** dans la table 'Ville pour sélection' pour la ville choisie et actualiser l'affichage des 2 contrôles Zone de liste (2 listes de villes).

Liée à : Contrôle **NonSelec** du Formulaire **Selection villes multiples par Requetes MAJ** et de même avec **>**

Événement : **Sur double-clic** et de même avec **CLIC** pour **>**

Contenu :


OuvrirRequête	R ville 2 MAJ sélectionner	: choix<=oui
Actualiser	Nom contrôle: NonSelec	
Actualiser	Nom contrôle: Selection	


Champ:	Choix	Ville pour sélectionner
Table:	Ville pour sélection	Ville pour sélection
Mise à jour:	Oui	
Critères:		[Formulaires]![selection villes multiples par Requetes MAJ]![NonSelec]

Ville pour sélection : Table		
	Ville pour sélectionner	Choix
	Albi	Oui
	Pau	Non
	Tarbes	Non
	Toulouse	Non

3) Macro 'ZL SélectionnéE, DésélectionnER 1'

Objectif : Affecter **NON** à **Choix** dans la table 'Ville pour sélection' pour la ville choisie et actualiser l'affichage des 2 contrôles

Liée à : Contrôle **Selection** du Formulaire **selection villes multiples par Requetes MAJ** et de même avec 

Evénement : **Sur double-clic** et de même avec **CLIC** pour 

Contenu :

OuvrirRequête	R ville 3 MAJ désélectionner	: choix<=non
Actualiser	Nom contrôle: NonSelec	
Actualiser	Nom contrôle: Selection	

4) Macro 'BT Liste des clients des villes, Ouvrir Formulaire'

Objectif : Ouvrir un formulaire Client basé sur une requête qui tient compte des villes choisies (Choix=oui) dans la table 'Ville pour sélection'

Liée à : Contrôle Bouton 'BT Liste des clients' du Formulaire **Selection villes multiples par Requetes MAJ**

Evénement : **Sur clic**

Contenu :

OuvrirFormulaire	Clients sélectionnés par villes multiples
------------------	---



The image shows a screenshot of a database application interface. At the top, a yellow banner reads "Sélection de clients par villes". Below it is a table with columns "Nom du client:", "Prénom:", and "Ville:". The table contains four rows of client data, with the "Ville" column highlighted in green. Below this is a screenshot of a query window titled "R ville 6 Source du formulaire final : choix=oui : Requête". The query window shows a table structure with columns "Nom du client", "Prénom", "Ville", and "Choix". A red arrow points from the "Choix" column in the query window to a table structure window titled "Ville pour sélection : Table". This table has columns "Ville pour sélectionner" and "Choix", with rows for "Albi", "Pau", "Tarbes", and "Toulouse". A red box highlights a note: "Lien tiré au niveau de la requête. Pas de 1 et ∞ ici. Remarque: on peut, dans une requête, lier des tables et requêtes et même requêtes et requêtes."

D. Les Fonctions en VBA : modifier 'Table pour Liste Ville'

Même travail que l'exemple précédemment avec du **code VBA** pour sélectionner une ville NonSésélectionnée.

Voici un exemple d'une Fonction VBA ' **NSclick()** ' effectuant le même traitement que la *macro* 'ZL Non sélectionnéE, sélectionnER 1' et sa Requête de mise à jour.

Vous n'avez rien à faire ici...

```
Private Function NSclick()
'en cas d'erreur, aller à l'étiquette error4 (en bas de fonction)
On Error GoTo error4

'Me = ce Formulaire lui-MEme
'column(0) = première colonne du contenu de la zone..
'de liste [Nonselect]=Liste des villes
'Monchoix$=la ville sélectionnée dans la colonne gauche [Nonselec]
Monchoix$ = Me! [NonSelec] .Column(0)

'Dynaset=type de résultat d'une requête pouvant être mis à jour
'ds est la variable de type Dynaset
Dim ds As Dynaset, db As Database

'db est la variable correspondant à la base de données courante
Set db = CurrentDb()

'Recherche de l'enregistrement de [Table pour liste ville]...
'choisi par(Monchoix$) par une requête SQL Sélection (where)
Set ds = db.CreateDynaset("select * from [Table pour Liste Ville]
where [Ville pour sélection] =" + Monchoix$ + "';")

'édition pour pouvoir modifier
ds.Edit
'modification à OUI du champ Choix de la table
ds.[choix] = True
'mis a jour du dynaset<=>[Table pour liste ville]
ds.Update
'fermeture du dynaset
ds.Close

'Actualisation (Requery) des 2 listes de villes en fonction
'du nouveau contenu de [Table pour liste ville]
'pour nouvel affichage des 2 listes de villes.
'DoCmd pour exécuter la Commande d'actualisation Requery
DoCmd.Requery "Nonselec"
DoCmd.Requery "Selection"

Exit Function 'quitter la fonction


error4: 'c'est une étiquette atteinte en cas d'erreur
MsgBox "Vous devez cliquer sur un nom!", 64, "Attention..."
Exit Function
Resume Next
End Function
```


3. Macros OuvrirFormulaire avec ConditionWhere

Ajouter une macro au groupe de macros: **Groupe Facture**

1) Macro FactureSF, Bouton lunettes, Ouvrir produit correspondant

Objectif : Depuis le sous-formulaire Facture, Ouvrir le formulaire **Produit** en se positionnant automatiquement sur le produit correspondant à celui de la ligne courante de la facture. C'est le contenu du contrôle de la liste déroulante **Ref produit** (CLA001 par exemple) qui sera le critère de recherche.

Liée à : Contrôle Bouton de commande  (à ajouter sans assistant contrôle) du Sous-Formulaire **Facture**

Evénement : **Sur double-clic**

BoîteMsg	Message: Détail du produit sélectionné...	Champ de la Table du formulaire à ouvrir
OuvrirFormulaire	Nom formulaire: Produit Condition where: [Produit]![Num produit] = [Produit]![Num produit] Formulaires![Facture et ses produits 2 FP]![Facture et ses produits 2 SF].Form![Ref produit]	

Utiliser le générateur d'expression →

Contrôle du Formulaire de départ

Attention à la syntaxe à partir du formulaire FP, puis le contrôle du FP, puis le formulaire du contrôle, puis LE contrôle...

4. Des Fonctions pour des Requêtes et Formulaires

Dans des requêtes, nous avons déjà défini des **Calculs** basés sur des contrôles, champs et opérateurs arithmétiques en utilisant le générateur d'expression.

Total du produit: [Quantité][Prix HT]*(1+Produit![Taux TVA])*

Dans des requêtes, nous avons déjà utilisé la puissance de la ligne **Opération** permettant de gérer des **Regroupements** et de faire des calculs (**Somme, Compte, Moyenne, Min, Max, Premier, Dernier, ...**) sur les éléments regroupés et triés.

Champ :	Catégorie: Gauche([Facture détail]![Ref produit];	Total HT: [Produit]![Prix HT]*[Facture détail]![Quantité]
Table :		
Opération :	Regroupement	Somme

Nous allons maintenant ajouter dans des requêtes complexes l'utilisation des **Fonctions Visual Basic prédéfinies** dans le Langage **VBA**.

Les Fonctions VBA ont un nom Anglais et Français.

Le nom Anglais est reconnu partout alors que le nom Français est reconnu dans les Requêtes, Formulaires, Etats et Macros **mais pas** dans les Modules VBA et **pas dans l'Aide en ligne**.

- Pour rechercher une fonction dans l'aide : Ouvrez le menu ? / **Aide sur Microsoft Access** puis dans l'onglet **Sommaire / Programmer en Visual Basic / Référence du langage Visual basic / Fonctions** (par ordre alphabétique anglais) ou bien à la place de Fonction **Groupe** ou **Index/liste** pour un classement thématique des fonctions.

Comme nous n'utilisons pas les modules, nous utiliserons le nom Français des Fonctions, c'est aussi celui que propose le Générateur d'expression.

Quelques fonctions VBA...

Nom Français	Nom Anglais	Rôle
<i>Mathématique</i>		
Round (exp ; n)	Round ()	Arrondit une expression à n décimales Round(10,129 ; 2) donne 10,13
Ent (exp)	Int ()	Partie entière sans arrondir Ent(10/3) donne 3
Aléat ()	Rnd ()	Nombre aléatoire compris entre [0 et 1[Aléat() donne par exemple 0,658975546812358
<i>Date/heure</i>		
Date ()	Date ()	Date du jour de l'horloge de l'ordinateur Date() donne 22/12/04
Temps ()	Time ()	Heure courante Temps() donne 09:12:59
Maintenant ()	Now ()	Date et heure courante
<i>Portion de temps</i>		
Jour (date)	Day ()	Jour du mois (entre 1 et 31)
Mois (date)	Month ()	Mois de la date (entre 1 et 12)
Année (date)	Year ()	Année de la date
Heure (heure)	Hour ()	Heure d'une heure complète: Heure(12:59:30) = 12
Minute (heure)	Minute ()	Minute d'une heure complète: Minute(12:59:30) = 59
Seconde (heure)	Second ()	Seconde d'une heure: Seconde(12:59:30) = 30
JourSem (date, 2)	Weekday ()	Jour/semaine (entre 1 et 7), 7 = Dimanche en France
<i>Caractère</i>		
Gauche (exp ; n)	Left ()	Extrait n caractères à gauche d'une chaîne Gauche("abcdef"; 3) donne "abc"
Droite (exp ; n)	Right ()	Même chose à partir de la droite
<i>Divers</i>		
EstNull (exp)	IsNull ()	Vrai si exp est Null (Vide) sinon Faux
Vraifaux (condition ; valeurSiVrai ; valeurSiFaux)	Iif ()	Equivalent au Si(condition ; alors ; sinon) d'Excel VraiFaux (a=1 Et b=2 ; 5 ; 6) donne 5 si (a=1 et b=2) donne 6 sinon

Utiliser des Fonctions dans des Requêtes.

Objectif: Effectuer des calculs concernant les visites des représentants.

Règle: Chaque visite est valorisée par un certain nombre de points quel que soit le représentant. Ce nombre de points est stocké dans la table **PARAMETRES** dans le champ **BasePointsVisite**. **Si** la durée de la visite ≥ 2 heures **et** qu'elle n'a pas lieu un '*Dimanche*', **Alors** le nombre de point calculé est le nombre d'heures de la visite x **BasePointsVisite** x 2 **Sinon** **BasePointsVisite** (**seulement**).

On souhaite gérer le Total des points et compter le Nombre de visites regroupées par **Représentant**, par **An**, par **Mois** en utilisant des Fonctions VBA.

Nom représentant	AnVisite	MoisVisite	Total Points	Nombre Visites
LESPOURCY	2003	11	3	1
LESPOURCY	2003	12	24	2
GUILLAUME	2003	3	12	1
GUILLAUME	2003	10	15	2
GUILLAUME	2003	12	3	1

Consulter la table **PARAMETRES** pour voir la valeur de **BasePointsVisite (3)**.

a) Créer une requête basée sur les tables Visite et PARAMETRES qui sera la Source du futur Formulaire (voir en bas de page).

Ces 2 tables ne sont pas liées, c'est normal. Gérer 5 colonnes pour cette requête.

Les 3 premières colonnes seront la base du **Regroupement** et du **Tri**:

Ref représentant	AnVisite: Année([Date visite])	MoisVisite: Mois([Date visite])
Visite		
Regroupement	Regroupement	Regroupement
Croissant	Croissant	Croissant

La quatrième colonne calcule le nombre de points de chaque visite.

Elle utilise quelques Fonctions présentées ci-dessus.

Total points : VraiFaux (SI
Heure ([Durée visite]) >=2 Et JourSem ([Date visite];2) < >7 ;	Condition double
Heure ([Durée visite]) * [BasePointsVisite] * 2 ;	Alors
[BasePointsVisite])	Sinon

Les calculs **Somme** et **Compte** pour la quatrième et la cinquième colonne sont effectués sur les enregistrements **Regroupés** par les 3 premières colonnes.

Total Points: VraiFaux (Heure([Durée visi	Nombre Visites: "*"	On compte les lignes d'une colonne quelconque
Somme	Compte	

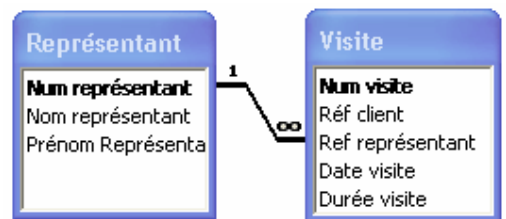
b) Créer un Formulaire

Maintenant que les résultats bruts sont obtenus par la **Requête**, nous allons présenter dans un **Formulaire** le bilan des visites et des points par **Représentant**, **An** et **Mois**.

Créer un Formulaire sur la table **Représentant** et sur la **requête précédente** sans le champ **Ref représentant** dans le sous-formulaire (inutile sinon affiche Lespourcy).

L'assistant formulaire détecte la possibilité de faire un FP et un SF entre les 2 champs **Num représentant** de la table **Représentant** et **Ref représentant** de la **Requête** en se référant à la fenêtre **Relation**.

Num représentant	2	Total général bis:	27
Nom représentant	LESPOURCY	Gérard	
Ventilation des visites par An et par Mois			
AnVisite	MoisVisite	Total Points	Nombre Visites
2003	11	3	1
2003	12	24	2
Total général:		27	



Ajouter dans le pied du SF un contrôle **Zone de texte** pour faire le Total général de tous les points du Représentant avec pour Source **=Somme([Total Points])**

5. Résumé des Macros

Rappel sur les **Actions** de Macros vues en TD:

Se déplacer dans les données : **TrouverSuivant**, **TrouverEnregistrement**,
AtteindreContrôle, **AtteindreEnregistrement**
Arrêter l'exécution : **AnnulerEvénement**, **Quitter**, **ArrêtToutesMacros**, **ArrêtMacro**
Déplacer ou redimensionner une fenêtre : **Agrandir**,
Ouvrir ou fermer un objet base de données : **Fermer**, **OuvrirFormulaire**, **OuvrirRequête**,
OuvrirEtat
Définir la valeur d'un champ, d'un contrôle ou d'une propriété : **DéfinirValeur**
Mettre à jour les données ou l'écran : **Actualiser**
Afficher des informations à l'écran : **BoîteMessage**, **Avertissements**
Emettre une tonalité : **Bip**

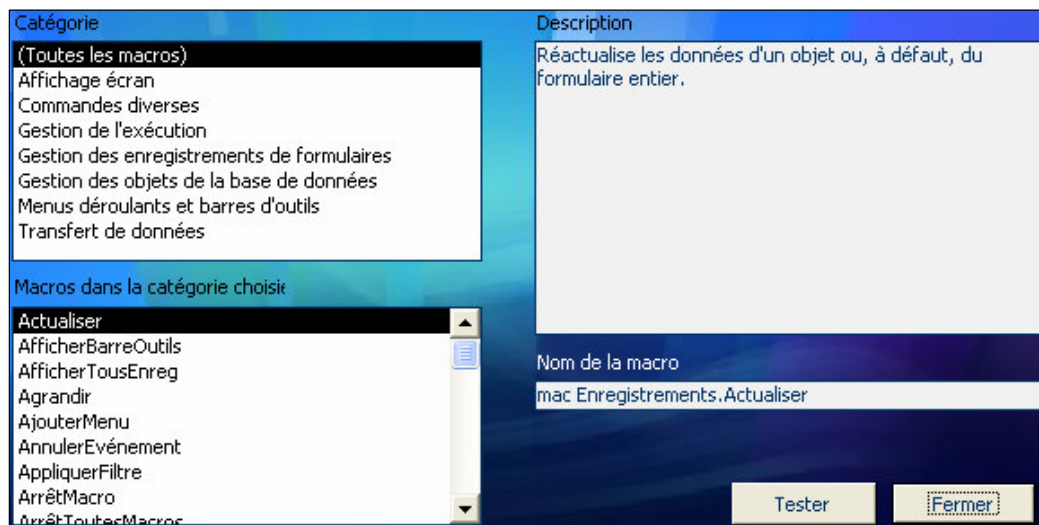
5 **Actions** intéressantes non vues en TD sont commentées ci-dessous

Restreindre les données : **AppliquerFiltre** : pour appliquer le filtre d'une requête à des données affichées
Exécuter : **ExécuterCommande** : action très puissante pour exécuter une commande parmi toutes celles disponibles par tous les menus standards = 500 commandes environ de Actualiser à Zoom en passant par **Annuler**, **Aperçu**, **AtteindreSuivant**, **Copier**, **Coller**, **Imprimer**, **Sauvegarder**, **Supprimer** ...
ExécuterMacro : exécuter un macro depuis une macro, intéressant pour créer une boucle grâce aux arguments *RépéterCompte* (= Pour) et *RépéterExpression* (= TantQue)
ExécuterApplication : exécuter une application Windows en donnant si besoin le chemin de dossiers qui y mène: C:\Program Files\Microsoft Office\WORD.EXE
Imprimer objet : **Imprimer** : imprimer l'objet en cours avec les options d'impression

Voir macro
.....

- Penser à faire des **groupes de macros** par thème (formulaire par exemple).
- Penser à choisir le bon **évènement** pour déclencher une macro
(voir Formulaire 'Produit TEST sur Evènements' de l'application des TD)
- Penser à la possibilité de gérer des **conditions** ou des **boucles** dans les macros.

L'application **DemonstrationActionsMacro.mdb** d'Hervé Inisan du livre "**Référence Access 2002**" illustre toutes les **Actions** de Macro par catégorie, avec un test sur une application et accès au code. **A voir sur le serveur avec les applications du cours.**



6. Autres traitements...

Formulaire *Représentant et ses Visites*

Nom concaténé au Prénom pour affichage seulement

LESPOURCY Gérard

1) Opérateur de concaténation : &

=SupprEspace([Nom représentant]) & " " & [Prénom Représentant]

2) Convertir Heure en Minutes

Fonction **Standard** dans **Access** : **Heure()**

=Heure([Durée visite])*60+Minute([Durée visite])

a) Dans un contrôle de formulaire

Nom FormDureeMn
 Source contrôle =Heure([Durée visite])*60+Minute([Durée visite])
 Format # ##0" Mn"

b) Dans une colonne de Requête

ReqDureeMn: Heure([Durée visite])*60+Minute([Durée visite])

Nom ReqDureeMn
 Source contrôle ReqDureeMn
 Format # ##0" Mn"

3) Total du Détail dans le Pied du SF.

a) Somme d'un Contrôle avec définition de calcul : Impossible

=Somme([FormDureeMn])

b) Somme d'un Contrôle avec calcul défini dans requête : Possible

=Somme([ReqDureeMn])

4) Convertir Minutes en heures

Fonction **Utilisateur** dans un **Module VBA** :

=DureeEnHeures([TotDuree])

```

Function DureeEnHeures(ByVal DureeMin As Integer) As String
'Role : Retourner une chaîne de caractères correspondant à une durée en Heures et Minutes
' à partir d'une Durée en minutes exemple : 65 Mn donne "1H05Mn"
'ENTREE : DureeMin : Durée en minutes (ex : 65)
Dim Heure As Integer, Minute As Integer
    Heure = DureeMin / 60
    Minute = DureeMin Mod 60 'Mod pour le reste de la division
    DureeEnHeures = Heure & "H" & Format(Minute, "00") & "Mn" 'Conversion implicite String
End Function
    
```

5) Utiliser les Fonctions VBA de Domaine

Formulaire **Statistiques 1 sur l'application**

Nombre de Joueurs Total:	=CpteDom("R";"Joueur")
Nombre d'Equipes Total:	=CpteDom("R";"Equipe sportive")
Nombre de joueurs "Bon":	=CpteDom("R";"R Joueurs bon")
Nombre de joueurs "Bon":	=CpteDom("R";"Joueur";"[Niveau sportif] = 'bon' ")
Choisir le Niveau sportif :	Indépendant
Nombre de joueurs du niveau sportif choisi :	=CpteDom("R";"Joueur";"[Niveau sportif] = '" & [ListeNiveau] & "'")
Date du premier client :	=MinDom("Date d'entrée";"Client")
Nom du premier client :	=RechDom("Nom";"Client";"[Date d'entrée] = [ControleDatePremier]")
Date du dernier client :	=MaxDom("Date d'entrée";"Client")
Total stock tous produits :	=SomDom("Qté stock";"Produit")
Total stock Clavier normal :	=SomDom("Qté stock";"Produit";"[Désignation] = 'Clavier normal' ")
Total stock Fournisseur 1 :	=SomDom("Qté stock";"Produit";"[Ref Fournisseur] = 1")
Choisir le Fournisseur :	Indépendant
Total stock Fournisseur choisi :	=VraiFaux(EstNull([ListeFournisseur]);0;SomDom("Qté stock";"Produit";"[Ref Fournisseur] = '" & [ListeFournisseur]))

Il n'y a pas de données Source (Table ou Requête) pour ce Formulaire.

Calculs basés sur des Fonctions de Domaine. Chaque fonction défini son Domaine (Table ou Requête) ainsi que condition de Filtrage.

6) Utiliser les Requetes pour les calculs

Formulaire **Statistiques 2 sur l'application**

7) Imprimer LA Facture courante

Voir formulaire Facture et ses produits

a) Créer un Etat basé sur la table **Facture** en tête et sur la requête **R Facture détail** avec un **regroupement** pour faire dans **Total Facture** la **Somme des lignes**. Nom : **Facture**

b) Créer un bouton dans le formulaire Facture et lui associer la macro suivante à ajouter dans le groupe de macro Facture.

Num facture	15		
Nom client	BCN		
Date d'échéance	03/01/2002		
Payé	Non		
Nom produit	Quantité	Prix HT	Total TTC
Souris	16	30,00 €	574,08 €
Comprendre Acces	25	25,00 €	659,38 €
Imprimante iet N&B	2	80,00 €	191,36 €
Ecran 19 pouces	4	150,00 €	717,60 €
Ecran 17 pouces	2	100,00 €	239,20 €
Clavier normal	4	25,00 €	119,60 €
Total Facture :			2 501,22 €

OuvrirEtat	<p>Nom état: Facture</p> <p>Affichage: Impression directe</p> <p>Condition where: [Facture en tête]![Num facture]=[Formulaires]![Facture et ses produits 2 FP]![Num facture]</p>
------------	---

Une mise en forme conditionnelle a été faite sur le contrôle *Total du produit*.

Afficher un fond Jaune si la Valeur est > 500

Sélectionner le contrôle de l'Etat puis menu **Format/ Mise en forme conditionnelle** puis définir la **Condition** et le **Format**.