

Access : Programmer avec Visual Basic Application (2)

Autant il est possible de créer des modules pour faire des opérations, il est surtout avantageux d'effectuer des opérations avec des tables ou sur des tables d'une base de données. C'est le but de ce complément.

§ 1 - La lecture séquentielle des tables

A - Cas « Xtron »

L'entreprise gère ses salariés à l'aide du SGBDR Access. La base de données « Xtron » est créée et la table des « Salariés » est la suivante :

Liste des salariés

Numéro	Nom du salarié	Prénom	Adresse	Code postal
132	BERTRAND	François	12 rue square Fleuri	60310
Commune	Embauche	Salaire brut	Code catégorie	
Lassigny	1978	12 000,00 F	1	

Numéro	Nom du salarié	Prénom	Adresse	Code postal
139	HUCHET	Franck	12 bd Sarrazin	60400
Commune	Embauche	Salaire brut	Code catégorie	
Noyon	1983	10 500,00 F	2	

Numéro	Nom du salarié	Prénom	Adresse	Code postal
142	LAMBERT	Adrien	20 rue de Savoie	60200
Commune	Embauche	Salaire brut	Code catégorie	
Compiègne	1981	9 900,00 F	3	

Le directeur du personnel, Mr Louis, souhaite attribuer un « Complément » aux salariés. Ce complément est calculée selon les règles de gestion suivantes :

Salaire Salarié	Prime
<= 10 000 Fr.	15 % du salaire brut
> 10 000 Fr. et <= 12 000 Fr.	10 % du salaire brut
> 12 000 Fr.	5 % du salaire brut

De plus, Mr Louis veut connaître le montant total des compléments pour le transmettre au directeur financier, Mr Grégoire.

Le problème est donc de :

- calculer le complément de chaque salarié,
- calculer le montant total des compléments.

Remarques :

- a) Le traitement à effectuer est le même pour tous les salariés.
- b) Il est nécessaire de lire toute la table pour ne pas oublier de salariés.
- c) On ne connaît pas à l'avance le nombre d'occurrences à traiter, le nombre de salariés n'étant pas limité.

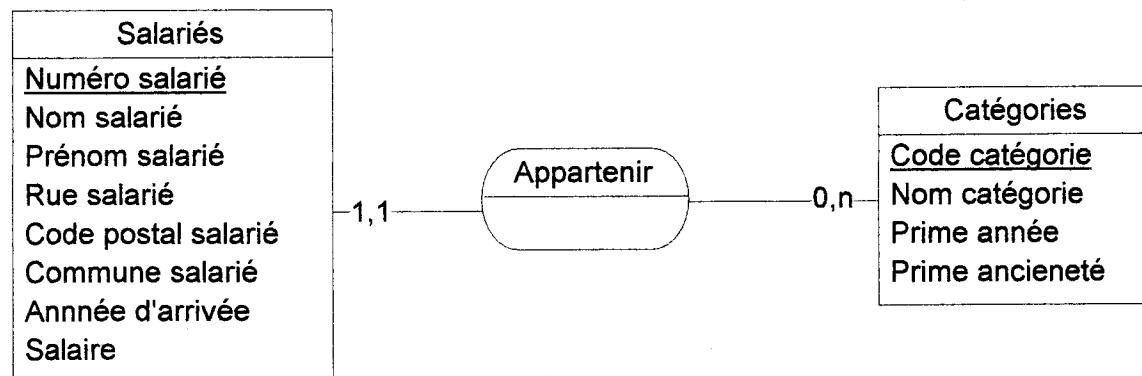
- d) La table peut-elle être vide, c'est-à-dire sans salariés (ce qui limite l'intérêt du calcul !).
- e) comment savoir que l'on a traité toutes les occurrences ? Il faut balayer toute la table, jusqu'à la fin.
- f) comment répéter le traitement ? Une boucle sera utile. Voir séquence 5.
- g) quel est le critère d'arrêt ? Comment arrête-t-on le traitement ? La fin de fichier (Salariés) est un bon critère d'arrêt. En effet, à la fin de ce fichier, tous les salariés auront été lus et leur complément calculé.
- h) quelles autres instructions sont utiles ? Y a-t-il un affichage ? Si oui, dans quelles conditions ? Un affichage n'est pas requis. Mais quel intérêt de faire des calculs si rien ne propose de le vérifier.
- i) comment allons-nous lire la table ? Va-t-on lire la table de 1^{er} enregistrement au dernier, en sélectionnant les enregistrements à traiter, au hasard ? Il est toujours possible de lire une table au hasard. Mais comment savoir si tous les enregistrements ont été lus. La meilleure solution reste la lecture séquentielle de la table du 1^{er} au dernier enregistrement.
- j) comment obtenir le total des primes ? Il s'agit d'un calcul.

B - Quelques définitions

Lecture séquentielle :
Il s'agit de lire une table et cela à partir du premier enregistrement. La lecture se déroulera jusqu'au point d'arrêt, par défaut, la fin de fichier.
Boucle :
Il s'agit de répéter un certain nombre d'opérations, d'instructions, un nombre fini de fois (voir séquences sur les algorithmes ainsi que le chapitre sur la programmation structurée.
Cumul :
Il s'agit de totaliser le contenu d'une variable à la somme des résultats précédents. La syntaxe est : Cumul $\leftarrow$ cumul + prime

C - Présentation de la base de données Xtron

La base de données Xtron permet de gérer les salariés de l'entreprise Xtron. La base de données Xtron a été conçue à partir du Schéma Conceptuel des Données suivant :



La base de données Xtron contient les tables suivantes :

- Catégories,
- Salariés.

D - Algorithme du calcul de la prime des salariés

DEBUT

**** Calcul de la prime des salariés ****

OUVRIR la base de données Xtron

OUVRIR la table des Salariés

CUMUL, Complément {réel}

LIRE le premier enregistrement de la table des Salariés

**** parcours de la table des Vendeurs et calcul de la mt_prime de chaque salarié ****

TANT QUE NON FIN DE TABLE Salariés FAIRE

SI [Salaire brut] <= 10000

ALORS

Complément = [Salaire brut] * 0.15

SINON

SI [Salaire brut] <= 12000

ALORS

Complément = [Salaire brut] * 0.1

SINON

Complément = [Salaire brut] * 0.05

FIN SI

FIN SI

CUMUL = CUMUL + Complément

Afficher le nom du salarié et son Complément

LIRE l'enregistrement suivant

FIN TANT QUE

AFFICHER CUMUL

FERMER la table des Salariés

FERMER la base de données Xtron

AFFICHER CUMUL

FIN

E - Programme Visual Basic Applications

```
Sub Complément()  
    ' Calcul du complément des salariés  
  
    Dim Xtron As Database  
    Set Xtron = CurrentDb  
    Dim Salarié As Recordset  
    Set Salarié = Xtron.OpenRecordset("Salarié")  
  
    Dim cumul, Complément As Integer  
  
    Salarié.MoveFirst  
    Do Until Salarié.EOF  
  
        If Salarié![Salaire brut] <= 10000 Then  
            Complément = Salarié![Salaire brut] * 0.15  
        Else  
            If Salarié![Salaire brut] <= 12000 Then  
                Complément = Salarié![Salaire brut] * 0.10  
            Else  
                Complément = Salarié![Salaire brut] * 0.05  
            End If  
        End If  
  
        cumul = cumul + Complément  
        Debug.Print "Montant du complément pour : " & Salarié![Nom-sal] & " " & Salarié![prenom_sal] & " -> "  
        & Complément  
  
        Salarié.MoveNext  
  
    Loop  
    Debug.Print "Total des Compléments : " & cumul  
  
    Salarié.Close  
    Xtron.Close  
  
End Sub
```

Remarque : Nous sommes en présence de 2 « SI » imbriqués. Visual Basic peut le traiter de le traiter de cette manière :

```
If Salarié![Salaire brut] <= 10000 Then  
    Complément = Salarié![Salaire brut] * 0.15  
ElseIf Salarié![Salaire brut] <= 12000 Then  
    Complément = Salarié![Salaire brut] * 0.10  
Else  
    Complément = Salarié![Salaire brut] * 0.05  
End If
```

La structure « Selon Cas » ... « Fin Selon » aurait pu être choisie. Tout est affaire de choix.

F - Instructions Access Basic nouvelles

Dim Xtron As Database
Déclare « Xtron » en tant que base de données
Set Xtron=CurrentDB()
Déclare « Xtron » comme base de données courante
Dim Salariés As Recordset
Déclare la table « Salariés » en tant que table de travail
Set Salariés = Xtron.OpenRecordset("Salariés")
Déclare la variable « salariés » en tant qu'enregistrement de la table « Salariés »
Salariés.MoveFirst
Demande au pointeur d'enregistrement de se rendre sur le 1 ^{er} enregistrement.
Salariés.EOF
Demande au pointeur de se rendre à la fin de fichier, c'est-à-dire le dernier enregistrement (EOF = End Of File)
Salariés.MoveNext
Demande au pointeur d'enregistrement de se rendre sur l'enregistrement suivant.
Debug.Print
Permet l'affichage dans la fenêtre d'exécution (voir « Access : programmer en VBA (1) »)
Salariés.close
Ordonne la fermeture de la table "Salariés".
Xtron.close
Ordonne la fermeture de la base de données "Xtron"

§ 2 - *Programme utilisant les valeurs affichées dans un formulaire*

Il s'agit d'afficher dans un formulaire un bouton qui donne le complément de salaire auquel a droit un salarié. Il est nécessaire de créer un bouton dans un formulaire permettant d'exécuter un programme.

Saisie des salariés

Numéro Salarié: 142
 Nom Salarié: LAMBERT
 Prénom Salarié: Adrien
 Adresse Salarié: 20 rue de Savoie
 Code postal Salarié: 60200
 Commune Salarié: Compiègne
 Année d'arrivée: 1981
 Code catégorie: 3
 Salaire brut Salarié: 9 900,00 F

Complément

Enr : 14 | 3 sur 14

Pour afficher dans une fenêtre le montant du complément de salaire accordé à chaque salarié.

Le résultat doit être le suivant :

Saisie des salariés

Numéro Salarié: 132
 Nom Salarié: BERTRAND
 Prénom Salarié: François
 Adresse Salarié: 12 rue square Fleuri
 Code postal Salarié: 60310
 Commune Salarié: Lassig
 Année d'arrivée:
 Code catégorie: 1
 Salaire brut Salarié: 12 000,00 F

Complément

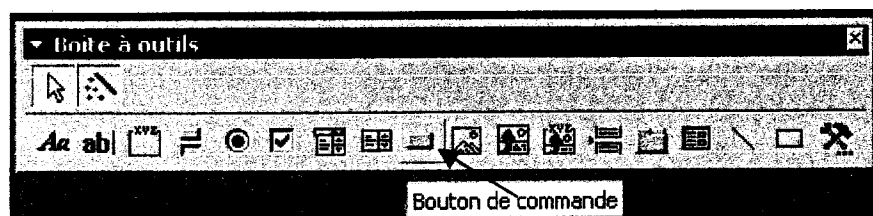
Enr : 1 | 1 sur 14

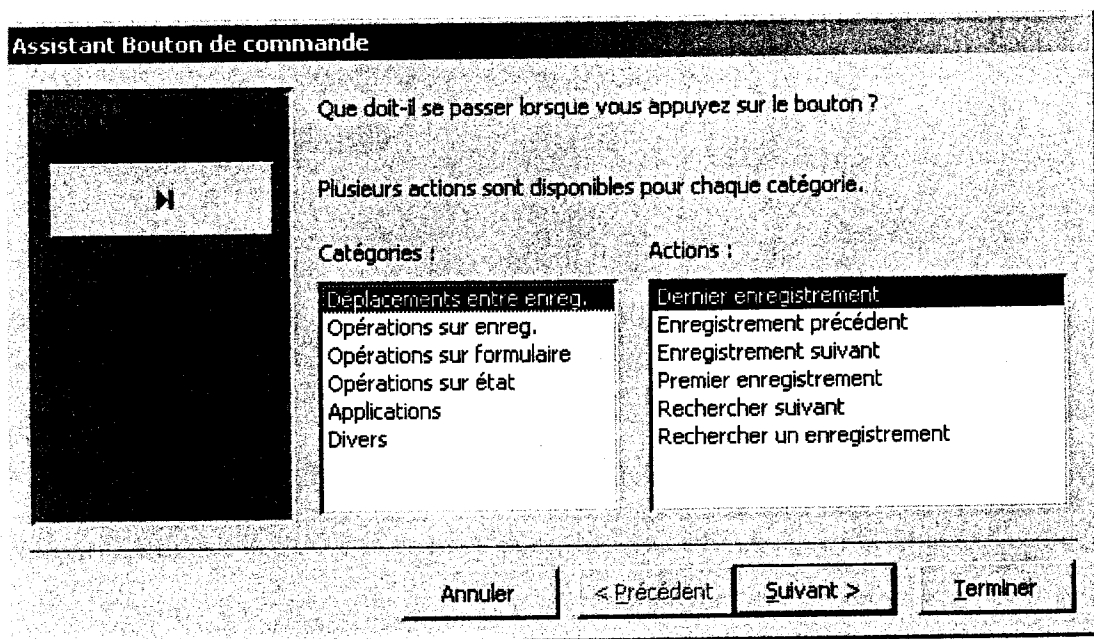
Microsoft Access

Le complément de salaire est de : 1200 Francs

OK

Pour créer un bouton dans un formulaire et y associer un événement pour exécuter le programme du calcul du complément, il faut choisir « Bouton de commande » dans la boîte à outils.

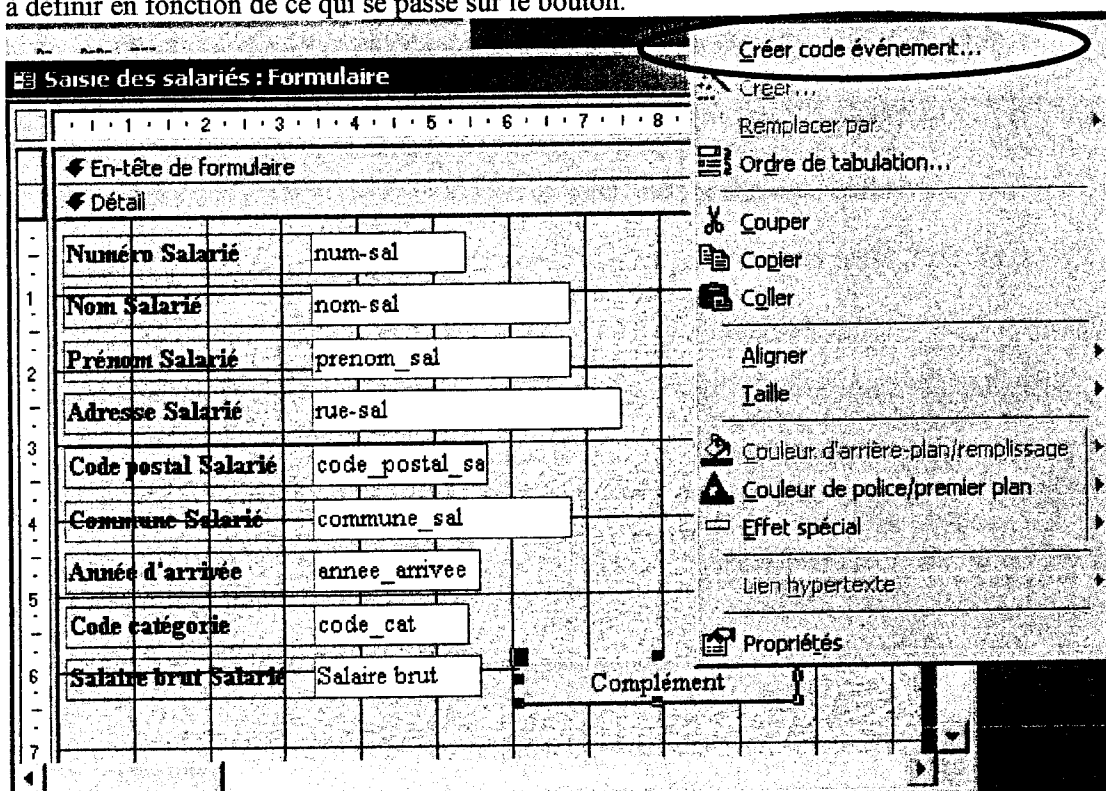


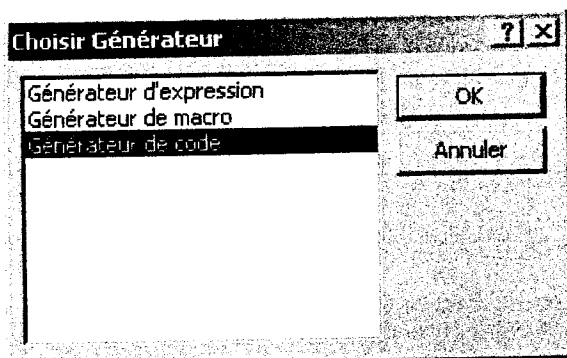


Dans notre cas, cet assistant ne sert à rien. Il est à noter que des contrôles sont préétablis. Ainsi, il est possible d'insérer des boutons pour :

- Rechercher des enregistrements,
- Aller d'enregistrements en enregistrements,
- Exécuter des opérations sur enregistrements : créer, supprimer, dupliquer des enregistrements...
- Faire des opérations sur les formulaires comme l'imprimer...
- Et ainsi de suite.

Le bouton prend un nom (exemple : commande20). Il faut créer un code événement, c'est-à-dire une action à définir en fonction de ce qui se passe sur le bouton.





Dans la mesure où on souhaite inclure une procédure (un module), il faut choisir « Générateur de code ».

La fenêtre de création des modules s'ouvre. Des contrôles d'erreurs sont automatiquement insérés. Leur utilisation étant hors-programme, il suffit de supprimer toutes les lignes (sauf les lignes d'option) et insérer le code suivant :

```
Sub Bouton_Complément_Click()

If Forms![Saisie des salariés]![Salaire brut] <= 10000 Then
MsgBox ("Le complément de salaire est de : " & Forms![Saisie des salariés]![Salaire brut] * 0.15 & "
Francs")
Else
If Forms![Saisie des salariés]![Salaire brut] <= 12000 Then
MsgBox ("Le complément de salaire est de : " & Forms![Saisie des salariés]![Salaire brut] * 0.1 & "
Francs")
Else
MsgBox ("Le complément de salaire est de : " & Forms![Saisie des salariés]![Salaire brut] * 0.05 & "
Francs")
End If
End If

End Sub
```

Dans une expression, on fait référence à la valeur d'un champ en tapant son nom entre crochets droits. Par exemple, **[Salaire brut]** fait référence à la valeur du champ [Salaire brut].

Pour faire référence à la valeur d'un champ dans un formulaire, vous faites précéder le nom du champ par le nom de l'objet système, **FORMS** pour Formulaire.

Exemple : Forms![Saisie des salariés]![Salaire brut]

Cette expression fait référence au champ « Salaire brut » de l'objet « Saisie des salariés » qui appartient à la catégorie des formulaires.

Le « ! » indique la relation entre les éléments. Cette opérateur indique qu'un élément de l'identificateur appartient à un autre. L'opérateur « ! » est toujours suivi du nom d'un élément défini par l'utilisateur, notamment les formulaires, les états, les contrôles et les champs que vous créez. Il indique que l'élément fait partie d'une collection.

Il est enfin possible de modifier les propriétés du bouton :

- La police de caractères,
- Le nom du bouton,
- La légende qui sera inscrite sur le bouton,
- On peut insérer un « texte info-bulle » qui renseignera l'opérateur sur l'utilité du bouton,
- Etc.

Noter que l'appel du code (module) se fait en tant que procédure événementielle (procédure sur événement), l'événement étant « Sur clic ».

Ceci explique également le nom de la procédure :

Sub Bouton_Complément_Click() : Procédure sur clic d'un bouton nommé « Bouton-Complément ».

Bouton de commande: Bouton_Complément

Format	Données	Événement	Autres	Toutes
Nom	Bouton_Complément			
Légende	Complément			
Image	(aucune)			
Type image	Intégré			
Transparent	Non			
Par défaut	Non			
Annuler	Non			
Auto répéter	Non			
Texte barre état				
Adresse lien hypertexte				
Sous-adresse lien hypertexte				
Visible	Oui			
Afficher	Toujours			
Activé	Oui			
Arrêt tabulation	Oui			
Index tabulation	9			
Gauche	6,099cm			
Haut	5,898cm			
Largeur	3,688cm			
Hauteur	0,619cm			
Couleur texte	0			
Police	Times New Roman			
Taille caractères	10			
Epaiss caractères	Standard			
Italique	Non			
Souligné	Non			
Barre de menu contextuel				
Texte d'Info-bulle	Calcul d'un complément de salaire			
Contexte Aide	0			
Remarque				
Sur entrée				
Sur sortie				
Sur réception focus				
Sur perte focus				
Sur clic	[Procédure événementielle]			
Sur double clic				