Institut Universitaire de Technolologie
Département Informatique
1993-1994

RAPPORT TECHNIQUE

LABORATOIRE DE ZOOECOLOGIE DU SOL (INRA)
CEFE/CNRS
1919, Route de Mende
BP: 5051 34033 MONTPELLIER Cedex 1

CONCEPTION ET REALISATION DE LA GESTION INTEGREE DU LABORATOIRE DE ZOOECOLOGIE DU SOL

TUTEURS ENTREPRISE :

M. SOTO Patricio
Mme ALIAGA Régine

TUTEUR ENSEIGNANT:

Mme BELLAHSENE

AUTEUR:

ONDZAGHE Arsène

LES TABLES

LA TABLE ASSOCIE :

Cette table sert à traduire le fait qu'un contrat commande une réalisation qui porte sur un thème donné et que cette même réalisation conduit à une publication des résultats des travaux constituant cette réalisation.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: associe

Nom du champ	Type de données	Description
codeTh	Texte	code du thème lié aux trois champs suivantss
codeCo	Texte	code du contrat lié aux trois autres champs
codePu	Texte	code de la publication liée aux trois autres champs
codeRe	Numérique	code de la réalisation liée aux trois champs précédents

Propriétés du champ

Taille du champ	8
Format	
Légende	
Valeur par défaut	
Valide si	
Message si erreur	
Indexé	Non

Saisissez

Propriétés de la table

Description	
Clé primaire	codeTh; codeCo
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE CHRONO :

Cette table contient les dates des réunions des groupes de travail, les dates des congrès, les délais de publication d'un article ainsi que les dates de prise d'effet et d'expiration des contrats signés par le laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: chrono

Nom du champ	Type de données	Description
codeCh	Texte	code du chrono
codeGT	Texte	code du groupe de travail lié au chrono
titreCh	Texte	nom du du chrono
prédélai	Date/Heure	date à laquelle
délai	Date/Heure	date
remarqueCh	Mémo	informations supplémentaires sur le chrono

Propriétés du champ

Taille du champ	19
Format	
Légende	
Valeur par défaut	
Valide si	
Message si erreur	
Indexé	Oui - Sans doublons

Saisissez

Propriétés de la table

Description	
Clé primaire	codeCh
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE CONTIENT :

Cette table permet de traduire le fait qu'une publication contient une diapositive.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: contient

Nom du champ	Type de données	Description
codeDi	Numérique	numéro du diapositive associé à la publication
codePu	Texte	code de la publication contenant le diapositive

Propriétés du champ

Taille du champ	Entier long
Format	
Décimales	0
Légende	
Valeur par défaut	0
Valide si	
Message d'erreur	
Indexé	Non

Propriétés de la table

Description	
Clé primaire	codeDi; codePu
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE CONTRAT :

Cette table contient des informations sur le contratssinés par le laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: contrat

Nom du champ	Type de données	Description
codeCo	Texte	numéro du contrat
idInstitution	Numérique	code de l'institution signataire du contrat
codeRe	Numérique	code de la réalisation attachée à ce contrat
codeCh	Texte	écheances liées au contrat
titreCo	Texte	titre du contrat
datedebut	Date/Heure	date de prise d' effet du contrat
datefin	Date/Heure	date d' expiration du contrat
remarqueCo	Mémo	

Propriétés du champ

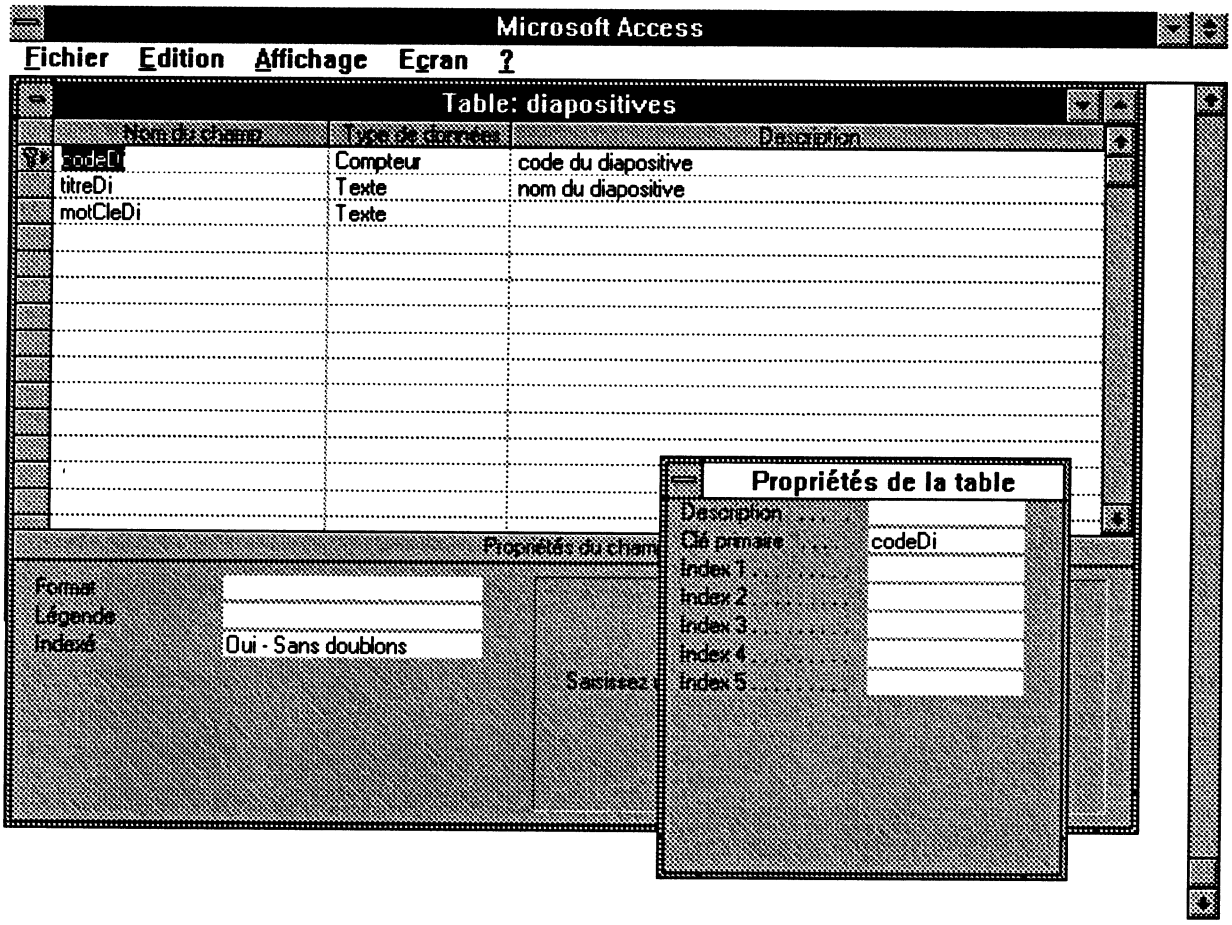
Taille du champ	6
Format	
Légende	
Valeur par défaut	
Valide si	
Message d'erreur	
Indexé	Oui - Sans doublons

Propriétés de la table

Description	
Clé primaire	codeCo
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

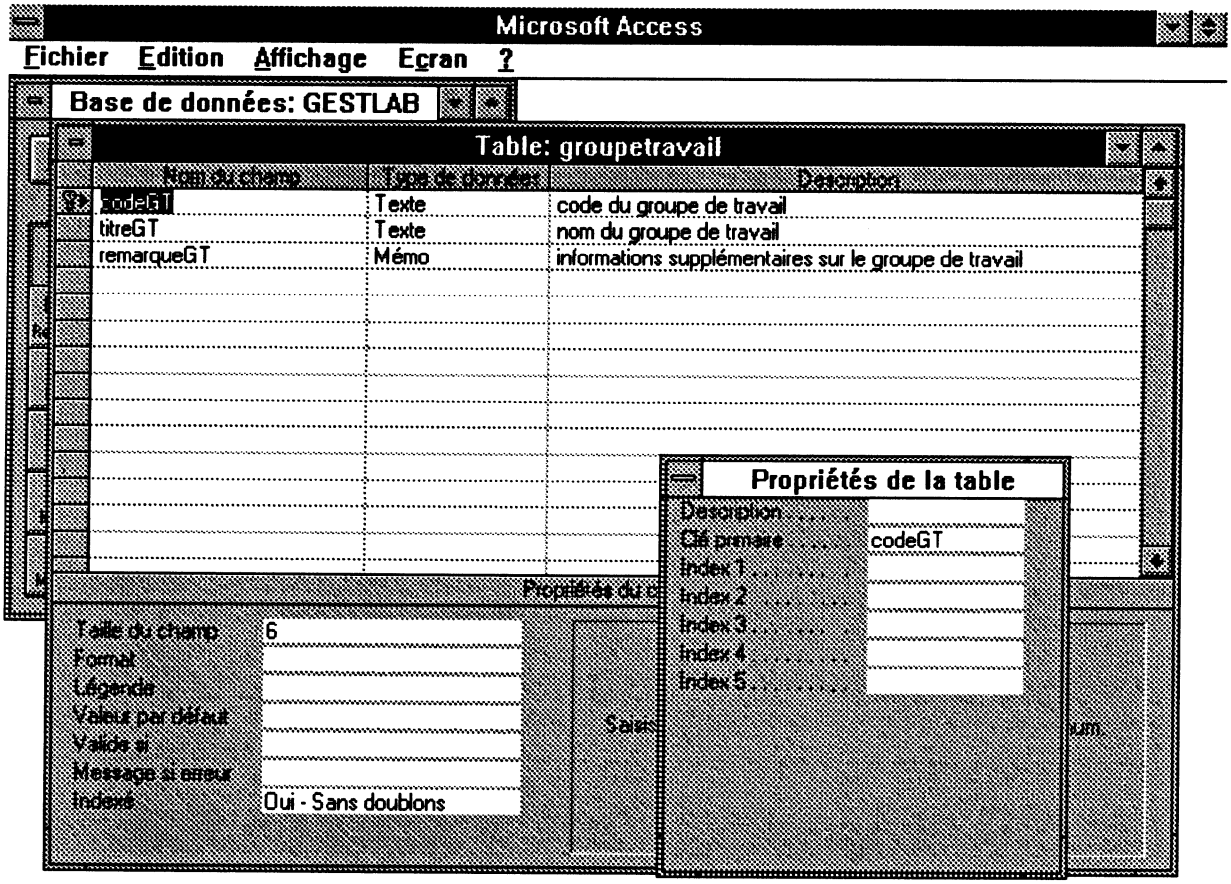
LA TABLE DIAPOSITIVES :

Cette table contient des informations sur les diapositives du laboratoire.



LA TABLE GROUPETRAVAIL :

Cette table contient des informations sur les groupes de travail.



LA TABLE INSTITUTIONS :

Cette table contient toutes les institutions qui entretienent des relations avec le laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: institutions

Nom du champ	Type de données	Description
idInstitution	Numérique	numéro de l' institution
codePays	Texte	numéro du pays auquel appartient l' institution
sigleInstitution	Texte	sigle de l'instiution
nomInstitution	Texte	nom de l'institution
adresse_1	Texte	adresse principale de l'institution
adresse_2	Texte	deuxième adresse de l'institution
codepostal	Texte	code postal de l'institution
ville	Texte	nom de la ville dans laquelle se trouve le siège de l'institution
telephone	Texte	numéro de téléphone de l'institution
telecopie	Texte	numéro de télécopie de l'institution
idObservation	Mémo	informations supplémentaires sur l' institution

Propriétés du champ

Format
Légende
Valeur par défaut
Valide si
Message d'erreur

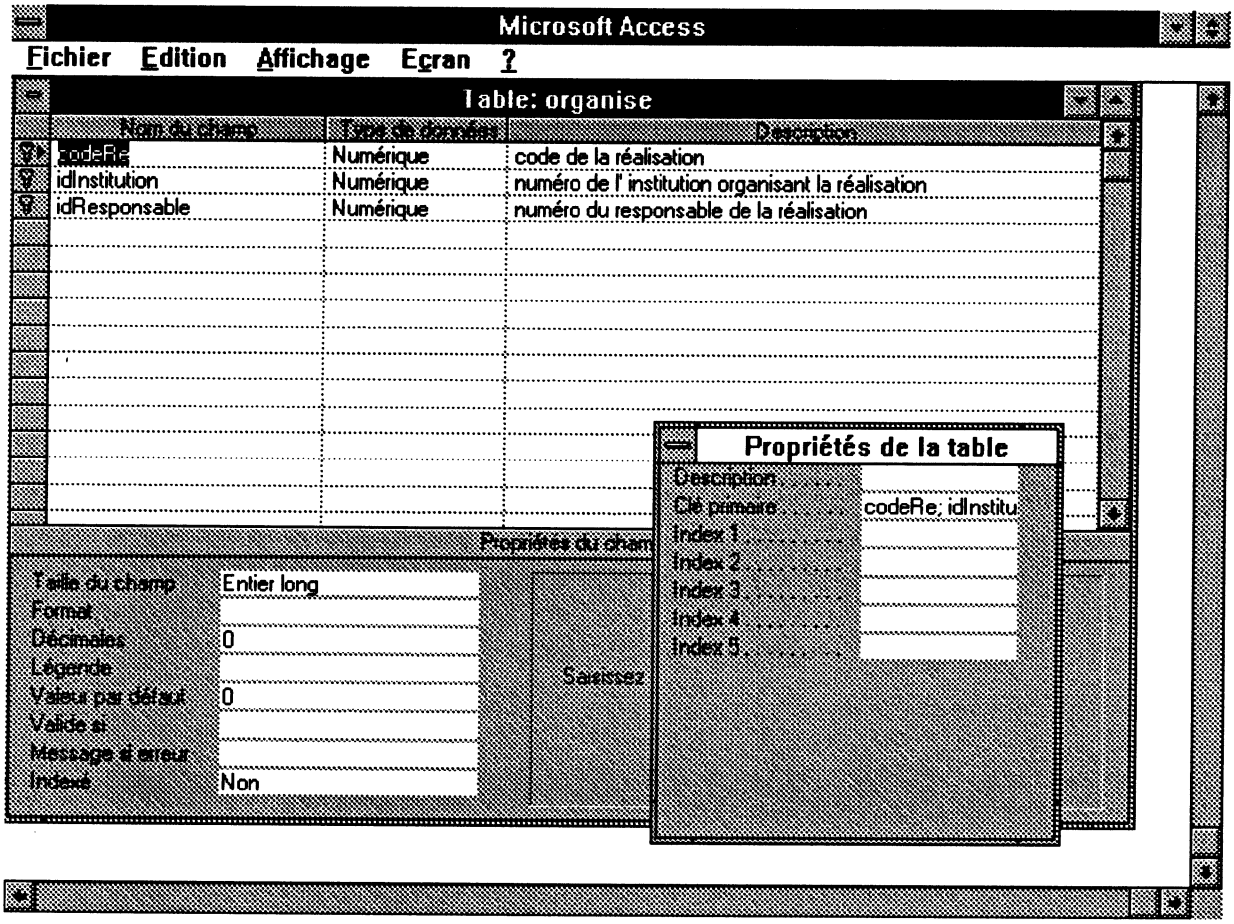
La descript
dans la barr

Propriétés de la table

Description
Clé primaire ... idInstitution
Index 1
Index 2
Index 3
Index 4
Index 5

LA TABLE ORGANISE :

Cette permet de traduire le fait qu'un responsable dépendant d'une institution dirige des travaux pour le laboratoire.



LA TABLE PAYS :

Cette table contient tous les pays qui abritent les institutions qui sont en relation avec le laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: pays

Nom du champ	Type de données	Description
codePays	Texte	numéro du pays
codenum	Numérique	
codecontinent	Numérique	numéro du continent dans lequel est situé l'institution
nompaysFra	Texte	nom du pays en français
NompaysAng	Texte	nom du pays en anglais
etatetutdes	Texte	

Propriétés du champ

Taille du champ	3
Format	
Légende	
Valeur par défaut	
Valide si	
Message si erreur	
Indexé	Oui - Sans doublons

Propriétés de la table

Description	
Clé primaire	codePays
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE PUBLICATION :

Cette table contient des infomations sur les publicatons écrites sur les travaux effectués au laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: publication

Nom du champ	Type de données	Description
codePu	Texte	code la publication
codeCh	Texte	code du délai de rédaction de la publication
auteur	Texte	
titrePu	Texte	titre de la publication
comitelecture	Oui/Non	spécifie si la publication a été soumise à un comité de lecture
revue	Oui/Non	spécifie si la publication est un article de revue
nom revue	Texte	
numero revue	Numérique	
volume revue	Numérique	
livre	Oui/Non	spécifie si la publication est un livre
chapitrelivre	Oui/Non	spécifie si la publication est un chapitre de livre
date parution	Numérique	
titre livre	Texte	
auteur livre	Texte	
editeur livre	Texte	

Propriétés du champ

Taille du champ	6
Format	
Légende	
Valeur par défaut	
Valide si	
Message si erreur	
Indexé	Oui - Sans doublons

Saisissez

Propriétés de la table

Description	
Clé primaire	codePu
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE REALISATIONS :

Cette table contient des informations sur les réalisations effectuées au laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: realisations

Nom du champ	Type de données	Description
codeRe	Numérique	code de la réalisation
Nomrealisation	Texte	Nom de la réalisation
remarqueRe	Mémo	informations supplémentaires sur la réalisation

Propriétés du champ

Taille du champ	Entier long
Format	
Décimales	Auto
Légende	
Valeur par défaut	0
Valide si	
Message si erreur	
Indexé	Oui - Sans doublons

Saisissez

Propriétés de la table

Description	
Clé primaire	codeRe
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE RESPONSABLES :

Cette table contient des informations sur toutes les personnes qui éffectuent des travaux pour le laboratoire ou qui entretiennent des relations scientifiques avec le laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: responsables

Nom du champ	Type de données	Description
idResponsable	Numérique	numéro du responsable
idInstitution	Numérique	nom de l'organisme duquel il dépend
nom_prenom	Texte	nom st prénom du responsable
fonction	Texte	responsabiblité occupée par le responsable
idRemarque	Mémo	informations supplémentaires su le responsable
telephone	Texte	numéro personnel
domicile	Texte	
service	Texte	

Propriétés du champ

Taille du champ	Entier long
Format	
Décimales	Auto
Légende	
Valeur par défaut	0
Valide si	
Message d'erreur	
Indexé	Oui - Sans doublons

Propriétés de la table

Description	
Clé primaire	idResponsable
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

LA TABLE THEME :

Cette table contient tous les thèmes des travaux effectués au laboratoire.

Microsoft Access

Fichier Edition Affichage Ecran ?

Table: theme

Nom du champ	Type de données	Description
codeTh	Texte	code du thème
titreTh	Texte	titre du thème
remarqueTh	Mémo	informations supplémentaires sur le thème
motCleTh	Texte	

Propriétés du cha

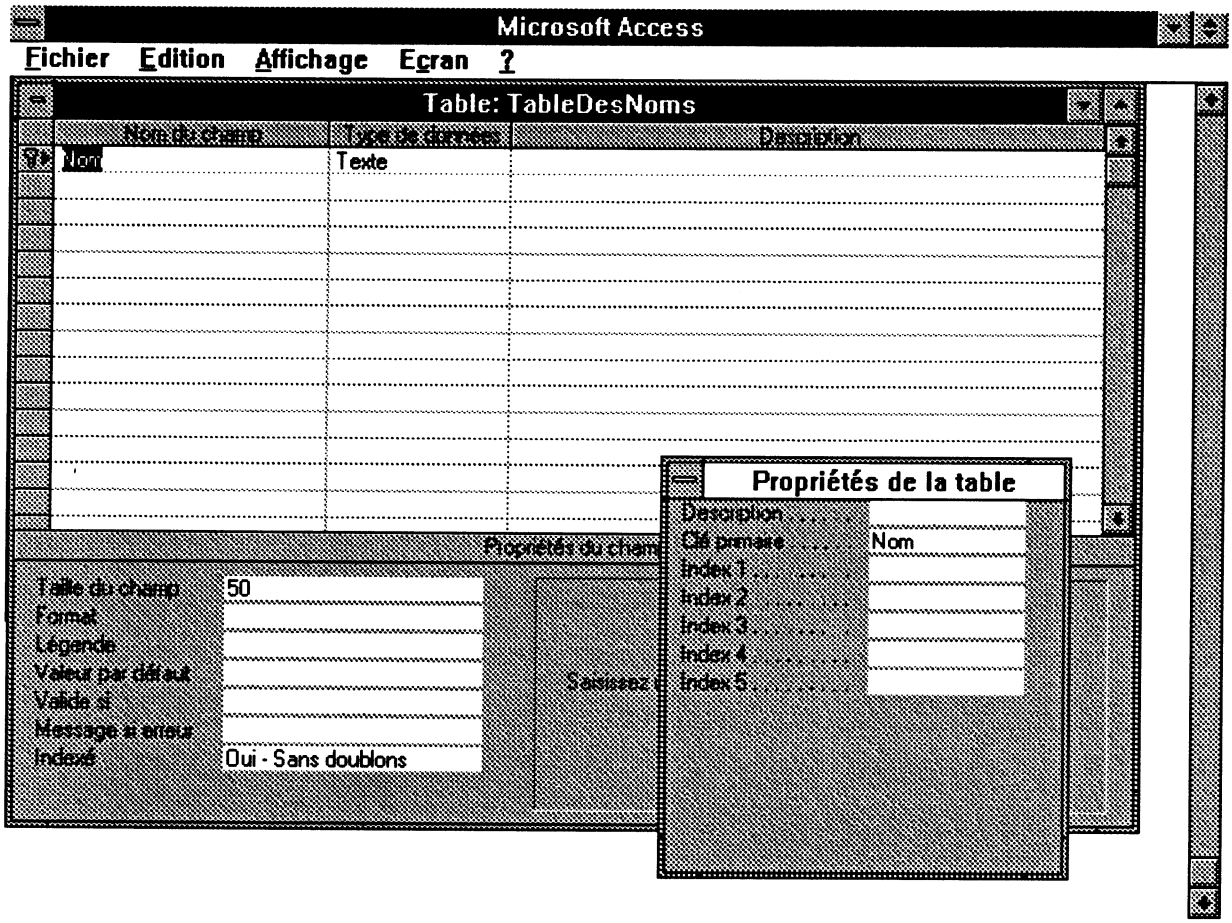
Taille du champ	8
Format	
Légende	
Valeur par défaut	
Valide si	
Message si erreur	
Indexé	Oui - Sans doublons

Propriétés de la table

Description	
Clé primaire	codeTh
Index 1	
Index 2	
Index 3	
Index 4	
Index 5	

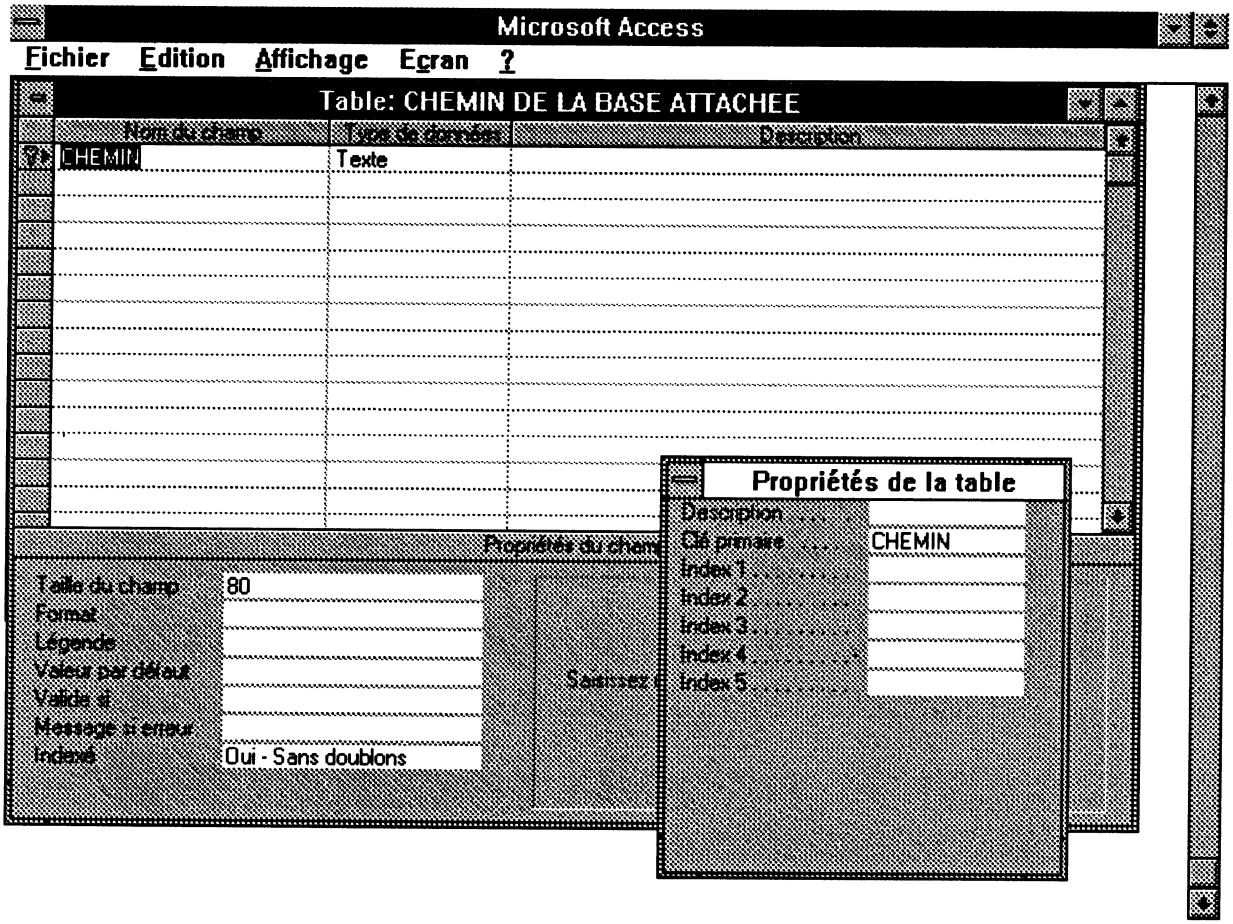
LA TABLE TABLEDESNOMS :

Cette table contient les noms de toutes les tables-dossiers et de toutes les tables-liens de la base données.



LA TABLE CHEMIN DE LA BASE ATTACHEE :

Cette table contient le chemin d'accès à la base de données de gestion intégrée du laboratoire.



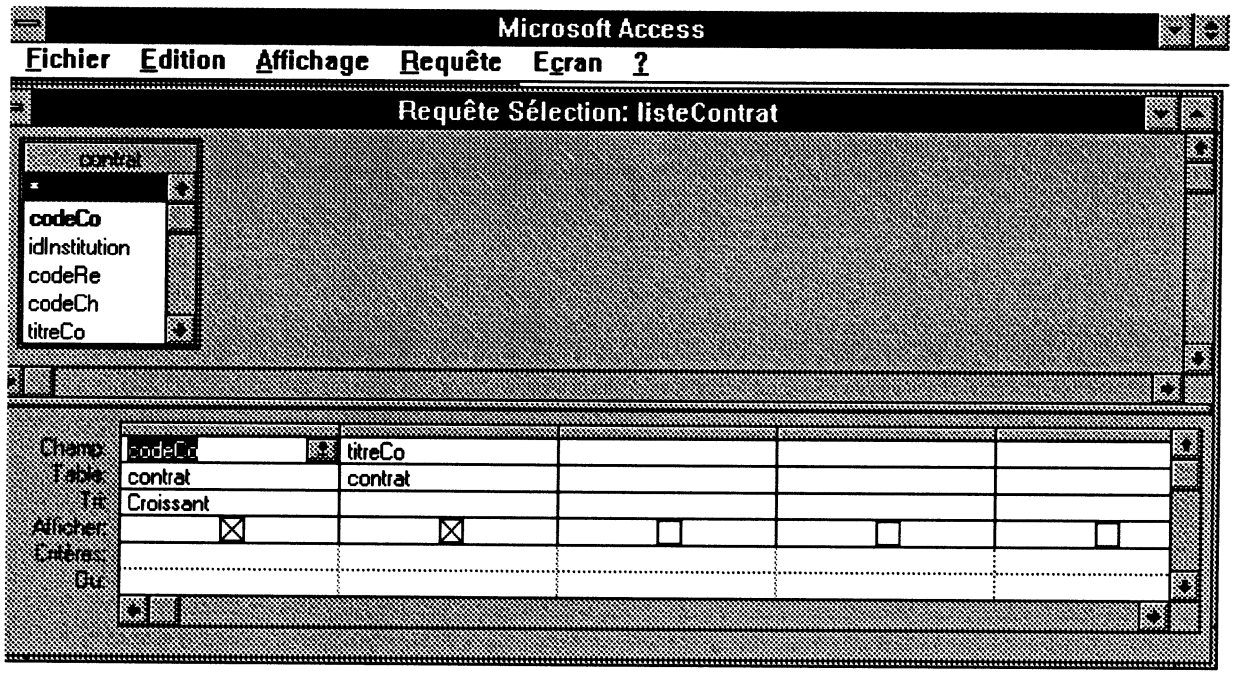
LES REQUETES

Pour réaliser le logiciel de gestion intégrée du laboratoire de Zooécologie, nous avons utilisé deux types de requêtes :
-les requêtes pour afficher le contenu d'une table. Ces requêtes sont utilisées dans les formulaires de saisie des clés afin de faciliter la création de liens entre les enregistrements des tables de la base de données qui sont mises en relation.
-les requêtes pour afficher l'ensemble des enregistrements d'une table liés à un enregistrement donné dans une table donnée. Ces requêtes sont utilisées dans les formulaires principaux pour afficher les liens entre les enregistrements.

1 Les requêtes pour afficher le contenu d'une table dans un formulaire de saisie des clés :

LISTECONTRAT :

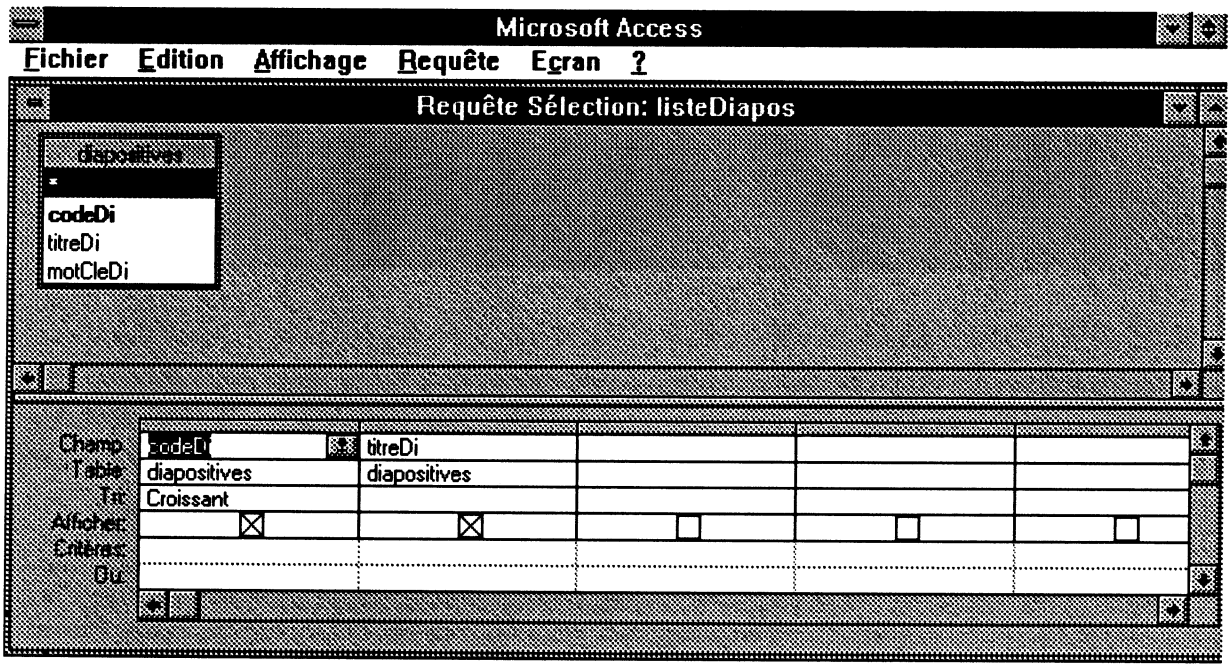
Cette requête affiche tous les enregistrements de la table Contrat.



Cette requête est utillisée pour afficher les contrats dans la liste déroulante correspondante du formulaire de saisie Nouvel Associe.

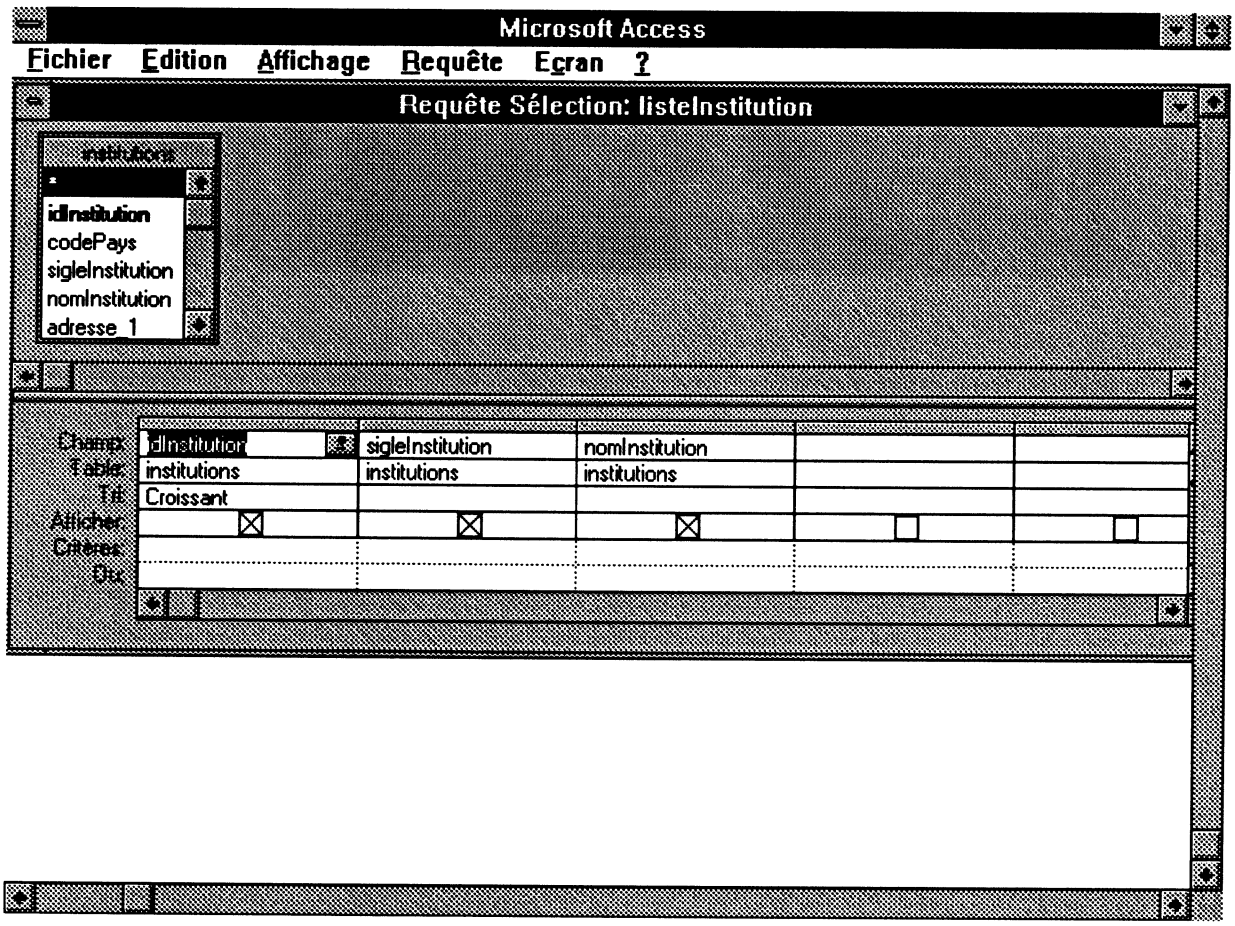
LISTEDIAPO :

Cette requête permet de lister tous les enregistrements de la table Diapositives. Elle est utilisée pour afficher les diapositives de la base de données dans la liste déroulante diapositive du formulaire de saisie des clés Nouveau contient.



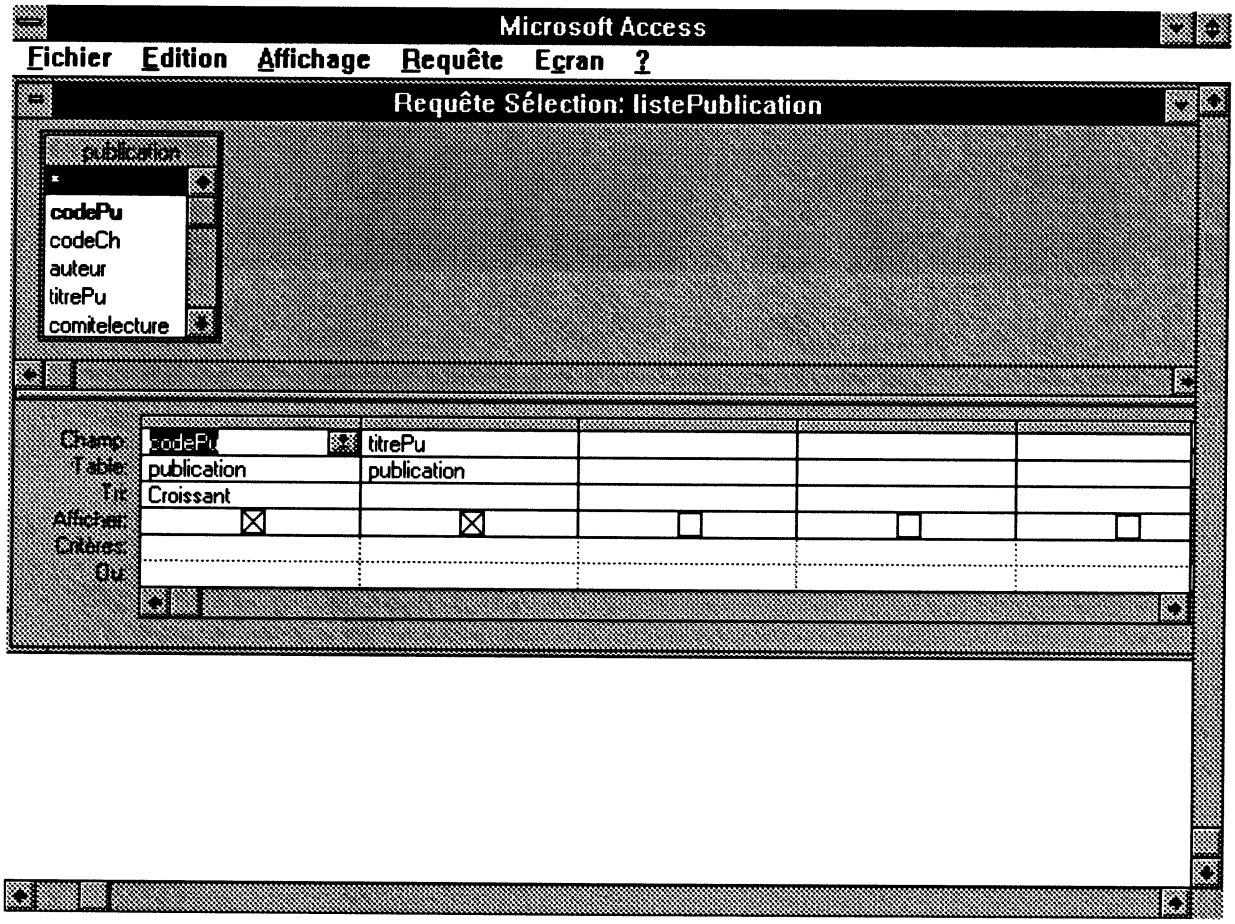
LISTEINSTITUTION :

Cette requête permet de lister tous les enregistrements de la table Institutions. Elle est utilisée pour afficher les institutions de la base de données dans la liste déroulante institutions du formulaire de saisie des clés Nouvel organise.



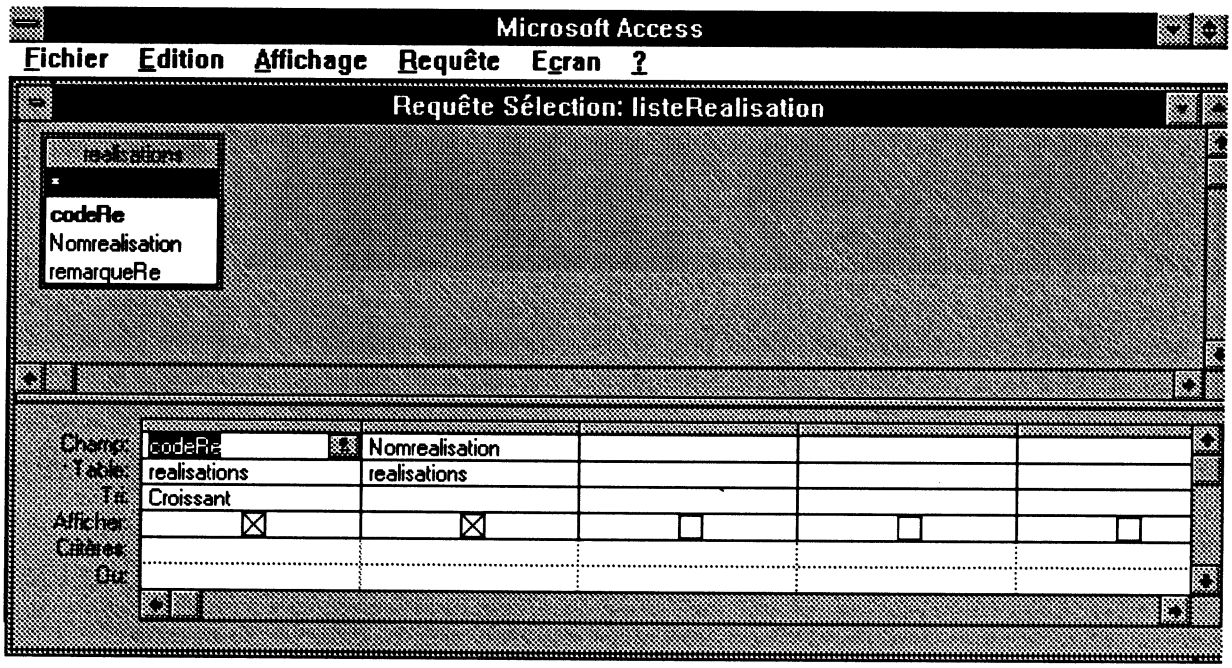
LISTEPUBLICATION :

Cette requête permet de lister tous les enregistrements de la table Publication. Elle est utilisée pour afficher les publications de la base de données dans la liste déroulante publications dans les formulaires de saisie des clés Nouveau contient et Nouvel associe.



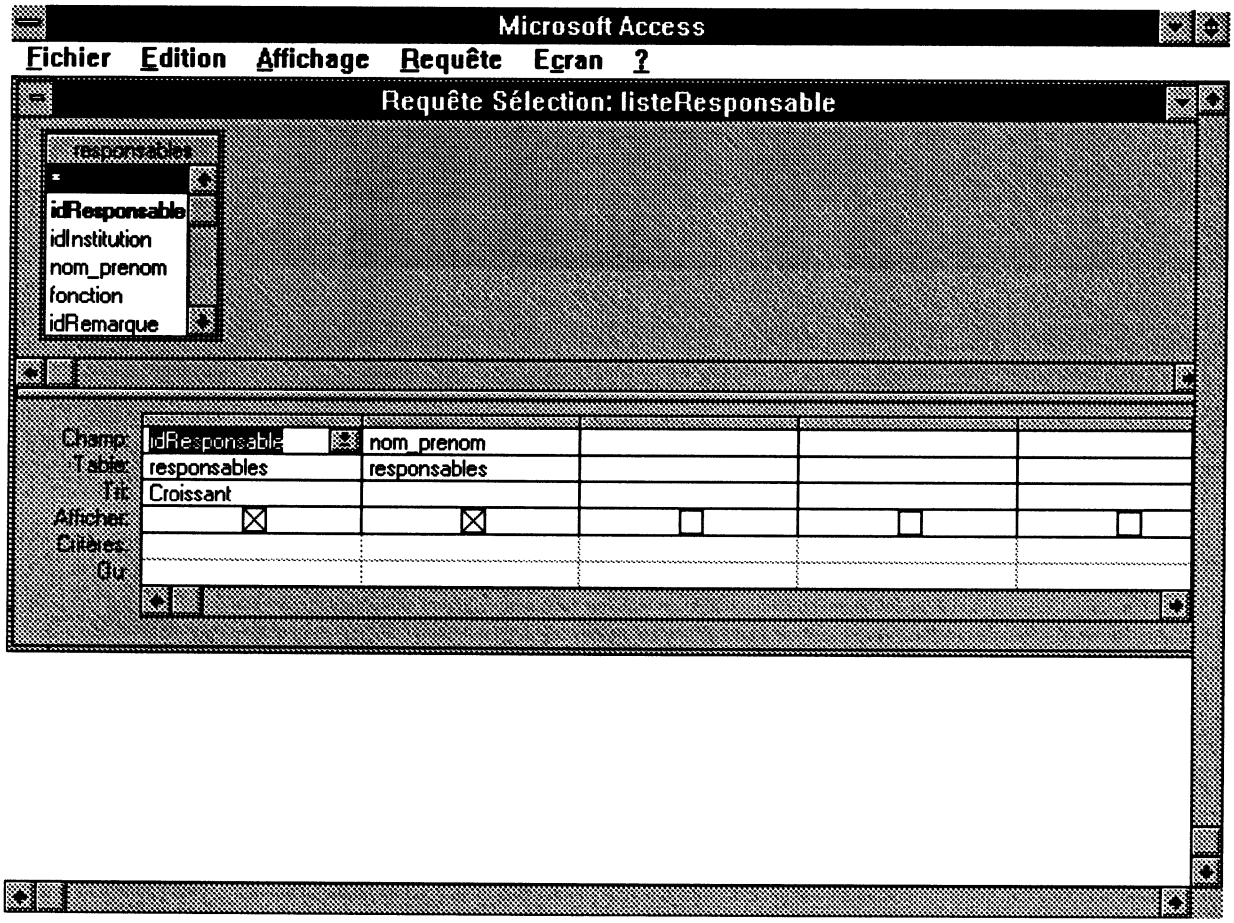
LISTEREALISATION :

Cette requête permet de lister tous les enregistrements de la table Realisations. Elle est utilisée pour afficher les réalisations de la base de données dans la liste déroulante réalisations dans les formulaires de saisie des clés Nouvel associe et Nouvel organise.



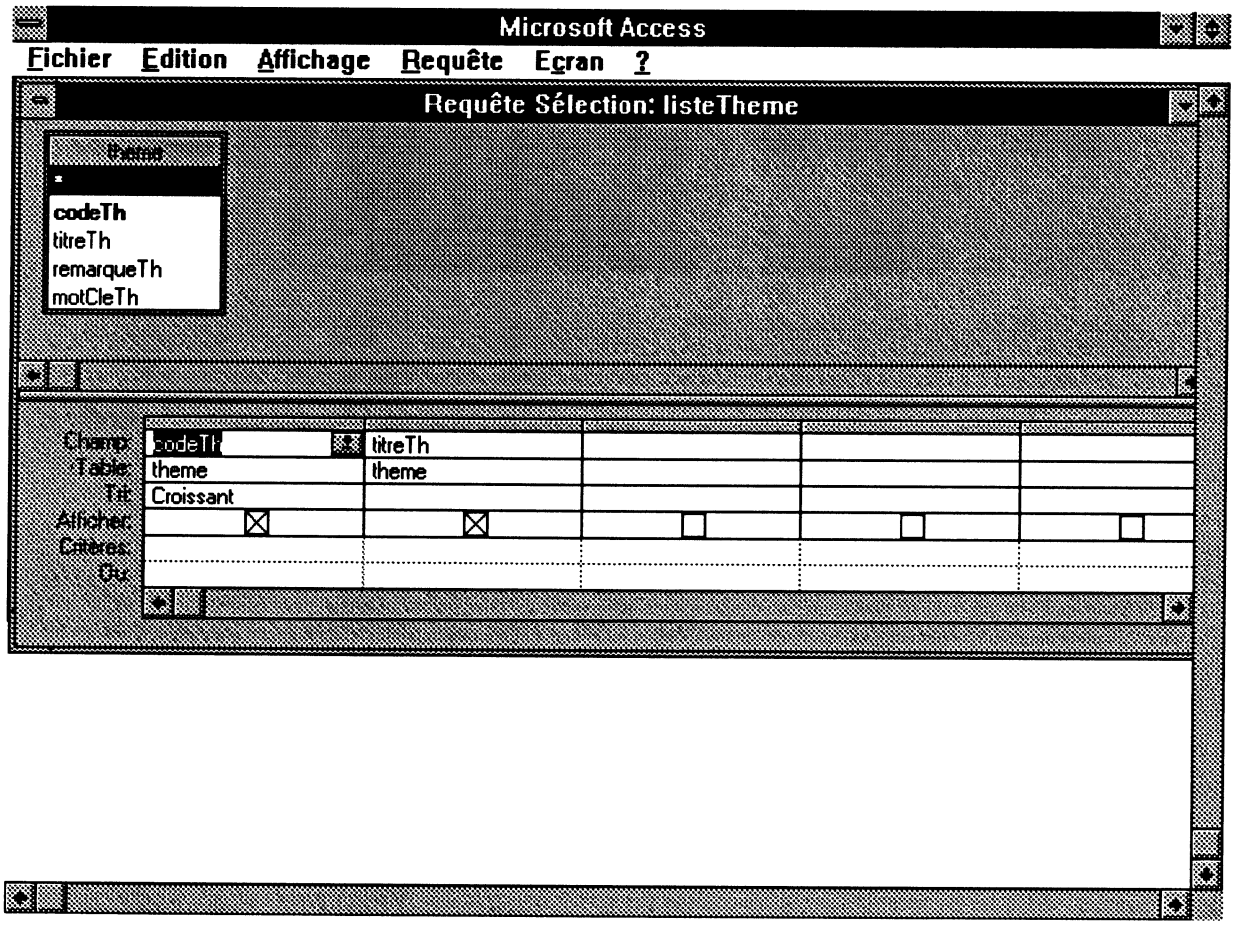
LISTERESPONSABLE :

Cette requête permet de lister tous les enregistrements de la table Responsables. Elle est utilisée pour afficher les responsables de la base de données dans la liste déroulante responsables dans le formulaire de saisie des clés Nouvel organise.



LISTETHEME :

Cette requête permet de lister tous les enregistrements de la table Theme.
Elle est utilisée pour afficher les thèmes de la base de données dans la liste déroulante thèmes dans le formulaire de saisie des clés Nouvel associe.



2 Les requêtes d'affichage des liens :

*Requête pour afficher le contrat lié au chrono actif dans le formulaire chrono :

SELECT [codeCo] FROM contrat WHERE [codeCh] = Formulaires![chrono]![codeCh];

Cette requête-ci est liée à la *propriété Contenu* de la liste déroulante contrats du formulaire principal Chrono.

*Requête pour afficher la publication liée au chrono actif dans le formulaire Chrono :

SELECT [codePu] FROM publication WHERE [codeCh] = Formulaires![chrono]![codeCh];

Cette requête-ci est attachée à la la *propriété Contenu* de la liste déroulante publications du formulaire Chrono.

*Requête pour afficher les publications liées au contrat actif dans le formulaire Contrat :

SELECT [codePu] FROM associe WHERE [codeCo] = Formulaires![contrat]![codeCo];

Cette requête-ci est attachée à la *propriété Contenu* de la liste déroulante publications du formulaire Contrat.

*Requête pour afficher les thèmes de recherche liés au contrat actif dans le formulaire Contrat :

SELECT DISTINCT [codeTh] FROM associe
WHERE [codeCo]=Formulaires![contrat]![codeCo];

Cette requête-ci est attachée à la *propriété Contenu* de la liste déroulante thème du formulaire Contrat.

*Requête pour afficher les publications contenant la diapositive active dans le formulaire Diapositives :

SELECT [codePu] FROM contient WHERE [codeDi]=Formulaires![diapositives]![codeDi];

Cette requête-ci est liée à la liste déroulante publications du formulaire Diapositives.

*Requête pour afficher les chronos liés au groupe de travail actif dans le formulaire Groupetravail :

SELECT [codeCh] FROM chrono
WHERE [codeGT] = Formulaires![groupetravail]![codeGT];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante chrono du formulaire Groupetravail.

*Requête pour afficher les responsables dépendants de l'institution active dans le formulaire Institutions :

SELECT [idResponsable] FROM responsables
WHERE [idInstitution]=Formulaires![institutions]![idInstitution];

La requête précédente est attachée à la *propriété Contenu* de la liste déroulante responsable du formulaire Institutions.

*Requête pour afficher les contrats signés par l'institution active avec le laboratoire dans le formulaire Institutions :

SELECT [codeCo] FROM contrat
WHERE [idInstitution]=Formulaires![institutions]![idInstitution];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante contrat du formulaire Institutions.

*Requête pour afficher les réalisations liées à l'institution active dans le formulaire Institutions :

SELECT [codeRE] FROM organise
WHERE [idInstitution]=Formulaires![institutions]![idInstitution];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante réalisations du formulaire Institutions.

*Requête pour afficher les institutions apartenant au pays actif dans le formulaire Pays :

SELECT [idInstitution] FROM institutions
WHERE [codePays]=Formulaires![pays]![codePays];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante institutions du formulaire Pays.

*Requête pour afficher le contrat lié à la publication active dans le formulaire Publications :

SELECT [codeCo] FROM associe WHERE [codePu]=Formulaires![publications]![codePu];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante contrat du formulaire Publications.

*Requête pour afficher les diapositives contenues dans la publication active dans le formulaire Publications :

SELECT [codeDI] FROM contient WHERE [codePu]=Formulaires![publications]![codePu];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante diapositive du formulaire Publications.

*Requête pour afficher les réalisations relatées dans la publication active dans le formulaire Publications :

SELECT [codeRe] FROM associe WHERE [codePu]=Formulaires![publications]![codePu];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante réalisation du formulaire Publications.

*Requête pour afficher les thèmes des réalisations relatées dans la publication active dans le formulaire Publications :

SELECT [codeTh] FROM associe WHERE [codePu]=Formulaires![publications]![codePu];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante thème du formulaire Publications.

*Requête pour afficher les institutions liées à la réalisation active dans le formulaire Realisations

SELECT [idInstitution] FROM organise
WHERE [codeRE] = Formulaires![realisations]![codeRe];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante institution du formulaire Realisations.

*Requête pour afficher les contrats liés à la réalisation active dans le formulaire Realisations

SELECT [codeCo] FROM contrat
WHERE [codeRe] = Formulaires![realisations]![codeRe];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante contrat du formulaire Realisations.

*Requête pour afficher les responsables de la réalisation active dans le formulaire Realisations :

SELECT [idResponsable] FROM organise
WHERE organise.codeRe=Formulaires![realisations]![codeRe];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante responsable du formulaire Realisations.

*Requête pour afficher les publications issues de la réalisation active dans le formulaire Realisations :

SELECT [codePu] FROM associe WHERE [codeRe] = Formulaires![realisations]![codeRe];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante publication du formulaire Realisations.

*Requête pour afficher les thèmes de la réalisation active dans le formulaire Realisations

SELECT DISTINCT [codeTh] FROM associe
WHERE [codeRe] = Formulaires![realisations]![codeRe];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante thème du formulaire Realisations.

*Requête pour afficher les réalisations effectuées par le responsable actif dans le formulaire :

SELECT [codeRe] FROM organise
WHERE [idResponsable]=Formulaires![responsables]![idResponsable];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante réalisation du formulaire Responsables.

*Requête pour afficher les contrats liés au thème actif du formulaire Themes :

SELECT DISTINCT [codeCo] FROM associe
WHERE [codeTh] = Formulaires![themes]![codeTh];

Cette requête est attachée à la *propriété Contenu* de la liste déroulante contrat du formulaire Themes.

*Requête pour afficher les publications portant sur le thème actif du formulaire Themes :

```
SELECT DISTINCT [codePu] FROM associe
WHERE [codeTh] = Formulaires![themes]![codeTh];
```

Cette requête est attachée à la *propriété Contenu* de la liste déroulante publication du formulaire Themes.

*Requête pour afficher les réalisations portant sur le thème actif dans le formulaire Themes :

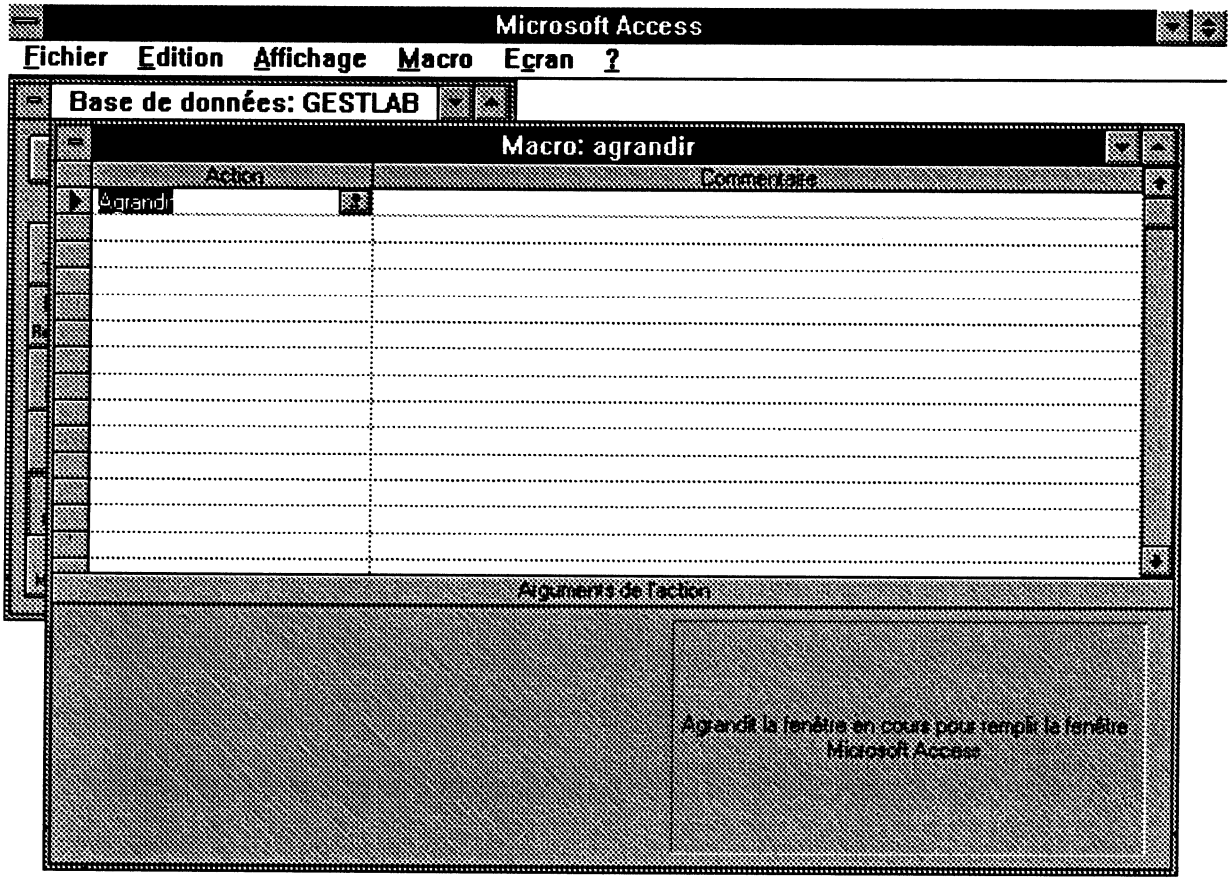
```
SELECT DISTINCT [codeRe] FROM associe
WHERE [codeTh] = Formulaires![themes]![codeTh];
```

Cette requête est attachée à la *propriété Contenu* de la liste déroulante réalisation du formulaire Themes.

LES MACROS

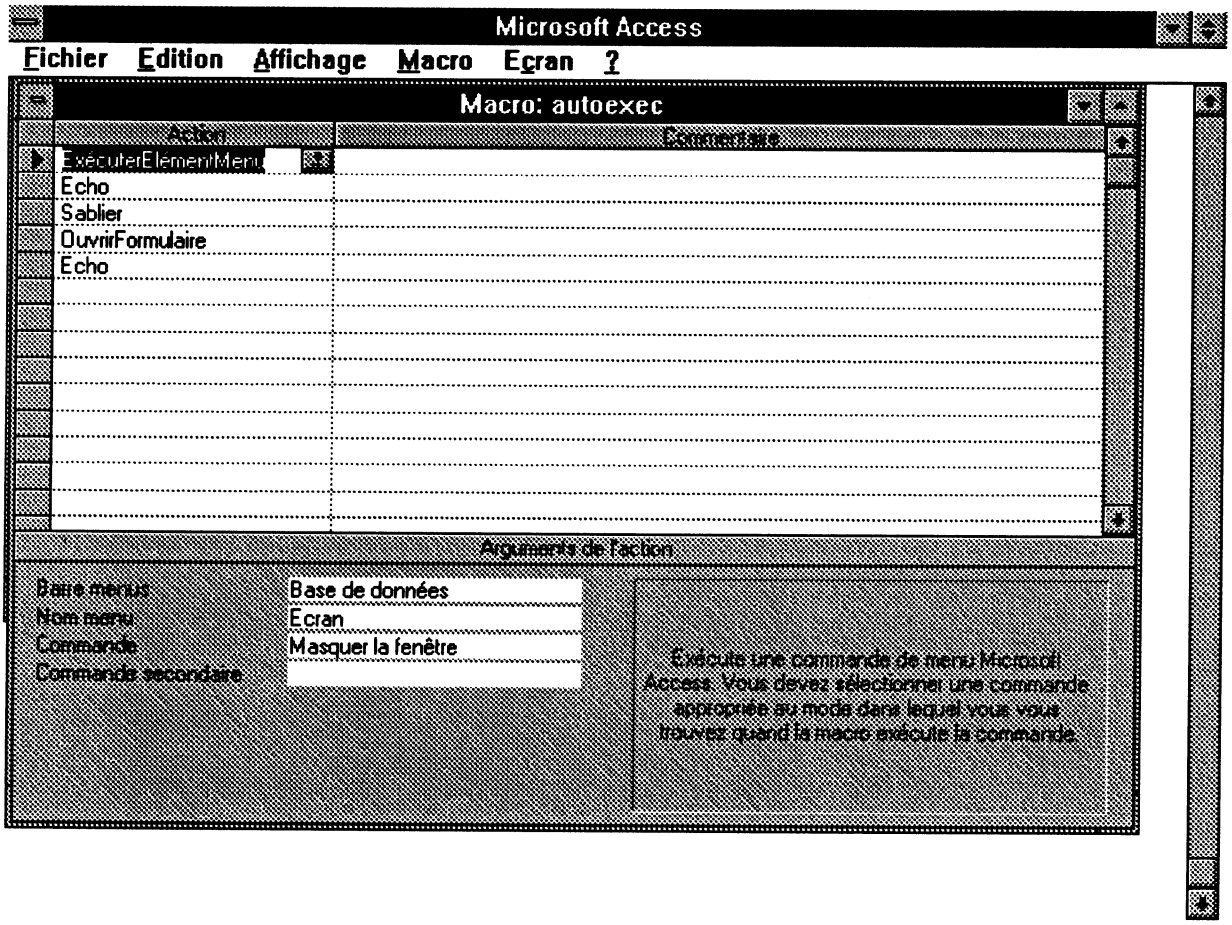
MACRO AGRANDIR :

Cette macro permet d'agrandir un formulaire.



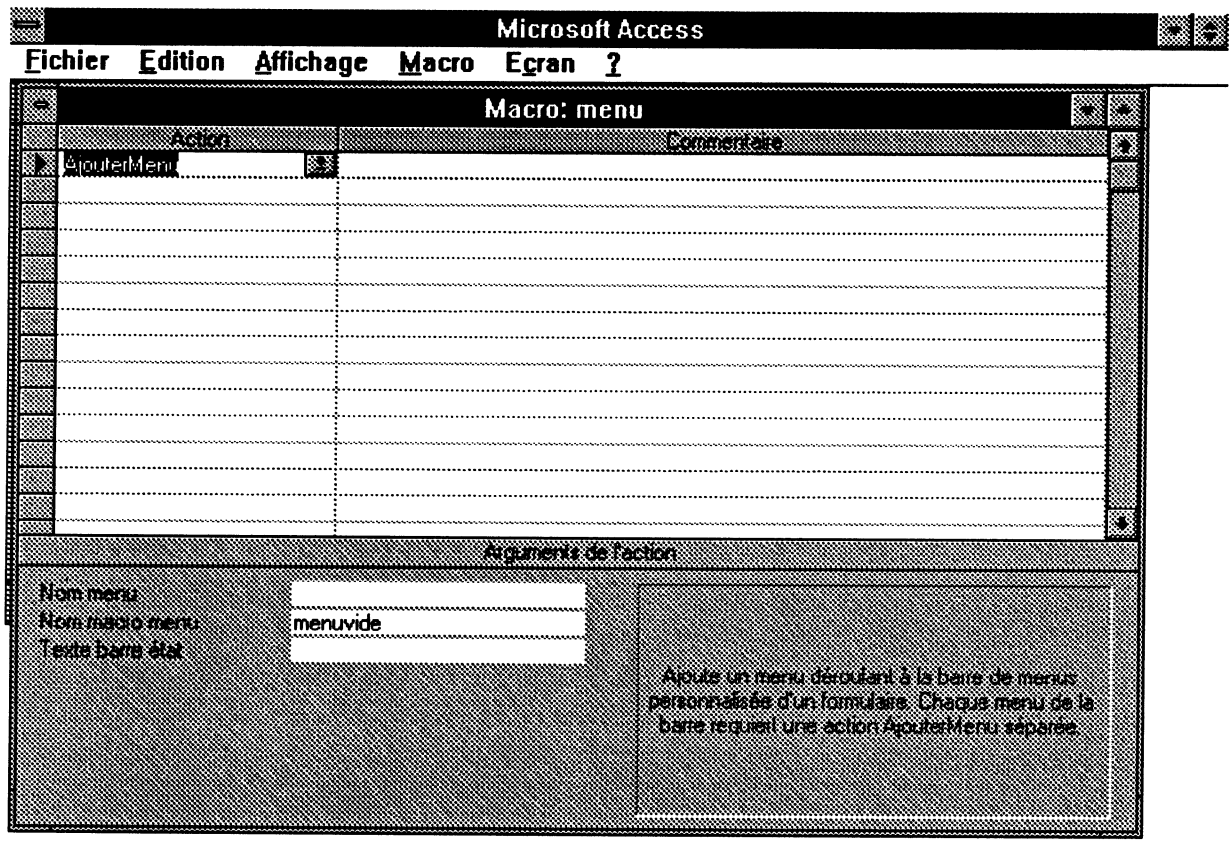
MACRO AUTOEXEC :

Cette macro permet d'ouvrir le Menu Général lors du lancement de l'application de gestion intégrée du laboratoire.



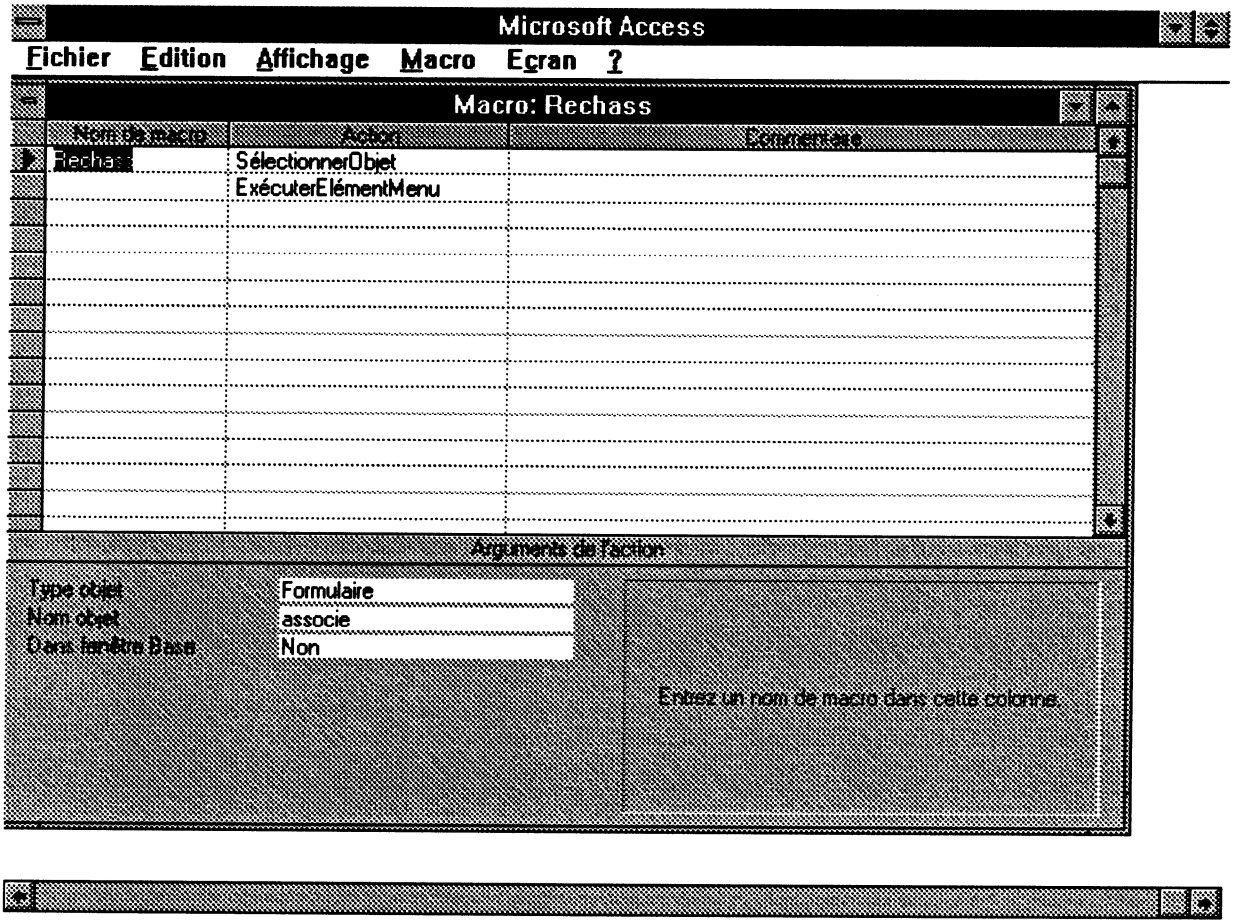
MACRO MENU :

Cette macro permet de cacher la barre de menu système de Access.



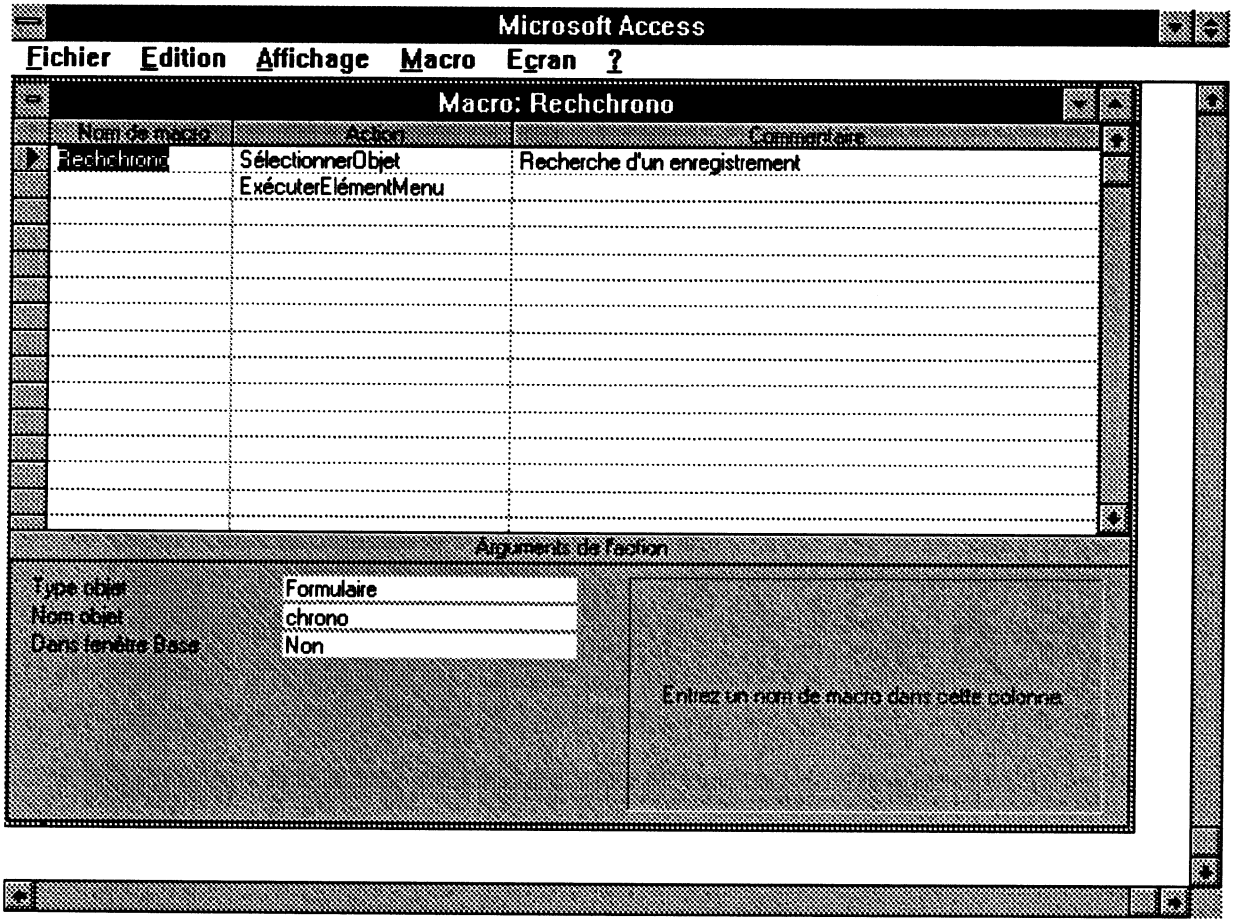
MACRO RECHASS :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Associe.



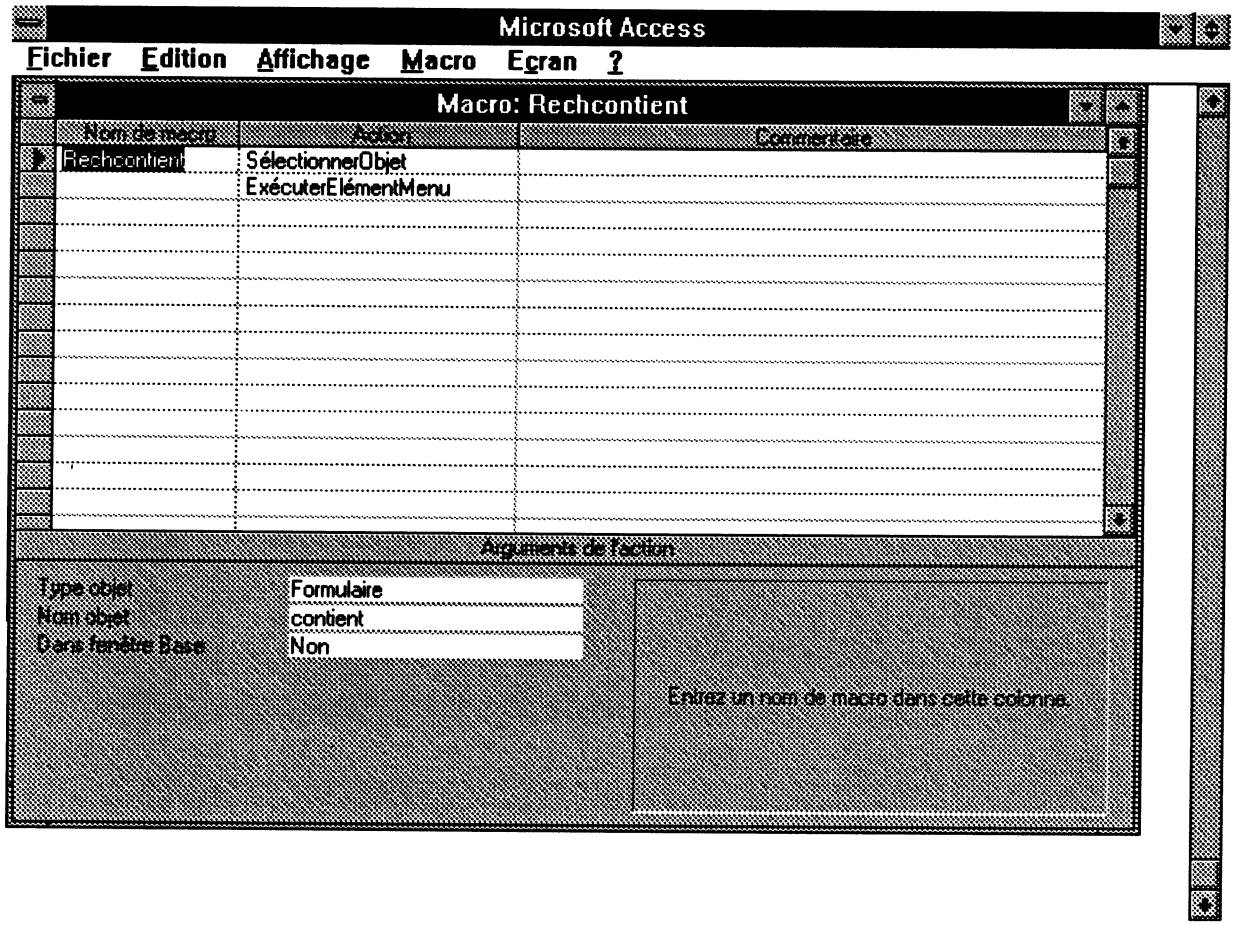
MACRO RECHCHRONO :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Chrono.



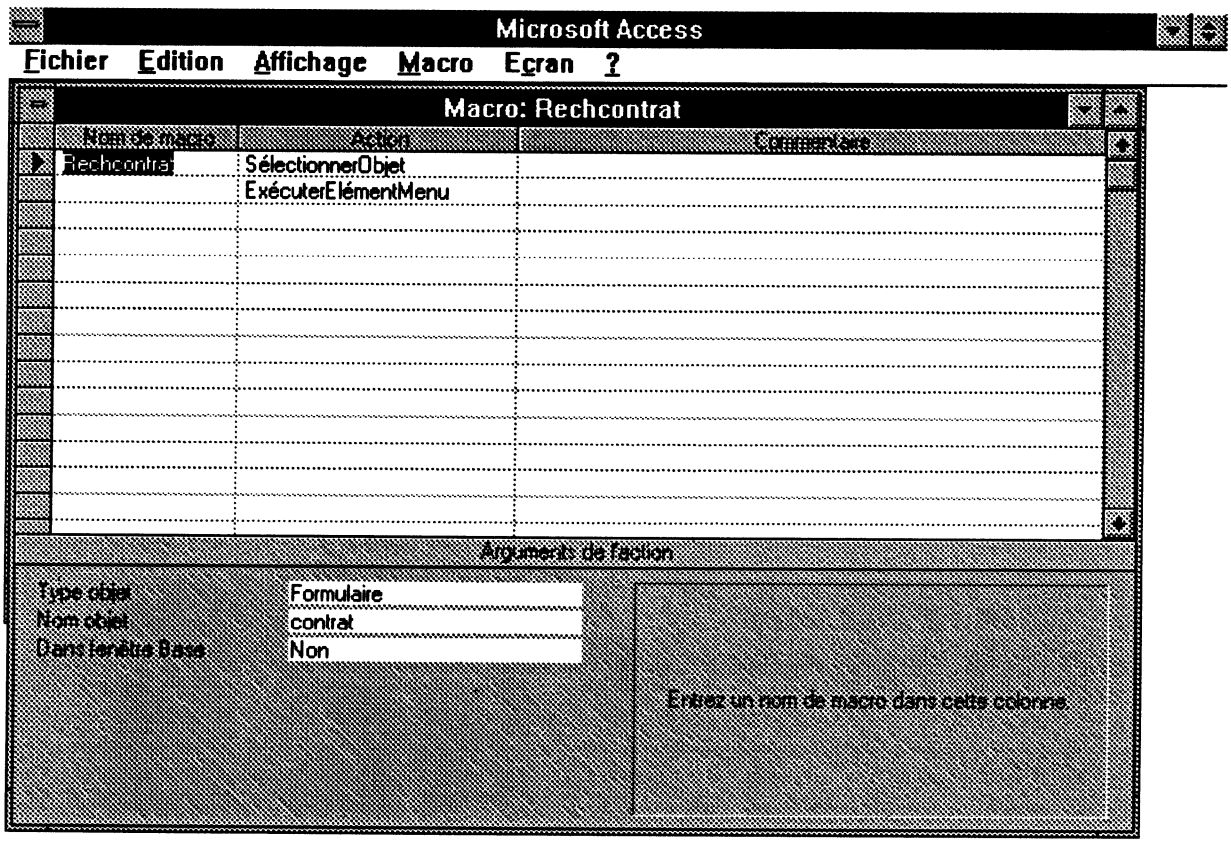
MACRO RECHCONTIENT :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Contient.



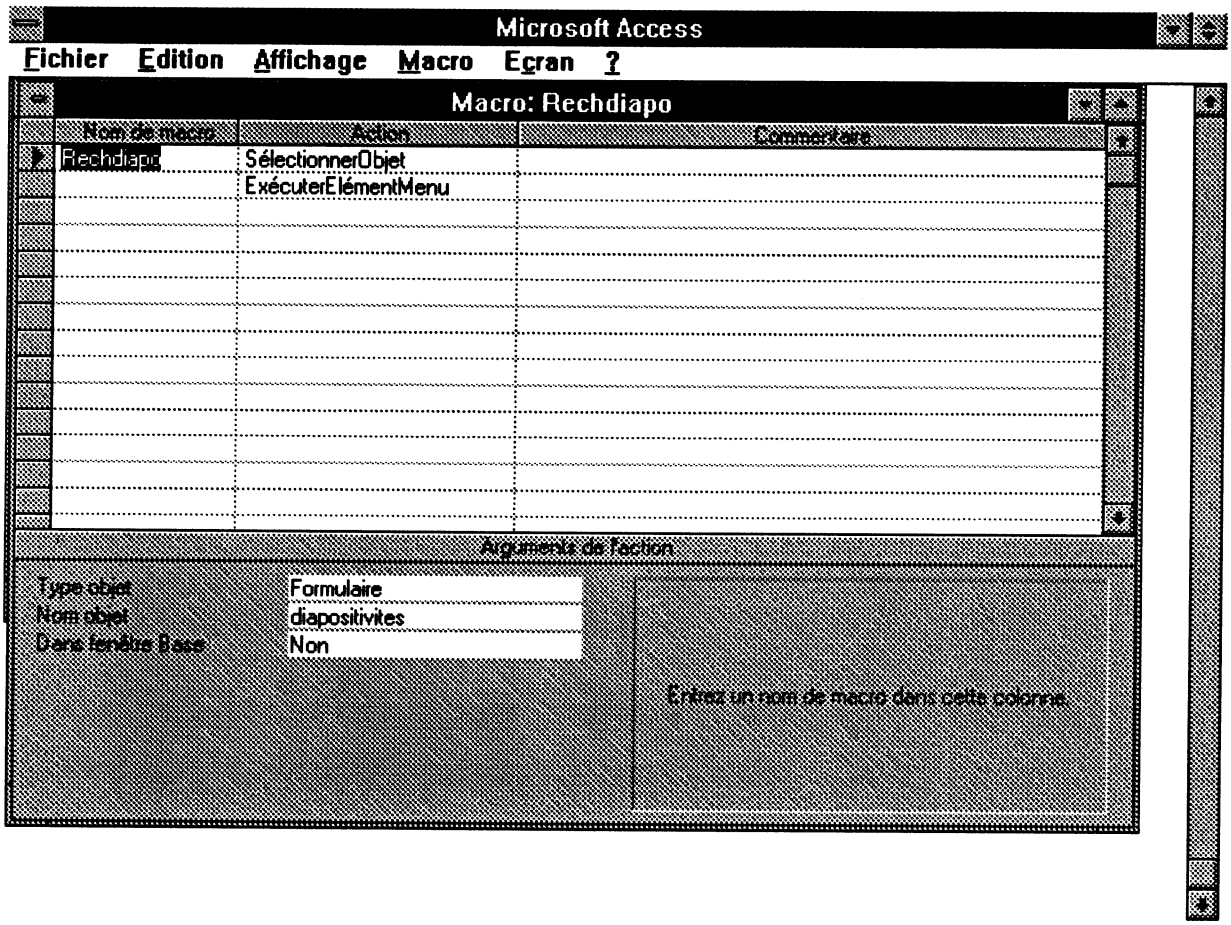
MACRO RECHCONTRAT :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Contrat.



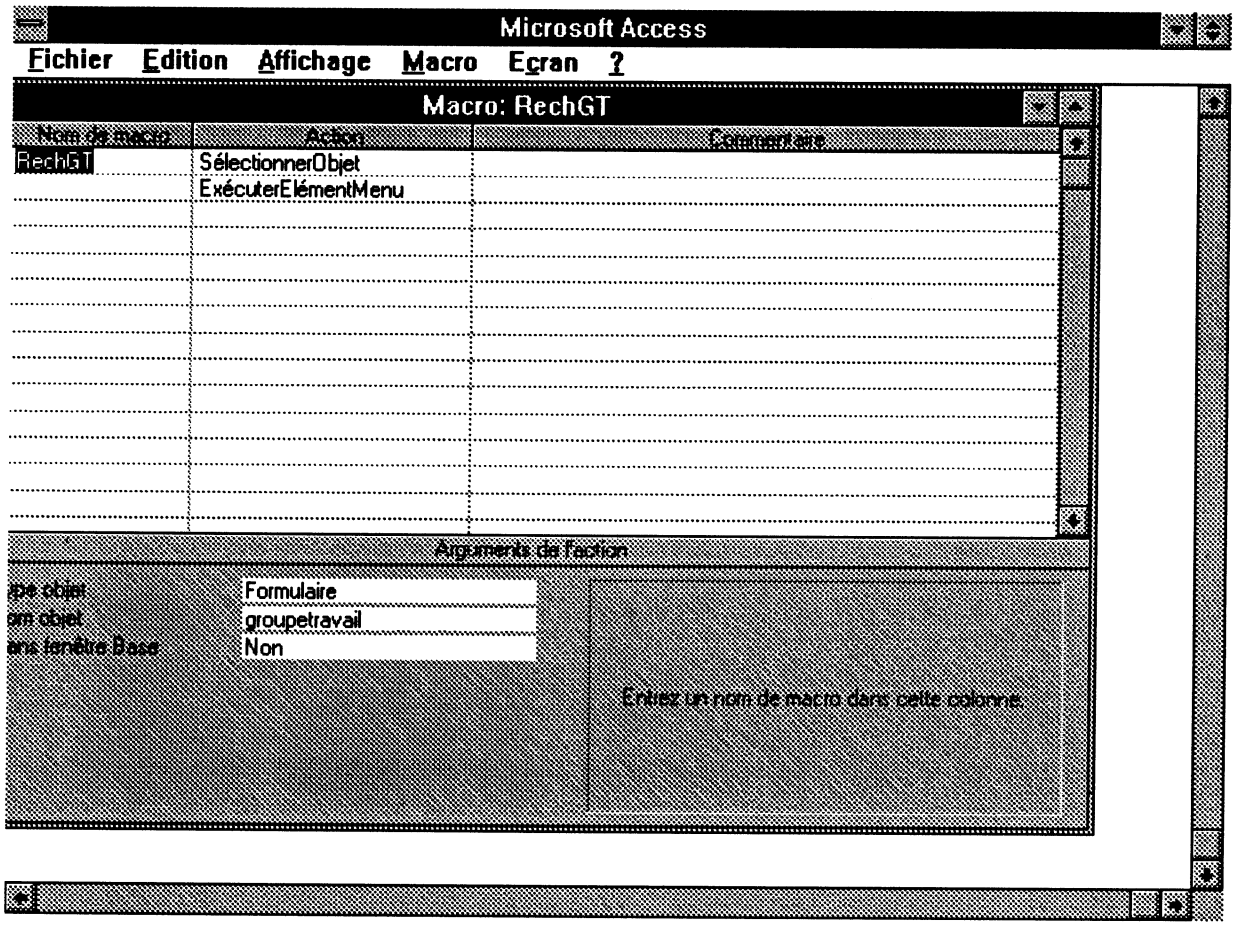
MACRO RECHDIAPO :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Diapositives.



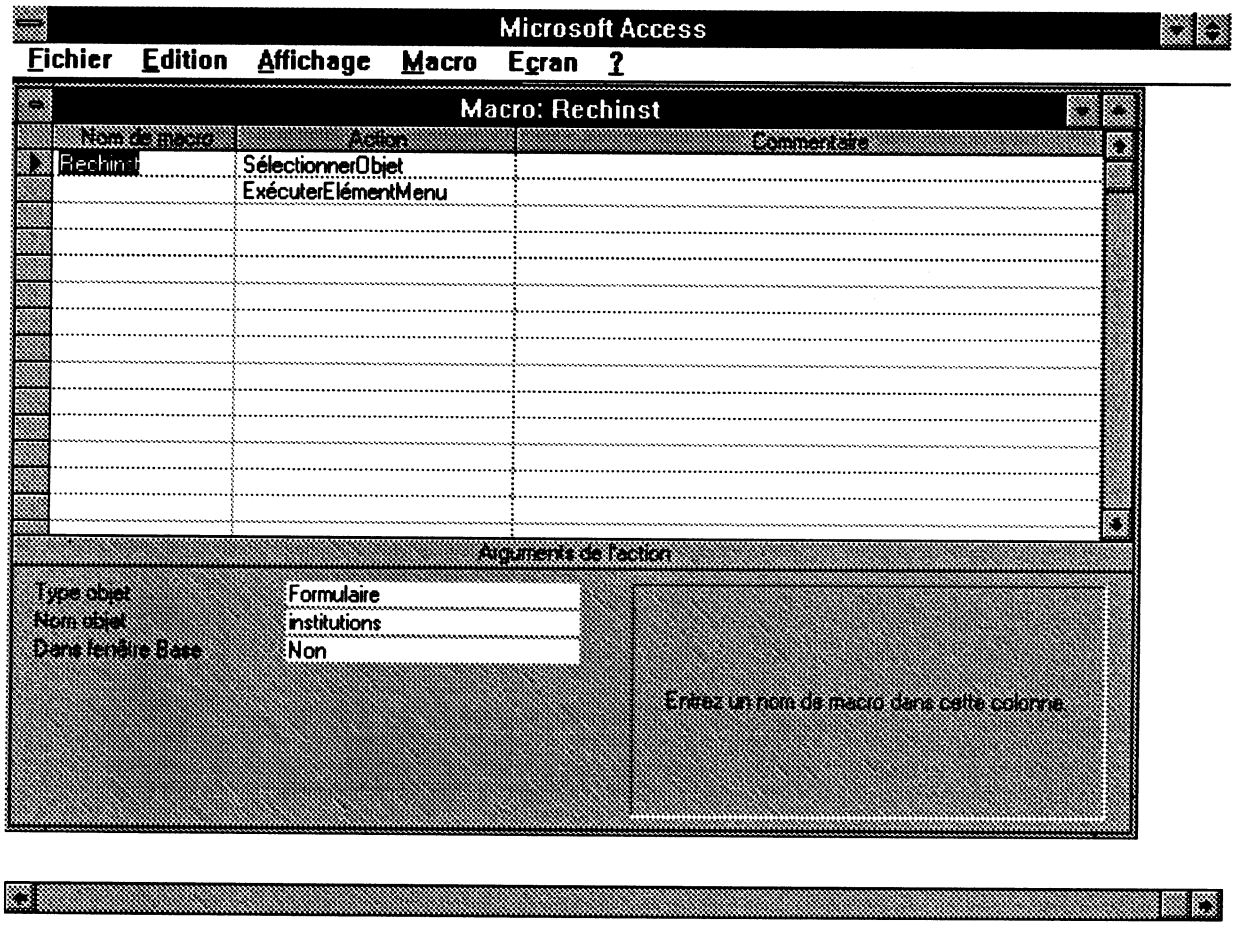
MACRO RECHGT :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Groupetravail.



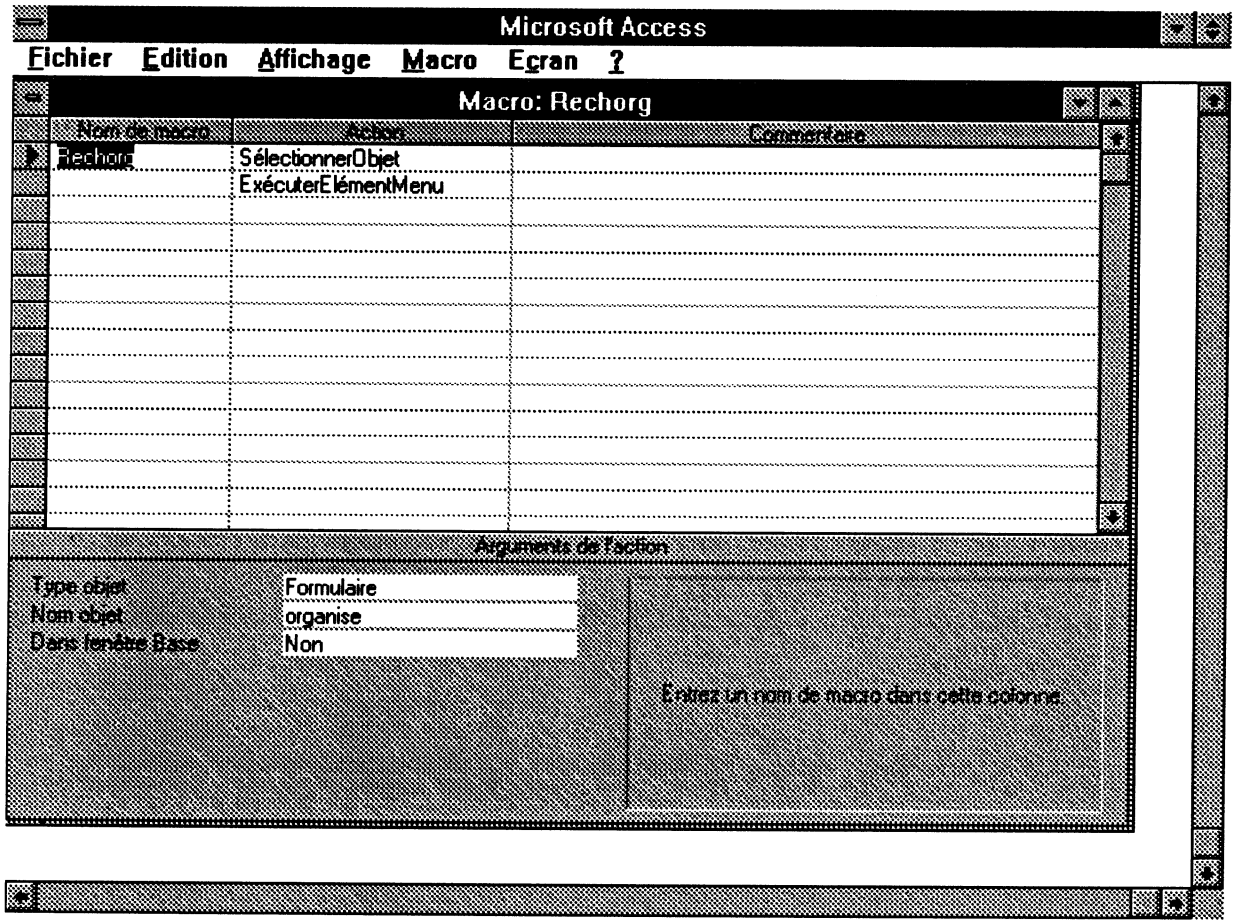
MACRO RECHINST :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Institutions.



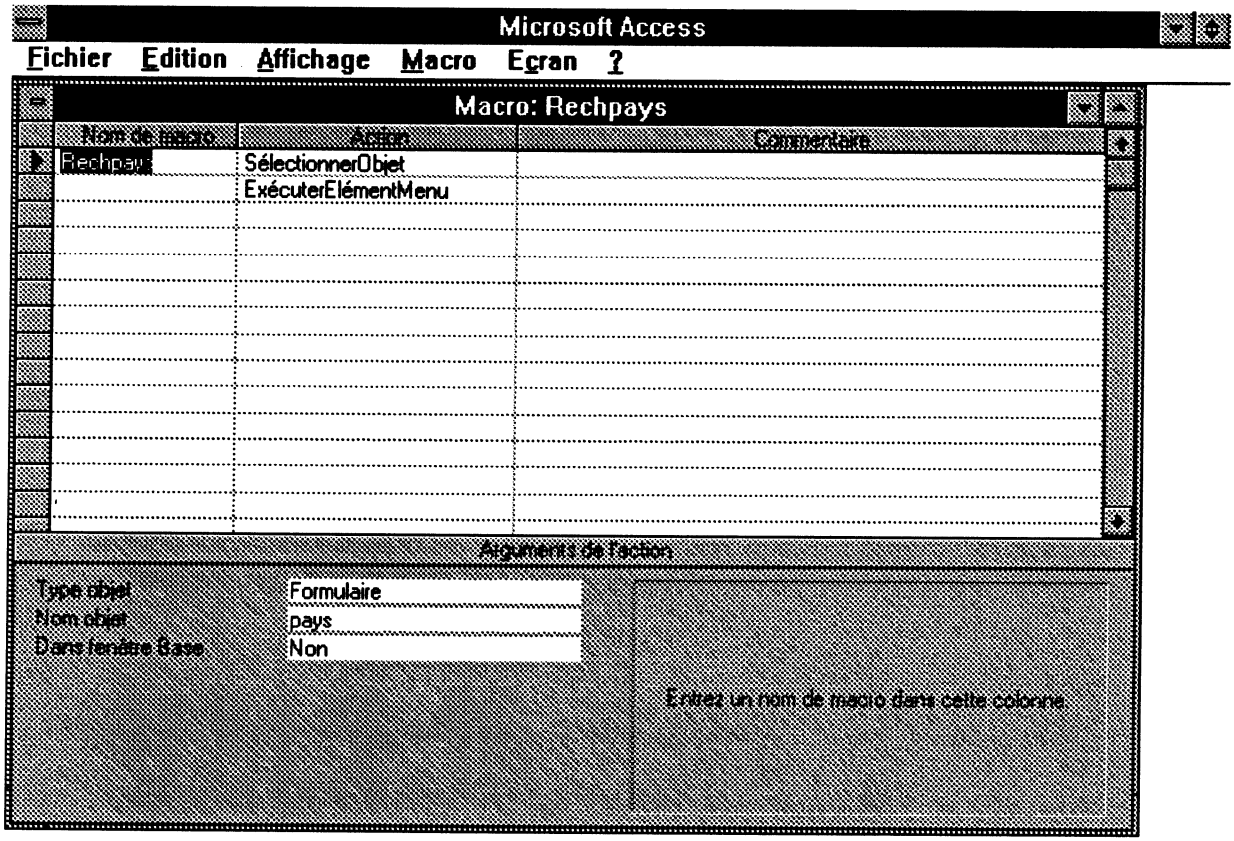
MACRO RECHORG :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Organise.



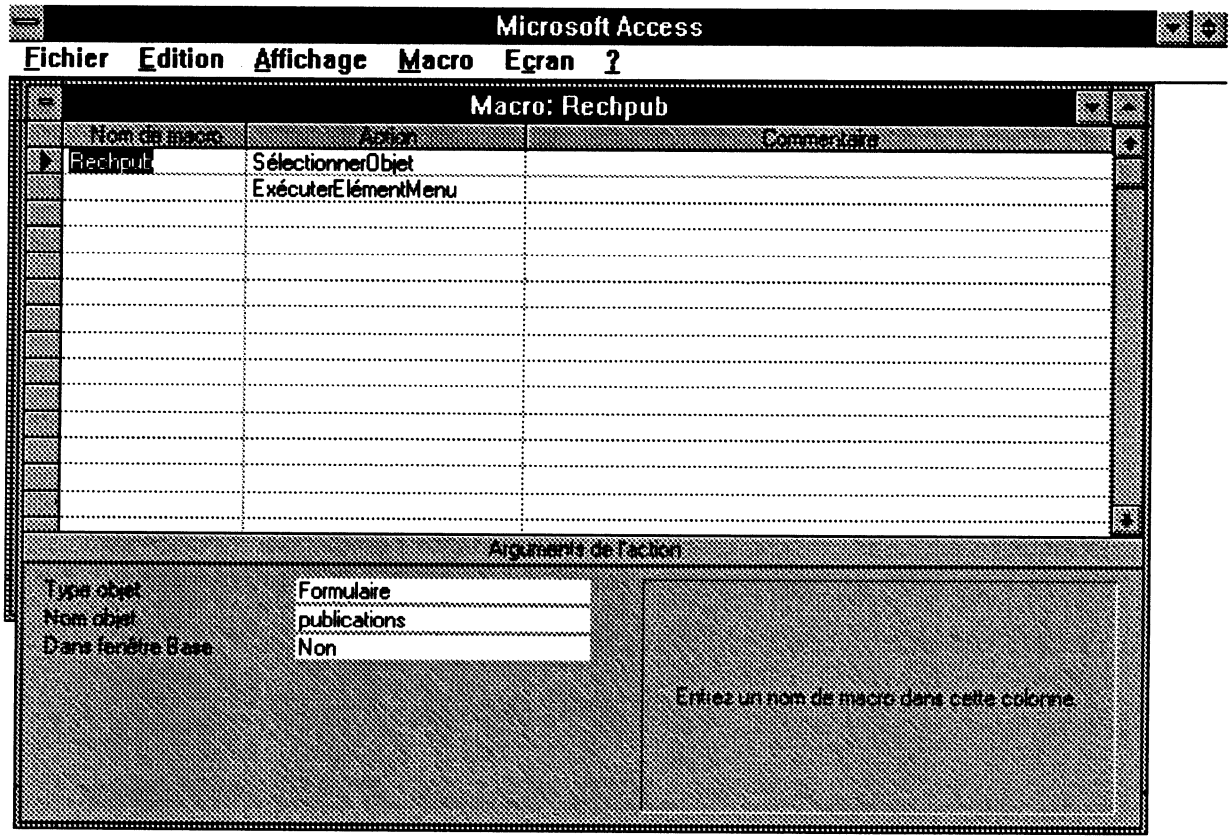
MACRO RECHPAYS :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Pays.



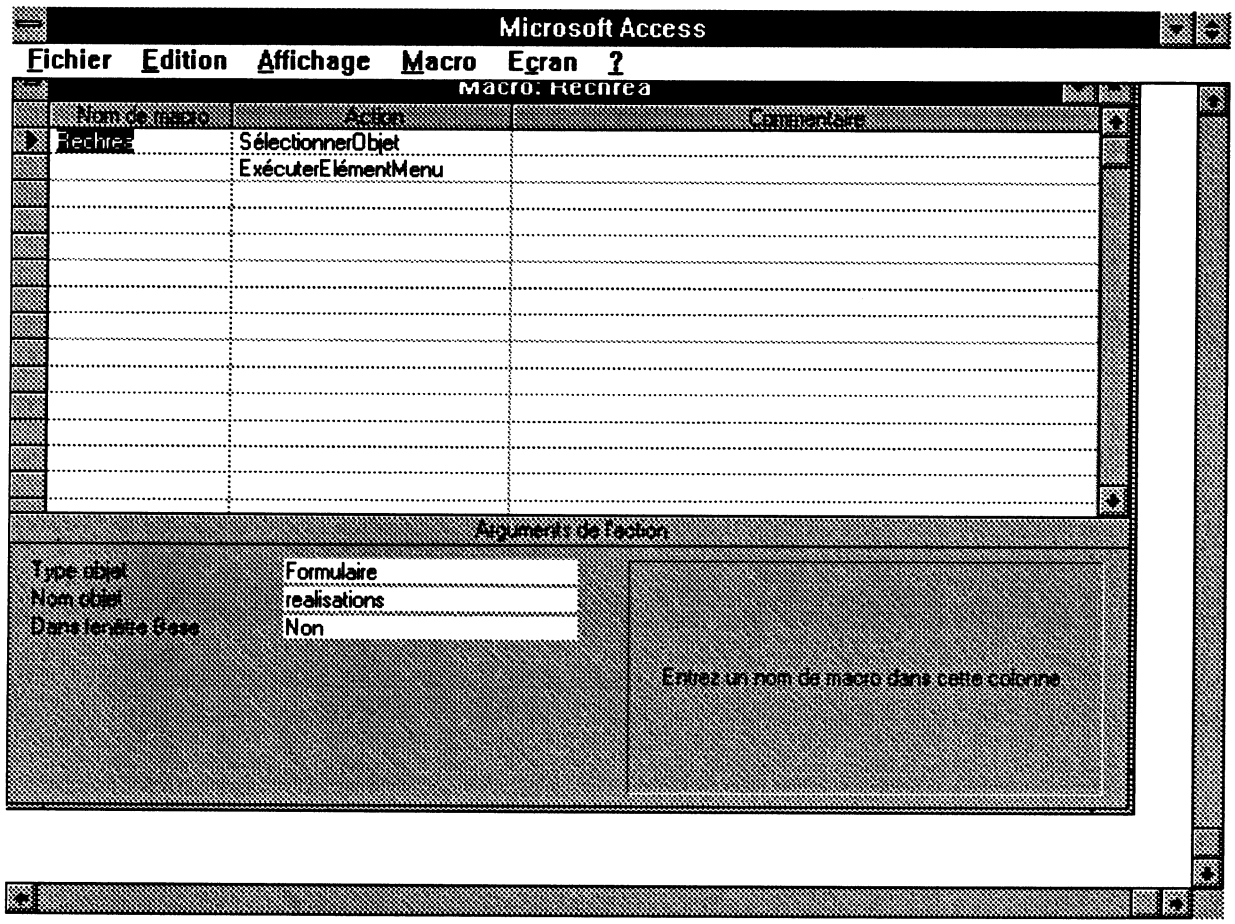
MACRO RECHPUB :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Publications.



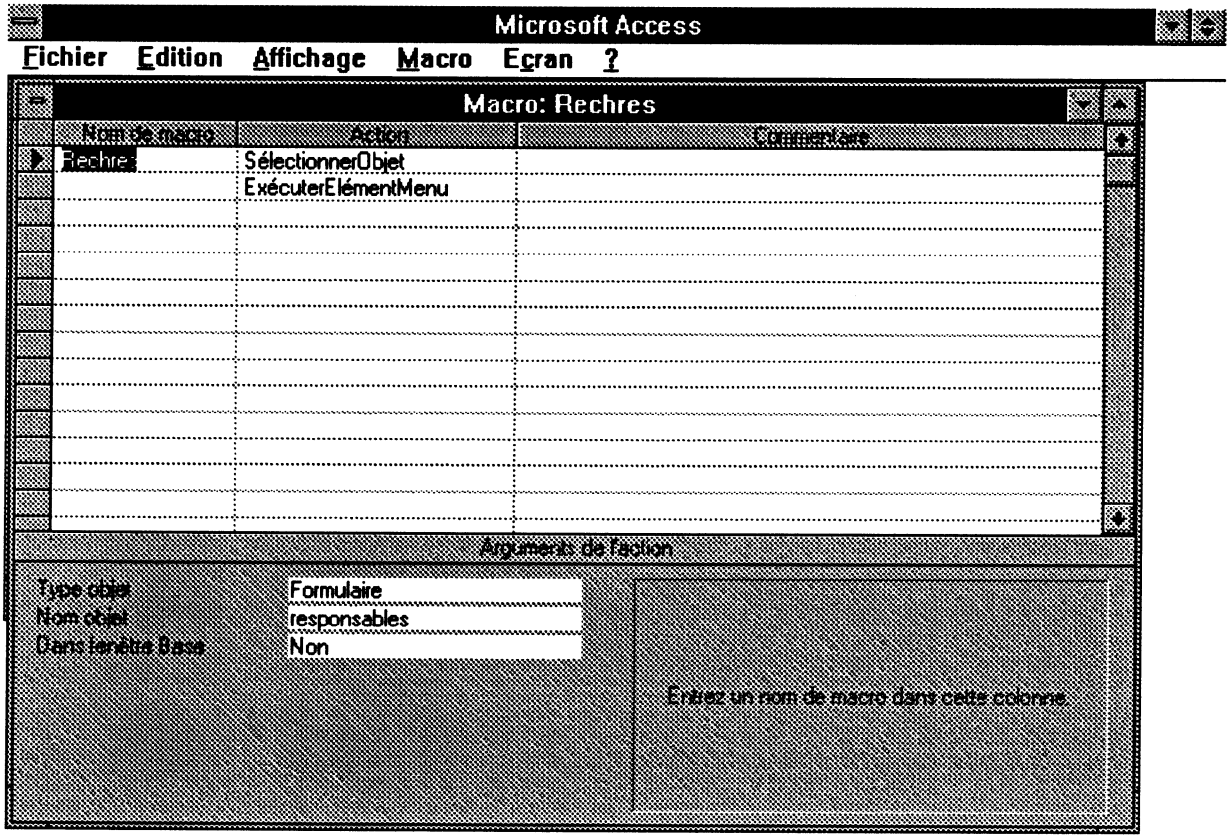
MACRO RECHREA :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Realisations.



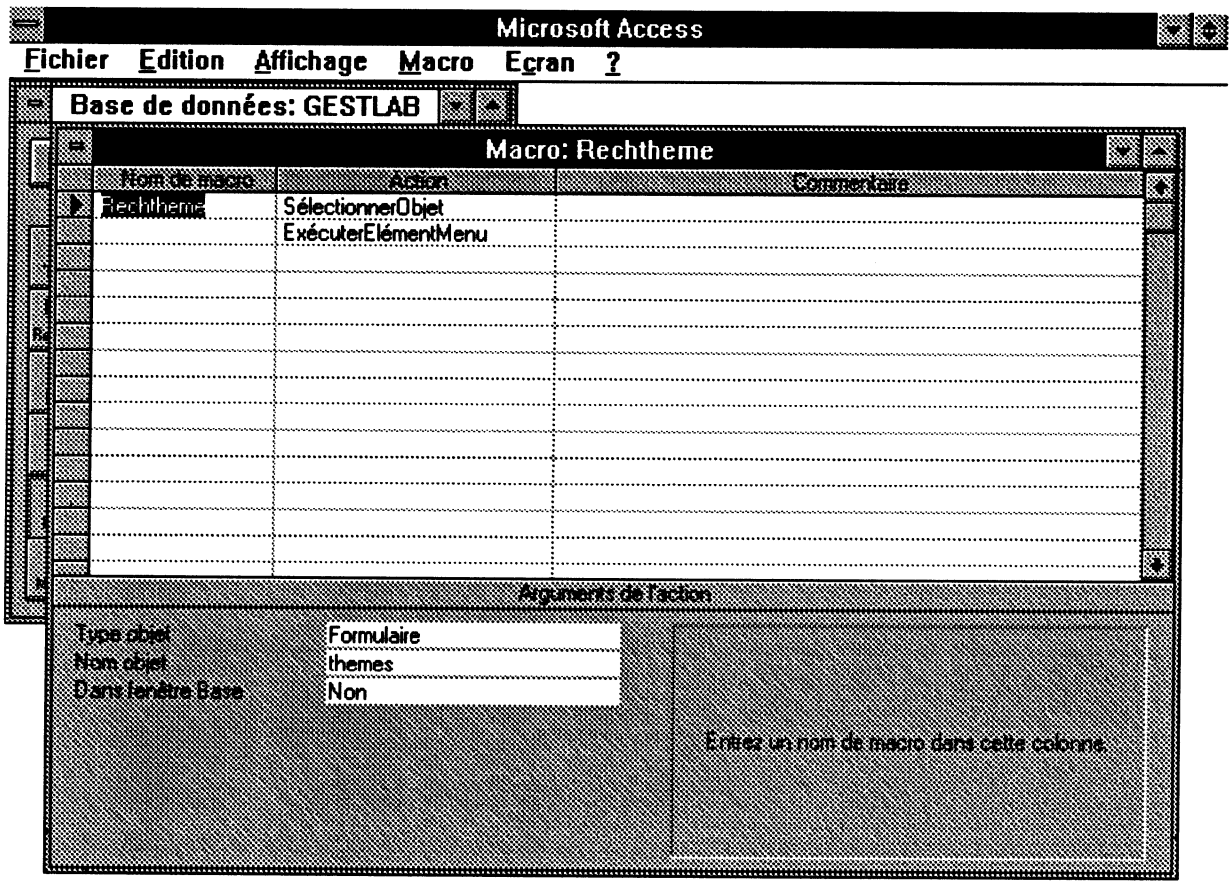
MACRO RECHRES :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire Responsables.



MACRO RECHTHEME :

Cette macro ouvre la boîte de dialogue Rechercher pour le formulaire thèmes.



LES PROGRAMMES

Afin de vous faciliter la compréhension des programmes utilisés dans l'application, nous vous présentons rapidement les principaux outils du langage Access Basic.

1 Les Modules :

Les programmes crées par l'utilisateurs sont enregistrés dans des modules. Un module contient en général plusieurs procédures utilisées indépendamment les unes par rapport aux autres. Certains éléments d'un code ne valent que pour une procédure (variables locales) tandis que d'autres valent pour l'ensemble des procédures d'un module ou pour tous les modules de la base de données (variables globales). Un module se compose d'une section de déclarations et de procédures. La section de déclarations peut être comparé à l'en-tête d'un formulaire : les éléments de cette zone se rapportent à toutes les procédures du module.

Chaque module de l'application contient les procédures qui contrôlent un formulaire donné ou ensemble de procédures communes à plusieurs formulaires.

Ces modules nous amènent à parler des variables et des procédures.

2 Les variables :

Nous avons utilisé deux types de variables dans les "codes" de l'application. Il s'agit notamment des variables locales qui sont propres à la procédure dans laquelle elles sont déclarées et les variables globales qui sont accessibles à toutes les procédures de la base de données.

2.1 Déclaration d'une variable locale :

Il existe deux modes de déclaration d'une variable locale : la déclaration directe et la déclaration indirecte.

La déclaration directe :
Elle se fait directement en affectant une variable d'une valeur sans que vous l'ayez déclaré auparavant. Dans ce cas, la déclaration se fait automatiquement par Access.
Exemple :

```
Nombre=17
```

Ici, on affecte la valeur 17 à la variable Nombre que Access identifie comme une variable de type entier.

La déclaration indirecte :
Elle se fait en déclarant tout d'abord le type de la variable puis en affectant la variable ainsi déclarée d'une valeur correspondant au type de la variable.
Exemple :

```
Dim Nombre As Integer     ' on déclare Nombre en variable de type entier
Set Nombre= 17            ' on affecte la valeur 17 à la variable Nombre
```

N.B. : L'instruction **Dim** sert à déclarer les variables et l'instruction **Set**, quant à elle, sert à affecter à ces variables des valeurs.
Il existe également l'instruction **Static** pour déclarer une variable locale. Elle n'est pas utilisée dans nos programmes.

2.2 Les variables globales :

Il existe deux sortes de variables globales qui sont les variables valables pour l'ensemble d'un module et les variables qui sont valables pour l'ensemble des modules de l'application.
Les variables valables pour un module :
Ces variables sont déclarées dans la section **Déclarations du module** en utilisant l'instruction **Dim**. Les variables ainsi déclarées sont accessibles à partir de n'importe quelle procédure du module dans lesquelles elles sont déclarées.

Les variables valables pour l'ensemble des modules :
Ces variables sont également déclarées dans la section **Déclarations** des modules en utilisant l'instruction **Global**. Ces variables sont alors valables pour l'ensemble des modules de la base de données.
Exemple :

```
Global ValRes As Integer;
```

2.3 Variables d'objet et les types d'objet :

Il existe un autre type de variable dans Access, ce sont les **variables d'objet**. Ces variables ne sont pas utilisées pour stocker des valeurs mais des objets permettant d'effectuer des manipulations dans la base de données. En effet, Access se compose de différents objets de données tels les formulaires, les tables et requêtes. Des variables d'objet peuvent êtres définies pour chacun de ces types d'objet. La définition d'une variable d'objet transforme une variable en un objet de données. Les variables d'objets utilisées dans les procédures sont :

DataBase : pour décrire un objet de type base de données;
Form : pour décrire un formulaire de la base de données active;
Control : pour décrire un champ de contrôle d'un formulaire;
Table : pour décrire un objet de type table;
Dynaset : pour décrire une table virtuelle actualisable (feuillede réponses);
Snapshot : pour décrire une table virtuelle non actualisable (instanatané);
Screen : pour le formulaire courant.

La définition des variables d'objet se fait avec les outils offerts par Basic, c'est-à-dire **Dim**, **Global** et **Static**. L'opération consiste à créer une variable avec l'instruction **Dim** et préciser le type d'objet approprié. L'instruction **Set** permet de faire, ici, le lien entre la variable d'objet et l'objet de données.
Exemples :

```
Dim Tamp As Control                  'on déclare Tamp comme un contrôle de champ
Set  Tamp= Forms![publications]![CodePu]  'Tamp reçoit le contenu du champ  CodePu
                                          ' du formulaire publications

Dim MABD As DataBase                 'on déclare MABD comme une base de données
Set MABD= CurrentDB()                'MABD reçoit la base de données actuelle

Dim LeForm As Form                   'on déclare LeForm comme un formulaire
Set LeForm= Screen.ActiveForm        'LeForm reçoit le formulaire actif à l'écran
```

D'une manière générale, une méthode d'objet correspond à un fonction ou une instruction dépendante d'un objet. Pour attribuer une méthode à un objet, la méthode est séparée de l'objet par un point.

Dim LaTable **As** *Table* 'on déclare LaTable comme une table
Dim MaBd **As** DataBase 'on déclare MaBd come une base de données
Set MaBd= *OpenDatabase* ("Gestlab.MDB") ' MaBd reçoit la BD ouverte Gestlab.MDB
Set LaTable= MaBd.*OpenTable* ("chrono") 'LaTable reçoit la table chrono de la BD ouverte

Dim MaBd **As** *DataBase*, Resul As *Dynas*et 'on déclare MaBd comme un base de donnée et 'et Resul comme un Dynaset
Set MaBd= *CurrentDB()* 'MaBd reçoit la base de données courante
Set Resul= *MaBd.CreateDynaset("Resultat Final")* 'Resul reçoit le Dynaset créé Resultat Final

Le type d'objet **Dynaset** permet tout comme le type d'objet table d'accéder aux tables Access,requêtes et tables liées. En fait, il permet de créer un extrait temporaire de la source des données en considérant que le mode feuille de données ne soit pas toujours actuel. L'ouverture s'effectue avec la méthode *CreateDynaset*.
Par contre, le type d'objet Snapshot se distingue des types Table et Dynaset par le fait que ses données sont immuables. L'ouverture générée par la méthode CreateSnapsho a pour conséquence de reproduire l'image de la source des données (instantané).

Dim BD **As** *DataBase*
Dim Nom As *Snapshot*
Set BD= *OpenDatabase ("Gestlab.MDB")*
Set Nom=*BD.CreateSnapshot("Noms")*

Les principales méthodes utilisées dans pour réaliser l'application sont :

AddNew : pour ajouter un enregistrement dans une table ouverte;
Close : pour fermer la base de données ouverte;
CreateDynaset : pour créer le Dynaset dont le nom est passé en paramètre;
CreateSnapshot : pour créer le Snapshot dont le nom est entré en paramètre;
FindFirst : cherche le premier enregistrement qui répond au critère spécifié;
OpenDatabase : ouvre la base de données dont le nom est passé en paramètre
OpenTable : ouvre la table dont le nom pasé en paramètre;
MoveNext : pour passer à l'enregistrement suivant l'enregistrement courant dans la table ouverte;
Update : pour valider une saisie saisie dans la table ouverte.

2.4 Les Types de données des variables:

Sous Access, on distingue les types de données suivants : *Integer*, *Long* , *Single*, *Double*, *Currency*, *String* et *Variant*. Ces types de données permettent de définir la nature des données pour lesquelles une variable est déclarée.
Les types de données utilisées dans l'application sont :

Variant : c'est le type de données principal et en même temps le type par défaut. Il peut contenir des valeurs numériques, Date/Heure,des chaînes de caractères mais également la valeur Null. Cela signifie que ce type de données peut contenir tous les formats de données;
Integer : pour les nombres entiers compris entre -32768 et 32767;
String : pour le type chaîne de caractères. Ce type peut contenir jusqu'à 65635 caractères.

Ces types de données sont également utilisées comme argument des fonctions.

Exemple : Function OuvrirLeFormM (LeNom As *String*)

3 Les fonctions :

3.1 Les conventions :

.Sous Access Basic les fonctions ont l'ossature suivante :
-si il s'agit d'une fonction

Function XX ()
End Function

-si il s'agit d'une sous-procédure

Sub YY ()
End Sub

Les procédures **Sub** ne ne sont utilisées que pour exécuter qu'une opération et elles ne retourne aucune valeur.

XX et YY correspondent ici à des noms quelconques de fonction et de sous-procédure.

.Les commentaires sont introduits par des apostrophes dans les fonctions.

'Cette fonction retourne la valeur du contrôle actif

.Dans les traitements des erreurs, on n'utilise pas des numéros de lignes mais des étiquettes.

```
 On Error GoTo Traitement
...
Traitement :
Print Error$
```

3.2 Appel d'actions et de fonctions et de sous-procédures :

Appel des actions :

Pour appeler une action ou une macro dans une fonction, on utilise l'instruction **DoCmd**. La syntaxe d'utilisation de cette instruction est :

DoCmd NomAction [Liste arguments]

Comme nom d'action, on indique toujours le nom anglais d'une action de macro, le nombre d'arguments dépend de l'action utilisée. Les principales actions utilisées sont :

-GoToControl : sélectionne un contrôle dans l'objet ouvert. On utilise cette action également pour comparer les données.

syntaxe: DoCmd GoToControl Nom control

Exemple : **DoCmd** *GoToControl* "CodeRe" 'on atteind le contrôle CodeRe

-Close : ferme la fenêtre désignée ou la fenêtre active.
syntaxe : DoCmd Close [Type d'objet], [Nom objet]

Exemple : **DoCmd** *Close* A_Form, LeNom.FormName 'on ferme le formulaire de nom Lenom

-GoToRecord : transforme l'enregistrement indiqué en enregistrement courant. Il est utitlisé pour atteindre directement un enregistrement donné. Notamment, le premier et le dernier enregistrement de la table ouverte mais aussi l'enregistrement suivant ou précédent l'enregistrement actif.
syntaxe : DoCmd GoToRecord [Type d'objet], [Nom d'objet], [Enregistrement], [Référence]

exemple : **DoCmd** *GoToRecord* A_Form, LeForm.FormName, A_LAST 'on atteind le 'dernier enregistrement de la table liée au formulaire de nom LeForm.

-FindRecord : recherche un enregistrement correspondant aux critères indiqués. Il est utilisé pour des recherches fréquentes sur des enregistrements correspondant à certains critères.
syntaxe : DoCmd FindRecord Rechercher, [Où], [Majuscule/Miniscule], [Direction], [Comme formaté], [Rechercher-dans], [Trouver-premier]

DoCmd *FindRecord* ValRes, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
'on retrouve l'enregistrement qu correspond à ValRes dans le **Champ entier**, sans tenir compte de la différence entre majuscule et miniscule, d'effectuer la recherche vers le bas, de ne rechercher que les valeurs enregistrées, dans le **Champ en cours**, et de commencer la recherche à partir de l'enregistrement courant.

-DoMenuItem : exécute une commande de menu Access.
syntaxe : DoCmd DoMenuItem Barre menus, Nom menu, Commande, [Commande secondaire]

DoCmd *DoMenuItem* A_FORMBAR, A_EDITMENU, A_UNDOFIELD
'on appelle, ici, la commande Annuler de la barre de menu Edit

-OpenForm : ouvre un formulaire dans le mode souhaité.
syntaxe : DoCmd OpenForm Nom formulaire, [affichage], [Nom filtre], [Condition Where], [Mode données], [Mode fenêtre]

DoCmd *OpenForm* LeForm, A_NORMAL, , ,A_EDIT 'on ouvre le formulaire de nom LeForm en mode formulaire et avec autorisation de modifier les données

-RunSQL : permet d' exécuter une requête action à l'aide d'une instruction SQL. Cette commande est utilisée dans le programme pour effacer l'enregistrement actif.
syntaxe : DoCmd RunSQL Instruction SQL

DoCmd *RunSQL* "DELETE FROM diapositives WHERE diapositives.codeDi =" & ValDi & ";"

Appel d'une fonctions et sous-procédures:

-l'appel d'une fonction s'effectue en affectant la fonction à une variable :
i=OuvertureFormM(LeNom) 'on récupère dans i la valeur de la fonction OuvertureFormM()

-l'appel d'une sous-procédure se fait en utilisant l'instruction Call :

Call MessValider 'on transfère le contrôle du programme à la sous-procédure MessValider

3.3 L'accès au contenu d'un champ de contrôle :

Il est utilisé deux méthodes pour aaccéder au contenu d'un champ de contrôle : la méthode direct qui utilise le nom d'objet et la méthode indirecte qui utilise l'action GoToControl.

L'accès direct :
Pour lire directement le contenu d'un champ de contrôle d'un formulaire la syntaxe est la suivante :

Descripteur objet![Nom objet]! NomChampControl

Dim Nom As String 'on déclare Nom en chaîne de caractères
Set Nom= *Forms!responsable!Nom_Prénom* 'Nom reçoit le contenu du contrôle Nom_Prénom du formulaire Rresponsable

L'accès indirect :
La méthode dans ce cas consiste à appeler l'action **GoToControl** pour atteindre directement le contrôle désiré, puis de lire le contenu de ce contrôle-ci.

DoCmd *GoToControl* "CodeRe" 'on se positionne au contrôle CodeRe
ValCodeR= *Screen.ActiveControl* 'ValCodeR reçoit le contenu du contrôle CodeRe

3.4 La gestion des erreurs :

La gestion des erreurs se fait à l'aide de l'instruction **On Error** et de la fontion **Err**.
La ligne d'instructions **On Error Resume Next** : spécifie qu'en cas d'erreur d'exécution, le contrôle du programme passe à l'instruction suivante. On utilise la fonction Err dans les lignes de code ultérieures pour obtenir le numéro d'erreur d'exécution.

On Error Resume Next
...
If **Err** Then MsgBox " Impossible d'accéder au dernier enregistrement"

3.5 L'appel des boîte de message :

L'appel des boîtes de message se fait à l'aide de la fonction **MsgBox** et l'instruction **MsgBox**. La différence entre ces deux éléments est que l'instruction ne renvoie pas de valeur et les arguments de cette dernière ne sont pas mis entre parenthèses.

La syntaxe : **MsgBox** (Texte message, [Nombre et Aspect des boutons], [Titre message])

Les différentes utilisation de **MsgBox** dans l'application sont :

MsgBox "une erreur est survenue pendant la fermeture" 'on affiche le message d'erreur entre""

Reponse= **MsgBox** (Msg, 36, "Avertissement")
If Reponse=6 Then ...

Dans l'exemple ci-dessus, on demande d'afficher une boîte de message d'avertissement avec les boutons de commande Oui/Non. Dans la variable Reponse, on récupère la valeur du bouton de commande activé. Puis, on teste si le bouton activé est le bouton Oui. En effet, la fonction **MsgBox** retourne des valeurs de 1 à 7 qui renseigne sur le bouton appuyé. *6 correspond au bouton Oui et 7 au bouton Non.*
L'argument *36* (32+4) demande à Access d'afficher dans la boîte de message *une icône point d'interrogation (32)* et *les boutons de commande Oui/Non (4).*

Rep= **MsgBox** (Msg, 48, "Saisie du nom du domaine") 'on demande d'afficher une boîte d'avertissement avec une icône point d'exclamation (!) dont le tite est le texte entre "".

i=**MsgBox** (Msg, 64, "Information") 'on affiche une boîte de message d'information avec une icône représentant la lettre i et dont le texte est le contenu de la variable Msg.

Les fonctions Access Basic :

Nous avons utilisé certaines fonctions prédéfinies dans la mise en oeuvre du logiciel de gestion du laboratoire. Le principales fonctions sont :

-**DLookup** (expression, domaine, [critère]) : cette fonction retourne la valeur d'un champ dans une doaine (table) déterminé.

X= **DLookup** ("[Nom du messager]","Messages", "[N° messager]=1")
Dans cet exemple, on retourne le nom du messager de la table Messagers dont le numéro et égale à 1.

-**DMax** (expression, domaine, [critère]) : indique la valeur maximale parmi les enregistrements d'un domaine (table).

X= **DMax** ("[codeDi]", "diapositives") 'on récupère le plus grand codeDi de la table diapositives

-**IsNull** (Variant): retourne une valeur qui détermine si le paramètre indiqué sous forme de type de données Variant est de type Null. La variable de type Null indique la variable ne contient aucune donnée connue.

If **IsNull** (codePu) Then ... 'on teste si le codePu est Null.

-**InputBox** (Texte,[Titre],[défaut],[[xpos],[ypos]]) : affiche une boîte de dialogue qui permet la saisie de données de type Variant.

Reponse=**InputBox** (Msg,"Saisie d'un nouveau contrat") 'on affiche une boîte de saisie de contrat.

-**EOF** : indique si la fin du fichier est atteinte

```
Do Until Resultat.EOF
...
Loop
```

-**Err** : indique le code d'erruer.

If **Err** Then MsgBox "Impossible d'accéder au dernier enregistrement"

-**Error** : affiche le message d'erreur correspondant au code d'erreur indiqué sous forme de données Variant.

On Error Resume Next

-**OpenDatabase** (base de données, [exclusif, [lecture seule]]) : ouvre la base de données spécifiée.

```
Dim AutreBD As Database
Set AutreBD= OpenDatabase (Chemin)
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCommun


Global Tmode As String  ' Consultation et modification de certaines données
                        ' Remplissez votre formulaire puis validez par OK

Global Mode As Integer  ' 0 -> Consultation
                        ' 1 -> Nouvel enregistrement


Global FermeCorrecte As Integer  ' 1 si la fermeture du formulaire se fait
                                 ' par les boutons de commandes

Global ValCodeR As Integer
Global ValNom As String
Global ValNum As Integer
Global ValCo As String
Global ValTh As String
Global ValPu As String
Global ValCh As String
Global ValDi As Integer
Global ValGT As String
Global ValIns As Integer
Global ValRes As Integer
Global ValPa As String

Function ActionQuitter ()

' Cette fonction permet de retourner au menu général

'*****************************************************************

i = OuvertureFormM("menu general")

End Function
```

```
Function Annule (LeNom As String)

' Cette fonction permet de retourner au premier enregistrement du formulaire dont le nom
' est passé en paramètre sans créer de nouvel enregistrement

'*****************************************************************************
On Error Resume Next

Mode = 0
i = OuvertureFormM(LeNom)
End Function

Function AnnulerSaisie (LeNom As String)

' Cette fonction permet de retourner au formulaire passé en
' paramètre sur l'enregistrement sur lequel ont été portées des
' modifications sans toute fois les enregistrer.

'*********************************************************************

On Error Resume Next
Dim Tamp As Control

If Mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Annulez-vous l'opération ?"
  Reponse = MsgBox(Msg, 36)
  If Reponse = 7 Then
    Exit Function
  Else
    Set Tamp = Screen.ActiveControl
    DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_UNDOFIELD
    DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
  End If
End If

End Function
```

```
Function CalculNvNum (NTable As String, NControle As String) As Integer

    ' Cette fonction retourne pour le controle de nom NControle
    ' de la table de nom NTable un numéro libre
    ' Le Controle de nom NControle doit être de type Entier

'*********************************************************************
On Error Resume Next

Dim x As Integer
Dim Texte As String
Dim Nbre As Integer

Texte = "[" & NControle & "]"
Nbre = DCount(Texte, NTable)

If Nbre = 0 Then

   CalculNvNum = 1

Else
   x = DMax(Texte, NTable)
   CalculNvNum = x + 1
End If
End Function

Function FermerFormActif () As Integer

  'Ferme le formulaire Actif, en cas d'erreur retourne 0
  'sinon retourne 1

'*********************************************************************

On Error Resume Next

Dim Leform As Form
Dim LeNomduForm As String

Set Leform = Screen.ActiveForm
LeNomduForm = Leform.FormName

DoCmd Close A_FORM, LeNomduForm

If (Err) Then

   FermerFormActif = 0
Else
   FermerFormActif = 1

End If
```

```
  End Function
Function InitMode ()

' Rempli la variable globale TMode du texte désignant le mode d'ouverture du formulaire

'*****************************************************************************
**************

If Mode = 0 Then
   Tmode = "Consultation et modification de certaines données"
Else Tmode = "Remplissez votre formulaire puis validez avec le bouton 'ENREGISTRER'"
End If
End Function

Function InitVarG ()

'Cette fonction initialise les variables globales

*************************************************************

Mode = 0
ValRes = 0
ValIns = 0
ValDi = 0
ValCodeR = 0

End Function

Function OkSauve ()

' Cette fonction sauvegarde les modifications de données du formulaire actif

'*****************************************************************************
*********


Dim Msg As String

    Msg = "Vos modifications sont automatiquement sauvegardées"
    i = MsgBox(Msg, 64, "Information")

End Function
```

```
Function OuvertureFormM (LeNom As String) As Integer

' Gère l'ouverture du formulaire dont le nom est passé en paramètre
' Ferme le formulaire actif et ouvre le formulaire demandé en mode modification

'*************************************************************************
***************

On Error Resume Next

Dim i As Integer

i = FermerFormActif()

If i <> 1 Then
   MsgBox "Une erreur est survenue pendant la fermeture"
   OuvertureFormM = 0
   Err = 0
Else
   i = OuvrirLeFormM(LeNom)
   If i <> 1 Then
      MsgBox "Une erreur est survenue pendant l'ouverture de " & """"& LeNom & """"
      OuvertureFormM = 0
   Else
      OuvertureFormM = 1
   End If


End If

End Function
```

```
Function OuvreMemeCritere (LeNom As String, LeControle As String)

' Cette fonction permet d'ouvrir le formulaire de nom LeNom sur
' l'enregistrement dont le controle de nom LeControle a la même valeur
' que ce même controle dans l'enregistrement actif du formulaire appelant.

'*************************************************************************

On Error Resume Next
If Mode = 1 Then

   Call MessValider
Else

   DoCmd GoToControl LeControle
   Tamp = Screen.ActiveControl

   If Not IsNull(Tamp) Then
     i = OuvertureFormM(LeNom)
     DoCmd GoToControl LeControle
     DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   Else
     i = OuvertureFormM(LeNom)
   End If

End If

End Function
```

```
Function OuvrirLeFormM (LeNom As String)

    'Cette Fonction ouvre le formulaire dont le nom est passé en paramètre
    'en mode Modification
    'Si ce nom est vide ou NULL retourne 0 (Il y a eu ERREUR )
    'S'il y a une erreur d'ouverture retourne 0 (Il y a eu ERREUR )
    'Sinon retourne 1 (Il n'y a pas eu d'ERREUR )

'*********************************************************************

On Error Resume Next

If IsNull(LeNom) Or LeNom = "" Then
    MsgBox "Le nom passé en paramêtre est vide ou nul"
    OuvrirLeForm = 0

Else
    DoCmd OpenForm LeNom, A_NORMAL, , , A_EDIT
    If (Err) Then
        OuvrirLeFormM = 0
    Else
        OuvrirLeFormM = 1
    End If


End If

End Function

Function Recup () As Variant

' Cette fonction retourne la valeur du contrôle actif

'*********************************************************************

Dim Vamp As Control

Set Vamp = Screen.ActiveControl
Recup = Vamp

End Function
```

```
Function RecupCodeCh ()

' Cette fonction récupère dans la variable globale ValCh la valeur
' du contrôle actif

'*****************************************************************

On Error Resume Next

ValCh = Recup()

End Function

Function RecupCodeCo ()

' Cette fonction récupère dans la variable globale ValCo la valeur
' du contrôle actif

'*****************************************************************

On Error Resume Next

ValCo = Recup()

End Function

Function RecupCodeDi ()

' Cette fonction récupère dans la variable globale ValDi la valeur
' du contrôle actif

'*****************************************************************

On Error Resume Next

ValDi = Recup()

End Function
```

```
Function RecupCodePu ()

' Cette fonction récupère dans la variable globale ValPu la valeur
' du contrôle actif

'*********************************************************************

On Error Resume Next

ValPu = Recup()


End Function

Function RecupCodeR ()

' Cette fonction récupère dans la variable globale ValCodeR la valeur
' du contrôle actif

'*********************************************************************

On Error Resume Next

ValCodeR = Recup()

End Function

Function RecupCodeTh ()

' Cette fonction récupère dans la variable globale ValTh la valeur
' du contrôle actif

'*********************************************************************

On Error Resume Next

ValTh = Recup()

End Function
```

```
Function RecupValIns ()

' Cette fonction récupère dans la variable globale ValIns la valeur
' du contrôle actif

'*********************************************************************

On Error Resume Next

ValIns = Recup()

End Function

Function RecupValRes ()

' Cette fonction récupère dans la variable globale ValRes la valeur
' du contrôle actif

'*********************************************************************

On Error Resume Next

ValRes = Recup()

End Function

Function RetourMode () As String

' Retourne la valeur de la variable globale TMode

'*****************************************************************************
***************

On Error Resume Next

RetourMode = Tmode

End Function
```

```
Function VerifFermeture ()

'Vérifie si le formulaire a été fermé à partir des
'boutons de commandes

'*********************************************************************

On Error Resume Next

If FermeCorrecte = 0 Then
        MsgBox "Pour sortir de l'application cliquer sur QUITTER"
        DoCmd CancelEvent
End If

End Function

Function OuvresiMode0 (LeNom As String)

' Ouvre le formulare passé en paramètre en mode 0,sur le
' premier enregistrement de la table liée au formulaire actif

'**********************************************************************

On Error Resume Next

If Mode = 1 Then
  Call MessValider
Else i = OuvertureFormM(LeNom)
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCMessages contient les messages communs aux formulaires de l'application

Function ConfirmationNv (FinTexte As String) As Integer

  ' Cette fonction retourne 1 si l'utilisateur a confirmé la création
  ' 0 sinon

'*********************************************************************

Dim MSG As String
Dim Reponse As Integer

MSG = "Vous voulez créer un nouvel enregistrement " & """" & FinTexte & """"
Reponse = MsgBox(MSG, 36, "Avertissement")
                '4 affichage des boutons Oui Non
                '32 affichage de l'icône Avertissement "?"

If Reponse = 6 Then   ' reponse = Oui

    ConfirmationNv = 1
Else
    ConfirmationNv = 0
End If

End Function

Sub ExistePas (NomTable As String)

'*****************************************************************

On Error Resume Next

MsgBox " La Table " & """" & NomTable & """" & "n'existe pas dans votre Base de Données.
Impossible de continuer."

End Sub


Sub MessErreurCrea ()

'Retourne un message d'erreur

'*************************************************************************

MSG = "Erreur lors de la création du nouvel enregistrement"
Rep = MsgBox(MSG, 48, "Avertissement")

End Sub
```

```
Function MessAide (I As String)

' Cette fonction permet d'afficher des messages d'aide

'*****************************************************************************

Dim MSG As String
Dim titre As String

Select Case I

Case "PUB"
   MSG = "Ce formulaire permet la modification des caractéristiques d'une PUBLICATION et
la création d'un nouvelle PUBLICATION dans la base de données"
   titre = " LE FORMULAIRE PUBLICATIONS"

Case "CTR"
   MSG = "Ce formulaire permet la modification des caractéristiques d'un CONTRAT et la
création d'un nouveau CONTRAT dans la base de données"
   titre = "LE FORMULAIRE CONTRAT"

Case "CHR"
   MSG = "Ce formulaire permet la modification de la description d'un CHRONO et la
création d'un nouveau CHRONO dans la base de données"
   titre = "LE FORMULAIRE CHRONO"

Case "DIA"
   MSG = "Ce formulaire permet la modification des caractéristiques des DIAPOSITIVES et
la création d'une nouvelle DIAPOSITIVE dans la base de données"
   titre = "LE FORMULAIRE DIAPOSITIVES"

Case "INS"
   MSG = "Ce formulaire permet la modification des caractéristiques d'une INSTITUTION et
la création d'une nouvelle INSTITUTION partenaire dans la base de données"
   titre = "LE FORMULAIRE INSTITUTIONS"

Case "CNT"
   MSG = "Ce formulaire traduit le fait qu'une publication contient une diapositive. Il permet
de faire des modifications et la saisie des PUBLICATIONS et des DIAPOSITIVES que celles-
ci contiennent"
   titre = "LE FORMULAIRE CONTIENT"

Case "THM"
   MSG = "Ce formulaire permet la modification des caractéristiques d'un THEME et la
création d'une nouveau THEME dans la base de données"
   titre = "LE FORMULAIRE THEME"

Case "ORG"
   MSG = "Ce formulaire traduit le fait qu'un chercheur dépendant d'un organisme effectue des
travaux pour le laboratoire. Ce formulaire permet d'energistrer les RESPONSABLES
```

dépendant d'une INSTITUTION qui effectuent des travaux pour le laboratoire ces INSTITUTIONS tout comme les REALISATIONS faites par ces RESPONSABLES"
titre = "LE FORMULAIRE ORGANISE"

Case "ASS"
MSG = "Ce formulaire traduit le fait qu'un contrat commande une réalisation qui porte sur un thème de recherche et que cette réalisation donnera lieu a une publication. Ce formulaire permet la modification et la saisie du CONTRATS, des REALISATIONS issues de ces CONTRATS, des PUBLICATIONS relatant ces REALISATIONS et enfin les THEMES de recherche de ces PUBLICATIONS"
titre = "LE FORMULAIRE ASSOCIE"

Case "REA"
MSG = "Ce formulaire permet la modification des caractéristiques d'une REALISATION et la création d'une nouvelle REALISATION dans la base de données"
titre = "LE FORMULAIRE REALISATIONS"

Case "RES"
MSG = "Ce formulaire permet la modification des caractéristiques d'un RESPONSABLE et l'introduction d'un nouveau RESPONSABLE dans la base de données"
titre = "LE FORMULAIRE RESPONSABLE"

Case "GTR"
MSG = "Ce formulaire permet la modification des caractéristiques d'un GROUPE de TRAVAIL et la création d'un nouveau GROUPE de TRAVAIL dans la base de données"
titre = "LE FORMULAIRE GROUPETRAVAIL"

Case "PAS"
MSG = "Ce formulaire permet la modification des caractéristiques d'un PAYS et la création d'un nouveau PAYS dans la base de données"
titre = "LE FORMULAIRE PAYS"

Case "STAT"
MSG = "Ce formulaire vous indique le nombre d'enregistrements des tables 'communes' de votre base de données, ainsi que le volume en Kilo-octets de celle-ci."
titre = "Nombre d'enregistrements des tables communes"

Case Else
MSG = "Aide non rédigée"

End Select

Reponse = MsgBox(MSG, 64, titre)

End Function

Sub MessValider ()

```
'Retourne un message de validatio de l'opération effectuée

'*****************************************************************************
On Error Resume Next

MSG = "Vous devez d'abord valider votre formulaire, en cliquant sur enregistrer"
I = MsgBox(MSG, 64, "Information")

End Sub
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes


'ModCTransfert contient les fonctions qui gèrent l'installation de la base de données

Global chemin As String  ' Contient le chemin d'accés à la BD où se trouvent
                         ' les tables


Global UneTableQuiManque As String


Function ActionTC ()

' Cette fonction permet d'ouvrir le Menu des formulaires si toutes les
' tables attachées sont présentes. Sinon propose leur attachement.

'*********************************************************************

On Error Resume Next

Dim MaBD As Database, gl As Snapshot

Set MaBD = CurrentDB()
Set gl = MaBD.ListTables()

i = VerifPresence(gl)

If i = 1 Then
   i = OuvertureFormM("Menu des formulaires")

Else
   MsgBox " LaTable " & """" & UneTableQuiManque & """" & " n'est plus attachée. Vous
devez refaire l'installation."
   i = DetacherTables()
   i = Princ()
End If

End Function
```

```
Function Attacher (nomtable As String)

' Cette fonction permet d'attacher la table de nom "NomTable"
' (en gardant le même nom) qui se trouve dans la BD de chemin "Chemin".
' (Var Globale). Retourne 1 si tout c'est bien passé, 0  sinon

'*****************************************************************

On Error Resume Next

DoCmd TransferDatabase A_ATTACH, "Microsoft Access", chemin, A_TABLE, nomtable,
nomtable

If Err Then

   Attacher = 0

Else

   Attacher = 1

End If

End Function


Function YEstElle (gl As Snapshot, nomtable As String) As Integer

   ' Cette fonction permet de verifier la présence d'un table de nom
   ' "NomTable" dans l'objet Snapshot GL, retourne 1 si elle y est
   ' 0 sinon
'*********************************************************************
gl.MoveFirst
Do Until gl.EOF

  If gl![name] = nomtable Then
      YEstElle = 1
      Exit Function
  End If

  gl.MoveNext
Loop
YEstElle = 0

End Function
```

```
Function VerifPresence (gl As Snapshot) As Integer

' Cette fonction permet de verifier dans l'objet Snapshot GL
' la présence de toutes les tables dont le nom est dans la table
' "TableDesNoms" .
' Retourne 1 si elles y sont toutes, 0 sinon

'*********************************************************************

On Error Resume Next

Dim MaTable As Table
Dim MaBD As Database

Set MaBD = CurrentDB()

Set MaBD = CurrentDB()
Set MaTable = MaBD.OpenTable("TableDesNoms")

Do Until MaTable.EOF

        nomtable = MaTable![nom]
        Dim LaTable As String
        LaTable = nomtable
        i = YEstElle(gl, LaTable)
        If i = 0 Then

           UneTableQuiManque = LaTable
           VerifPresence = 0
           Exit Function

        End If
        MaTable.MoveNext
Loop
VerifPresence = 1
End Function
```

```
Function SaisieChemin () As Integer

' Cette fonction permet de saisir dans la variable globale Chemin
' le nom et le chemin d'accès de la Base de Données où se trouvent
' les tables. Retourne 1 si le chemin est valide et 0 si l'utilisateur
' désire annuler.
' Chemin ne doit pas être vide, il ne doit pas être égal à
' "Chemin\NomBd.MDB" qui la 1 ere valeur par défaut la longueur du
' chemin doit être > à 4 à 4 lettres et doit se terminer par .MDB


'*****************************************************************************
On Error Resume Next

Msg = "Entrez le Chemin et le nom de la Base de Données où se trouvent les tables. "
Titre = "Saisie du chemin"

MsgErreur = "Le chemin n'est pas correct. Vous devez saisir le chemin et le nom (suivi de
.MDB) de votre Base de Données.    (EX: C:\ACCESS\COMPTOIR.MDB) "

Dim MaDB As Database
Dim tb As Table
Set MaDB = CurrentDB()
Set tb = MaDB.OpenTable("CHEMIN DE LA BASE ATTACHEE")
defvaleur = tb![CHEMIN]
tb.Close
While True

chemin = InputBox$(Msg, Titre, defvaleur)


Select Case chemin

   Case " "
   Reponse = MsgBox("Desirez vous sortir de l'application ? ", 4, "Sortie ?")
      If Reponse = 6 Then

         SaisieChemin = 0
         Exit Function

      End If

   Case "Chemin\NomBD.MDB"

      MsgBox MsgErreur

   Case Else
      defvaleur = chemin
      If Len(chemin) <= 4 Then

         MsgBox MsgErreur
```

```
    Else

      Var = Right(chemin, 4)

      If Var <> ".MDB" Then
        MsgBox "Le nom de votre Base de Données doit se terminer par .MDB"
      Else

        SaisieChemin = 1
        Exit Function
      End If

    End If


End Select

Wend

End Function

Sub Effacer (nomtable As String)

' Cette fonction permet d'effacer la table de nom "NomTable" de la
' Base de Données courante

'*********************************************************************

On Error Resume Next

Dim DBA As Database, TBA As Table

DoCmd SelectObject A_TABLE, nomtable, True

DoCmd SetWarnings False

DoCmd DoMenuItem A_FORMBAR, A_EDITMENU, A_DELETE

End Sub
```

```
Function DetacherTables ()

'Cette fonction permet de détacher toutes les tables attachées

'*********************************************************************

On Error Resume Next

Dim nomtable As String


Dim MaTable As Table
Dim MaBD As Database

Set MaBD = CurrentDB()
Set MaTable = MaBD.OpenTable("TableDesNoms")
MaTable.MoveNext

Do Until MaTable.EOF

   nomtable = MaTable![nom]
   Call Effacer(nomtable)
   MaTable.MoveNext
Loop

DoCmd SelectObject A_FORM, "menu general", False

End Function
```

```
Function AttacherTables ()

' Cette fonction permet d'attacher les tables à partir du nom et du
' chemin d'accès saisi par l'utilisateur.

'*********************************************************************
On Error Resume Next

Dim nomtable As String

   i = Attacher("associe")

   If i = 0 Then
      MsgBox " Une erreur s'est produite lors de l'attache des tables. Verifiez que le chemin
d'accés est correct et que votre Base contient toutes les tables communes."
      Exit Function
   Else
      Dim MaTable As Table
      Dim MaBD As Database

      Set MaBD = CurrentDB()
      Set MaTable = MaBD.OpenTable("TableDesNoms")
      MaTable.MoveNext

      Do Until MaTable.EOF

            nomtable = MaTable![nom]
            i = Attacher(nomtable)
            If i = 0 Then

               MsgBox "Impossible de trouver la table " & nomtable & ". Verifiez qu'elle se
trouve bien dans votre Base de Données"
               i = DetacherTables()
               Exit Function

            End If
            MaTable.MoveNext
         Loop

Dim tb As Dynaset
Set tb = MaBD.CreateDynaset("CHEMIN DE LA BASE ATTACHEE")
tb.Edit
tb.[CHEMIN] = chemin
tb.Update
tb.Close

         i = OuvertureFormM("menu general")

   End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCMouv, ce module contrôle les déplacements dans les formulaires.

'*************************************************************************

Function ActionAvancer ()

'Cette fonction rend actif l'enregistrements suivant si il n'est pas vide

'*************************************************************************
*

On Error Resume Next
Dim LeForm As Form

Set LeForm = Screen.ActiveForm
If Mode = 0 Then
  DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
  If Err Then
     MsgBox "Vous êtes sur le dernier enregistrement."
  End If
End If

End Function

Function ActionDernier ()

'Cette fonction rend actif le dernier enregistrement

'*************************************************************************
*

On Error Resume Next
Dim LeForm As Form

Set LeForm = Screen.ActiveForm
If Mode = 0 Then
  DoCmd GoToRecord A_FORM, LeForm.FormName, A_LAST
  If Err Then MsgBox " Impossible d'accéder au dernier enregistrement."
End If

End Function
```

```
Function ActionPremier ()

'Cette fonction rend actif le premier enregistrement.

'*************************************************************************
*

On Error Resume Next
Dim LeForm As Form

Set LeForm = Screen.ActiveForm
If Mode = 0 Then
  DoCmd GoToRecord A_FORM, LeForm.FormName, A_FIRST
  If Err Then MsgBox " Impossible d'accéder au premier enregistrement!"
End If

End Function


Function ActionReculer ()

'Cette fonction rend actif l'enregistrement précédent si non vide

'*****************************************************************

On Error Resume Next
Dim LeForm As Form

Set LeForm = Screen.ActiveForm
If Mode = 0 Then
  DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
  If Err Then MsgBox "Vous êtes sur le dernier enregistrement."
End If
End Function
```

```
Option Compare Database

' ModCAPropos, ce module contrôle le formulaire A propos
'*************************************************************************
*******


Type RECT
   left As Integer
   top As Integer
   right As Integer
   bottom As Integer
End Type

Type POINTAPI
   X As Integer
   Y As Integer
End Type

Declare Sub MoveWindow Lib "User" (ByVal hWnd As Integer, ByVal X As Integer, ByVal
Y As Integer, ByVal nWidth As Integer, ByVal nHeight As Integer, ByVal bRepaint As
Integer)
Declare Function GetParent Lib "User" (ByVal hWnd As Integer) As Integer
Declare Sub GetClientRect Lib "User" (ByVal hWnd As Integer, lpRect As RECT)
Declare Sub GetWindowRect Lib "User" (ByVal hWnd As Integer, lpRect As RECT)
Declare Sub ClientToScreen Lib "User" (ByVal hWnd As Integer, lpPoint As POINTAPI)


Dim pt As POINTAPI
Dim swp As RECT
Dim rParent As RECT
Dim iwidth As Integer
Dim iheight As Integer
Dim hWndMain As Integer
```

```
Function fctCentrer (hWnd As Integer)

' Cette fonction place la boîte de dialogue A propos au centre
' de la fenêtre Access

'*********************************************************************

' Déterminer le rectangle de la fenêtre parent
' et de la fenêtre enfant
 hWndMain = GetParent(hWnd)
 Call GetWindowRect(hWnd, swp)
 Call GetClientRect(hWndMain, rParent)

'Calculer la hauteur et la largeur pour MoveWindow
 iwidth = swp.right - swp.left
 iheight = swp.bottom - swp.top

'Trouver le centre à partir de la fenêtre parent
'et le convertir en coordonnées écran
 pt.X = (rParent.right - rParent.left) / 2
 pt.Y = (rParent.bottom - rParent.top) / 2
 Call ClientToScreen(hWndMain, pt)

'Calculer le nouveau point de départ (x,y)
 pt.X = pt.X - (iwidth / 2)
 pt.Y = pt.Y - (iheight / 2)

'Déplacer la fenêtre
 Call MoveWindow(hWnd, pt.X, pt.Y, iwidth, iheight, False)

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCAssocie contrôle le formulaire associe

'*****************************************************************************


Function CreerNvAssocie () As Integer

' Cette fonction crée un nouvel enregistrement avec les codes saisis
' Retourne 1 si tout c'est bien passé, 0 sinon

'*****************************************************************************
***************

On Error Resume Next

Dim MaBD As Database
Dim MD As Dynaset

Set MaBD = CurrentDB()
Set MD = MaBD.CreateDynaset("associe")
MD.AddNew
MD.[codePu] = ValPu
MD.[codeRe] = ValCodeR
MD.[codeCo] = ValCo
MD.[codeTh] = ValTh
MD.Update
MD.Close

If Err Then
  CreerNvAssocie = 0
Else
   CreerNvAssocie = 1
End If

End Function
```

```
Function NouvAssocie ()

' Cette ouvre le formulaire Nouveau associe, en mode 1, dans lequel vont être saisies les
' différentes clés par l'utilisateur

'*****************************************************************************
On Error Resume Next

Reponse = ConfirmationNv(" de associe")

If Reponse = 1 Then
  ValCodeR = 0
  ValPu = " "
  ValCo = " "
  ValTh = " "
  mode=1
  i = OuvertureFormM("Nouvel associe")

End If

End Function


Function OkAssocie ()

' Cette fonction permet de passer en mode consultation sur le nouvel enregistrement

'*****************************************************************************
***************

If mode = 0 Then
   i = OkSauve()

Else
  mode = 0
  i = OuvertureFormM("associe")

  If i <> 0 Then

    DoCmd GoToControl "codeRe"
    DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True

    DoCmd GoToControl "codeCo"
    If Screen.ActiveControl <> ValCo Then
      DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
    End If

    DoCmd GoToControl "codePu"
    If Screen.ActiveControl <> ValPu Then
      DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
```

```
    End If

    DoCmd GoToControl "codeTh"
    If Screen.ActiveControl <> ValTh Then
      DoCmd FindRecord ValTh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
    End If

  End If

End If

End Function


Function RetourCodeCo ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValCo

'*************************************************************************
On Error Resume Next

ValCo = Forms![Nouvel associe]!ListeContrats.Column(0)
DoCmd Requery "ListeContrats"

End Function


Function RetourCodePu ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValPu

'*************************************************************************
************


On Error Resume Next

ValPu = Forms![Nouvel associe]!ListePublications.Column(0)
DoCmd Requery "ListePublications"

End Function
```

```
Function RetourCodeR ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValCodeR

'*****************************************************************************
************

On Error Resume Next

ValCodeR = Forms![Nouvel associe]!ListeRealisations.Column(0)
DoCmd Requery "ListeRealisations"

End Function


Function RetourCodeTh ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValTh

'*****************************************************************************
************

On Error Resume Next

ValTh = Forms![Nouvel associe]!ListeThemes.Column(0)
DoCmd Requery "ListeThemes"

End Function


Function SupprimAssocie ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
**************

On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")
```

```
  If Reponse = 7 Then
    Exit Function
  Else
    ValCo = Forms!associe![codeCo]
    ValCodeR = Forms!associe![codeRe]
    ValTh = Forms!associe![codeTh]
    ValPu = Forms!associe![codePu]

    DoCmd RunSQL "DELETE FROM associe WHERE [codePu] =" & """" & ValPu & """"
& "AND [codeRe] =" & ValCodeR & "AND [codeTh] =" & """" & ValTh & """" & "AND
[codeCo] =" & """" & ValCo & """" & ";"

    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function


Function VerifAsso ()

' Cette fonction vérifie la validité des codes saisis
' retourne 1 si les clés sont correctes, 0 sinon

'*********************************************************************

On Error Resume Next

Dim DB As Database
Dim LaTable As Dynaset
Dim LaTab As Dynaset
Dim Texte1 As String
Dim Texte As String

Set DB = CurrentDB()

If IsNull(ValCodeR) Or ValCodeR = 0 Then
    Reponse = MsgBox("Vous devez sélectionner le code réalisation", 48,
"AVERTISSEMENT")
    Exit Function
End If

If IsNull(ValPu) Or ValPu = " " Then
  Reponse = MsgBox("Vous devez sélectionner le code publication", 48,
"AVERTISSEMENT")
  Exit Function
```

```
End If

If IsNull(ValCo) Or ValCo = " " Then
    Reponse = MsgBox("Vous devez sélectionner le code contrat", 48, "AVERTISSEMENT")
   Exit Function
End If

If IsNull(ValTh) Or ValTh = " " Then
     Reponse = MsgBox("Vous devez sélectionner le code thème", 48, "AVERTISSEMENT")
     Exit Function
End If

Set LaTab = DB.CreateDynaset("associe")
Texte = "[codeRe]=" & ValCodeR & "AND [codePu]=" & """" & ValPu & """" & " AND [codeTh]=" & """" & ValTh & """" & " AND [codeCo]=" & """" & ValCo & """"
Texte1 = "[codeRe]=" & ValCodeR

LaTab.FindFirst Texte1

If LaTab.NoMatch = True Then
  Call MessErreurCrea
  j = OuvertureFormM("associe")
  Exit Function
Else
        'Test de l'existence du nouvel enregistrement

  If (LaTab![codePu] = codePu) And (LaTab![codeTh] = codeTh) And (LaTab![codeCo] = codeCo) Then
     Msg = "Cet enregistrement existe déjà !"
     Reponse = MsgBox(Msg, 48, "AVERTISSEMENT")
     Exit Function

  Else
              'Création du nouvel enregistrement
     i = CreerNvAssocie()
     If i = 0 Then
       Call MessErreurCrea
       j = OuvertureFormM("associe")
     Else
       i = FermerFormActif()
       If i <> 0 Then
         mode = 1
         DoCmd OpenForm "associe", A_NORMAL, , Texte, A_EDIT, A_NORMAL
       End If
     End If
  End If
End If
LaTab.Close
DB.Close
End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCDiapo contrôle le formulaire diapositives


Function AnnuleDiapo ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table diapositives
' et ouvre le formulaire diapositive

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau diapositive]
ValDi = Forms![LeForm]![codeDi]

DoCmd RunSQL "DELETE FROM diapositives WHERE diapositives.codeDi =" & ValDi &
";"
mode = 0
i = OuvertureFormM("diapositives")

End Function



Function OkDiapo ()

' Cette fonction permet de revenir en mode consultation sur le dernier diapo créé

'*************************************************************************
On Error Resume Next

If mode = 0 Then
    i = OkSauve()

Else
   mode = 0
   i = OuvertureFormM("diapositives")
   If i <> 0 Then
      DoCmd GoToControl "codeDi"
      DoCmd FindRecord ValDi, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If

End If

End Function
```

```
Function CreerNvDiapo () As Integer

' Cette fonction crée un nouvel enregistrement avec un n° de Diapositive
' calculé (Le max + 1)
' Retourne le codeDi du nouvel enregistrement  s'il n'y a pas eu d'erreurs
' 0 sinon

'*********************************************************************

On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset

   Dim X As Integer

   X = CalculNvNum("DIAPOSITIVES", "CODEDI")
   ValDi = X
   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("DIAPOSITIVES")
   MD.AddNew
   MD.[codeDi] = ValDi
   MD.Update
   MD.Close

   If Err Then
      CreerNvDiapo = 0
   Else
      CreerNvDiapo = X
   End If

End Function
```

```
Function VerifNom () As Integer

' Cette fonction vérifie que le nom donné au nouveau diapo n'existe
' pas déjà
' retourne 1 s'il n'existe pas et 0 s'il existe
'*********************************************************************

   Dim Tamp As Control
   Dim Msg As String
   Dim Rep As Integer

   DoCmd GoToControl "titreDi"
   Set Tamp = Screen.ActiveControl

   If IsNull(Tamp) Or Tamp = " " Then

      Msg = " Vous devez donner un nom au diapositive que vous venez de créer"
      Rep = MsgBox(Msg, 48, "Saisie du nom du Nouveau Diapositive")
      PreActionOK = 0
   Else
      X = DLookup("[CODEDI]", "DIAPOSITIVES", "[TITREDI] = " & """" & Tamp &
""")

      If IsNull(X) Then
         VerifNom = 1
      Else
         Msg = "Le diapositive " & Tamp & " existe déjà "
         Rep = MsgBox(Msg, 48, "Saisie du nom du Nouveau Diapositive")
         VerifNom = 0
      End If

   End If

End Function
```

```
Function Ouvrepubli ()

' Cette fonction ouvre le formulaire publications sur l'enregistrement
' lié au diapositive actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "CODEDI"
   Tamp = Screen.ActiveControl
   ValDi = Tamp
   Dim Texte As String
   Texte = "[CODEDI] = " & ValDi
   X = DLookup("[codePu]", "contient", Texte) 'retourne le codePu lié au codeDi

   If IsNull(X) Then
     MsgBox "Aucune publication ne contient ce diapositive! Vous devez en créer un."
     i = OuvertureFormM("publications")
     i = NouvPub()

   Else

     i = OuvertureFormM("publications")
     DoCmd GoToControl "codePu"
     DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function
```

```
Function SupprimeDi ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!diapositives
    ValDi = Forms!diapositives![codeDi]

    DoCmd RunSQL "DELETE FROM diapositives WHERE diapositives.codeDi =" & ValDi
& ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

 'ModCGT contrôle le formulaire groupetravail


Function AnnuleGT ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table
' groupetravail et ouvre le formulaire groupetravail

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau groupetravail]
ValGT = Forms![LeForm]![codeGT]

DoCmd RunSQL "DELETE FROM groupetravail WHERE [codeGT] =" & """" & ValGT &
"""" & ";"
mode = 0
i = OuvertureFormM("groupetravail")

End Function


Function CreerNvGT () As Integer

  ' Cette fonction crée un nouvel enregistrement et retourne le nouveau codeGT
  'saisi par l'utilisateur s'il n'y a pas eu d'erreurs, 0 sinon

'*********************************************************************

On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset


   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("groupetravail")
   MD.AddNew
   MD.[codeGT] = ValGT   'saisi du code entré par l'utilisateur
   MD.Update
   MD.Close

   If Err Then
      CreerNvGT = 0
   Else
      CreerNvGT = 1
   End If
End Function
```

```
Function NouvGT ()

' Cette fonction crée un nouvel enregistrement dans la table groupetravail et ouvre
' le formulaire groupetravail afin de saisir les données relatives à ce nouveau groupe
' de travail.

'*****************************************************************************
On Error Resume Next

If mode = 0 Then

  Reponse = ConfirmationNv("de Groupe de travail")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = "Veuillez saisir le code groupe de travail.Il ne doit pas être vide!"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'un nouveau groupe de travail", DefValeur)
      If Reponse = " " Then Exit Function
      Dim UnGt As String
      UnGt = Reponse
      i = VerifcodeGt(UnGt)

      If i <> 0 Then
        i = CreerNvGT()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codeGT]=" & """" & ValGT & """"
          DoCmd OpenForm "Nouveau groupetravail", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
        Else
          Call MessErreurCrea
        End If
      Else
        DefValeur = UnGt
      End If
    Wend

  End If
End If

End Function
```

```
Function OkGT ()

' Cette fonction permet de revenir en mode consultation sur le dernier groupe de travail
' créé.

'*************************************************************************
On Error Resume Next

If mode = 0 Then
   i = OkSauve()

Else
  mode = 0
  i = OuvertureFormM("groupetravail")
  If i <> 0 Then
     DoCmd GoToControl "codeGT"
     DoCmd FindRecord ValGT, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
     End If
   End If

End Function
```

```
Function OuvreChrono ()

' Ouvre le formulaire chrono sur le chrono lié à ce groupe de travail.
' Si aucun chrono ne correspond au codeGT actif un message le signale et
' propose de le créer

'*************************************************************************
On Error Resume Next

If mode = 1 Then
  Call MessValider
Else
  DoCmd GoToControl "codeGT"
  Tamp = Screen.ActiveControl
  ValGT = Tamp

  Dim Texte As String
  Texte = "[CODEGT]=" & """" & ValGT & """"
  X = DLookup("[codeGT]", "chrono", Texte)

  If IsNull(X) Then
    MsgBox "Aucun chrono n'est encore enregistré pour ce groupe de travail!Vous devez en
créer un. "
    i = OuvertureFormM("chrono")
    i = NouvChrono()

  Else
    i = OuvertureFormM("chrono")
    DoCmd GoToControl "codeCh"
    DoCmd FindRecord ValCh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
  End If
End If

End Function
```

```
Function SupprimeGT ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************

On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!groupetravail
    ValGT = Forms!groupetravail![codeGT]

    DoCmd RunSQL "DELETE FROM groupetravail WHERE [codeGT] =" & """" & ValGT
& """" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function VerifcodeGt (UnGt As String)

'retourne 1 si le codeGt donné n'existe pas déjà, 0 sinon

'*****************************************************************************
**************

On Error Resume Next


If UnGt = "?" Then
   MsgBox "Impossible d'avoir comme Identificateur '?' "
   VerifcodeGt = 0
   Exit Function
End If

If UnGt Like "GT#" Or UnGt Like "GT##" Or UnGt Like "GT###" Or UnGt Like "GT####"
Then
   VerifcodeGt = 1
Else
   MsgBox "Le codeGT doit commencer par les lettres 'GT' suivies de 1 à 4 chiffres; exemple
:GT1 "
   VerifcodeGt = 0
   Exit Function
End If

If Len(UnGt) > 6 Or Len(UnGt) < 3 Then
   MsgBox "Le codeGT doit commencer par les lettres 'GT' suivies de 1 à 4 chiffres; exemple
:GT1 "
   VerifcodeGt = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codeGT] = " & """" & UnGt & """"

X = DLookup("[codeGT]", "groupetravail", Texte)

If IsNull(X) Then
   ValGT = UnGt
   VerifcodeGt = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme codeGT   " & UnGt
  VerifcodeGt = 0
End If


End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModChrono, ce module contrôle le formulaire chrono


Function AnnuleCh ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table chrono
' et ouvre le formulaire chrono

'*****************************************************************************
**************

On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau chrono]
ValCh = Forms![LeForm]![codeCh]

DoCmd RunSQL "DELETE FROM chrono WHERE [codeCh] =" & """" & ValCh & """" &
";"
mode = 0
i = OuvertureFormM("chrono")

End Function


Function CreerNvChrono () As Integer

' Cette fonction crée un enregistrement avec le code Chrono saisi par l'utilisateur.
' Retourne 1 si tout c'est bien passé, 0 sinon
'*****************************************************************************
On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset

   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("Chrono")
   MD.AddNew
   MD.[codeCh] = ValCh
   MD.Update
   MD.Close

   If Err Then

      CreerNvChrono = 0
   Else
      CreerNvChrono = 1
   End If
End Function
```

```
Function NouvChrono ()

' Cette fonction permet de créer un nouvel enregistrement pour le formulaire chrono.
' Après confirmation, un enregistrement est crée avec le codeCh entré par l'utilisateur
' dans la boite de dialogue.

'*****************************************************************************
************

On Error Resume Next

If mode = 0 Then
  Dim mess As String
  Reponse = ConfirmationNv("de chrono")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = " Veuillez saisir le code chrono. Il ne doit pas être vide"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'un nouveau code", DefValeur)
      If Reponse = " " Then Exit Function
        Dim Unchrono As String
        Unchrono = Reponse
        i = VerifChrono(Unchrono)

     If i <> 0 Then
        i = CreerNvChrono()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codeCh] = " & """" & ValCh & """"
          DoCmd OpenForm "Nouveau chrono", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
        Else
          Call MessErreurCrea
        End If

     Else
        DefValeur = Unchrono
    End If

  Wend
End If
End If
End Function
```

```
Function OKChrono ()

' Cette fonction permet de repasser en mode 0 (Consultation) sur le dernier enregistrement
' créé. Le contrôle "codeCh" est verrouillé.

'*************************************************************************

On Error Resume Next

If mode = 0 Then
    i = OkSauve()
 Else
   mode = 0
   i = OuvertureFormM("chrono")
   If i <> 0 Then

      DoCmd GoToControl "codeCh"
      DoCmd FindRecord ValCh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If

End If

End Function


Function OuvreCo ()

' Cette fonction permet d'ouvrir le formulaire contrat sur le même
' codeCh que celui du formulaire actif, s'il existe

'*********************************************************************
On Error Resume Next

If mode = 1 Then
   Call MessValider

Else
   DoCmd GoToControl "codeCh"
   Tamp = Screen.ActiveControl

   mode = 0
   i = OuvertureFormM("contrat")

   If i <> 0 Then
      DoCmd GoToControl "codeCh"
      DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If
End Function
```

```
Function OuvreGT ()

   ' Cette fonction permet d'ouvrir le formulaire groupetravail sur le même
   ' codeGT que celui du formulaire actif, s'il existe

'*****************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider
Else
   DoCmd GoToControl "codeGT"
   Tamp = Screen.ActiveControl

   mode = 0
   i = OuvertureFormM("groupetravail")

   If Not IsNull(i) Then

      DoCmd GoToControl "codeGT"
      DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If

End If

End Function


Function OuvrePub1 ()

' Cette fonction permet d'ouvrir le formulaire publication sur le même
' codeCh que celui du formulaire actif, s'il existe

'*********************************************************************
On Error Resume Next
If mode = 1 Then
   Call MessValider
Else
   DoCmd GoToControl "codeCh"
   Tamp = Screen.ActiveControl
   mode = 0
   i = OuvertureFormM("publications")
    If Not IsNull(i) Then
       DoCmd GoToControl "codeCh"
      DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If
End Function
```

```
Function SupprimeCh ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
**************

On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!chrono
    ValCh = Forms!chrono![codeCh]


    DoCmd RunSQL "DELETE FROM chrono WHERE chrono.codeCh =" & """" & ValCh
& """" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function VerifChrono (Unchrono As String)

' Cette fonction retourne 1 si le code chrono est correcte, 0 sinon

'*********************************************************************

On Error Resume Next


If Unchrono = "?" Then
   MsgBox "Impossible d'avoir comme Identificateur '?' "
   VerifChrono = 0
   Exit Function
End If

If Unchrono Like "CH##/##/##" Or Unchrono Like "CH##-##/##/##" Or Unchrono Like
"CH##/##/##-##/##/##" Then
   VerifChrono = 1
Else
   MsgBox "Le codeChrono doit commencer par les lettres 'CH' suivies d'une date de format
jj/mm/aa; exemples :CH01/10/92, CH02-03/10/12/93, CH25/10/92-12/11/92 "
   VerifChrono = 1
   Exit Function
End If

If Len(Unchrono) > 19 Or Len(Unchrono) < 10 Then
   MsgBox "Le codeChrono doit commencer par les lettres 'CH' suivies d'une date de format
jj/mm/aa; exemples :CH01/10/92, CH02-03/10/12/93, CH25/10/92-12/11/92 "
   VerifChrono = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codeCh] = " & """" & Unchrono & """"

X = DLookup("[codeCh]", "chrono", Texte)

If IsNull(X) Then
   ValCh = Unchrono
   VerifChrono = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme codeCh " & Unchrono
   VerifChrono = 0
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCIns contrôle le formulaire institutions


Function AnnulIns ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table
' institutions et ouvre le formulaire institutions

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouvelle institution]
ValIns = Forms![LeForm]![idInstitution]

DoCmd RunSQL "DELETE FROM institutions WHERE [idInstitution] =" & ValIns & ";"
mode = 0
i = OuvertureFormM("institutions")

End Function


Function OkInstitution ()

' Cette fonction permet de repasser en mode 0 (consultation) après la saisie d'une nouvelle
' institution

'*************************************************************************
***************

On Error Resume Next

If mode = 0 Then
   i = OkSauve()

Else
   mode = 0
   i = OuvertureFormM("institutions")
   If i <> 0 Then
      DoCmd GoToControl "idInstitution"
      DoCmd FindRecord ValIns, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If

End If

End Function
```

```
Function CreerNvInstitution ()

' Cette fonction permet de créer une nouvelle institution en lui attribuant
' un nouvel idInstitution.' Retourne le code de la nouvelle institution
' si tout s'est bien passé, 0 sinon.

'*********************************************************************

On Error Resume Next

Dim MaBD As Database
Dim MD As Dynaset

X = CalculNvNum("institutions", "idInstitution")
ValIns = X
Set MaBD = CurrentDB()
Set MD = MaBD.CreateDynaset("institutions")
MD.AddNew
MD.[idInstitution] = X
MD.Update
MD.Close

If Err Then
   CreerNvInstitution = 0
Else
   CreerNvInstitution = X
End If

End Function
```

```
Function NouvIns ()

' Cette fonction permet de saisir une nouvelle institution

'*********************************************************************

On Error Resume Next

If mode = 0 Then
  Dim UnEnt As Integer
  Dim UnNum As Integer

  UnEnt = ConfirmationNv(" d'institution")

  If UnEnt = 1 Then

    UnNum = CreerNvInstitution()
    If UnNum <> 0 Then
      ValIns = UnNum
      i = FermerFormActif()
      If i <> 0 Then
        mode = 1
        DoCmd OpenForm "Nouvelle institution", A_NORMAL, , "[idInstitution] = " &
ValIns, A_EDIT, A_NORMAL
      End If
    Else
      Call MessErreurCrea
    End If
  End If
End If

End Function


Function OuvrePays ()

' Cette fonction ouvre le formulaire pays sur l'enregistrement
' lié à l'institution active lors de l'appel

'*************************************************************************
*******

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codePays"
```

```
    Tamp = Screen.ActiveControl

    mode = 0


    If Not IsNull(Tamp) Then
      i = OuvertureFormM("pays")
      DoCmd GoToControl "codePays"
      DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
    Else
      i = OuvertureFormM("pays")
    End If

End If

End Function


Function OuvreRes ()

' Cette fonction ouvre le formulaire responsables sur l'enregistrement
' lié à l'institution active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

    Call MessValider

Else
    mode = 0
    i = OuvertureFormM("responsables")
    DoCmd GoToControl "idResponsable"
    DoCmd FindRecord ValRes, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

End If

End Function
```

```
Function OuvrirCo ()

' Cette fonction ouvre le formulaire contrat sur l'enregistrement
' lié à l'institution active lors de l'appel

'*****************************************************************************
*************

On Error Resume Next

If mode = 1 Then
  Call MessValider

Else

   DoCmd GoToControl "idInstitution"
   Tamp = Screen.ActiveControl
   ValIns = Tamp

   Dim Texte As String
   Texte = "[idInstitution] = " & ValIns
   X = DLookup("[codeCo]", "contrat", Texte) 'retourne le codeCo lié à l'idInstitution

   If Not IsNull(X) Then
     mode = 0
     i = OuvertureFormM("contrat")
     DoCmd GoToControl "codeCo"
     DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvrirRea ()

' Cette fonction ouvre le formulaire realisations sur l'enregistrement
' lié à l'institution active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "idInstitution"
   Tamp = Screen.ActiveControl
   ValIns = Tamp
   Dim Texte As String
   Texte = "[idInstitution] = " & ValIns
   X = DLookup("[CodeRe]", "organise", Texte) 'retourne le codeRe lié à l'idInstitution

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("realisations")
     DoCmd GoToControl "codeRe"
     DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True

   End If
End If

End Function
```

```
Function SupprimIns ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
**************

On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!institutions
    ValIns = Forms!institutions![idInstitution]

    DoCmd RunSQL "DELETE FROM institutions WHERE [idInstitution] =" & ValIns & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
   End If
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModContient contrôle le formulaire contient


Function CreerNvContient ()

' Cette fonction crée un nouvel enregistrement avec les codes saisis
' Retourne 1 si tout c'est bien passé, 0 sinon

'*********************************************************************

On Error Resume Next

Dim MaBD As Database
Dim MD As Dynaset

Set MaBD = CurrentDB()
Set MD = MaBD.CreateDynaset("contient")
MD.AddNew
MD.[codeDi] = ValDi
MD.[codePu] = ValPu
MD.Update
MD.Close

If Err Then
  CreerNvContient = 0
Else
   CreerNvContient = 1
End If

End Function
```

```
Function NouvContient ()

' Cette fonction ouvre le formulaire Nouveau contient dans lequel
' vont être saisies les différentes clés

'*********************************************************************

On Error Resume Next

If mode = 0 Then
  Reponse = ConfirmationNv(" de contient")

  If Reponse = 1 Then
    DoCmd GoToControl "codePu"
    Tamp = Screen.ActiveControl
    ValPu = Tamp

    DoCmd GoToControl "codeDi"
    Tamp = Screen.ActiveControl
    ValDi = Tamp
    mode=1
    i = OuvertureFormM("Nouveau contient")

  End If

End If

End Function


Function OkContient ()

' Cette fonction permet de passer en mode consultation sur le nouvel enregistrement

'*************************************************************************
***************

If mode = 0 Then
   i = OkSauve()

Else
  mode = 0
  i = OuvertureFormM("contient")

  If i <> 0 Then

    DoCmd GoToControl "codeDi"
    DoCmd FindRecord ValDi, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

    DoCmd GoToControl "codePu"
    If Screen.ActiveControl <> ValPu Then
```

```
      DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
    End If

  End If

End If

End Function


Function RetourValDi ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValDi

'*****************************************************************************
On Error Resume Next

 ValDi = Forms![nouveau contient]!ListeDiapos.Column(0)
 DoCmd Requery "ListeDiapos"

End Function
```

```
Function VerifContient ()

' Cette fonction ouvre le formulaire contient avec les codes saisis
' valides en mode 1

'*********************************************************************

On Error Resume Next

If IsNull(ValPu) Or ValPu = " " Then
    MsgBox ("Vous devez saisir le code publication")
    Exit Function
  End If

If IsNull(ValDi) Or ValDi = 0 Then
    MsgBox ("Vous devez saisir le code diapositive")
    Exit Function
  End If

Dim Texte1 As String
Texte1 = "[CodePu]=" & """" & ValPu & """" & " And [codeDi]=" & ValDi

X = DLookup("[codePu]", "contient", Texte1)

If IsNull(X) Then
  i = CreerNvContient()

  If i = 0 Then
    Call MessErreurCrea
    j = OuvertureFormM("contient")
  Else
    mode = 0
    i = FermerFormActif()
    DoCmd OpenForm "contient", A_NORMAL, , Texte1, A_EDIT, A_NORMAL
  End If
Else
  MsgBox "Cet enregistrement existe déjà !"

End If

End Function
```

```
Function SupprimeContient ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else

    ValDi = Forms!contient![codeDi]
    ValPu = Forms!contient![codePu]

    DoCmd RunSQL "DELETE FROM contient WHERE contient.codeDi =" & ValDi &
"And contient.codePu =" & """" & ValPu & """" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModContrat contrôle le formulaire contrat

Function AnnuleCo ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table contrat
' et ouvre le formulaire contrat

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau contrat]
ValCo = Forms![LeForm]![codeCo]

DoCmd RunSQL "DELETE FROM contrat WHERE [codeCo] =" & """" & ValCo & """" &
";"
mode = 0
i = OuvertureFormM("contrat")

End Function


Function CreerNvContrat () As Integer

' Cette fonction crée un enregistrement avec le code contrat saisi par l'utilisateur.
' Retourne 1 si tout c'est bien passé, 0 sinon
'*************************************************************************

On Error Resume Next

Dim MABD As Database
Dim MD As Dynaset
Set MABD = CurrentDB()
Set MD = MABD.CreateDynaset("contrat")
MD.AddNew
MD.[codeCo] = ValCo
MD.Update
MD.Close

If Err Then
  CreerNvContrat = 0
Else
  CreerNvContrat = 1
End If

End Function
```

```
Function Nouvcontrat ()

' Cette fonction permet de créer un nouvel enregistrement pour le formulaire contrat.
' Après confirmation, un enregistrement est crée avec le codeCo entré par l'utilisateur
' dans la boite de dialogue.

'*****************************************************************************
On Error Resume Next

If mode = 0 Then
  Dim mess As String
  Reponse = ConfirmationNv("de contrat")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = " Veuillez saisir le code contrat. Il ne doit pas être vide"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'un nouveau contrat", DefValeur)
      If Reponse = " " Then Exit Function
        Dim Uncont As String
        Uncont = Reponse
        i = VerifContrat(Uncont)

     If i <> 0 Then
        i = CreerNvContrat()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codeCo] = " & """" & ValCo & """"
          DoCmd OpenForm "Nouveau contrat", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
       Else
         Call MessErreurCrea
       End If

     Else
        DefValeur = Uncont
    End If

  Wend
End If
End If

End Function
```

```
Function OKContrat ()

' Cette fonction permet de repasser en mode consultation (mode 0) sur le nouvel
' enregistrement

'*****************************************************************

Dim ok As Integer
ok = 1
On Error Resume Next

If mode = 0 Then
   i = OkSauve()

Else
   If IsNull(Forms!contrat![titreCo]) Then
    MsgBox "Vous devez décrire le contrat que vous venez de créer"
    ok = 0
   End If
   If ok = 1 Then
    mode = 0
    i = OuvertureFormM("contrat")
     If i <> 0 Then
      DoCmd GoToControl "codeCo"
      DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
     End If
   End If
End If
ok = 1

End Function
```

```
Function OuvreChrono1 ()
' Cette fonction ouvre le formulaire chrono sur l'enregistrement
' lié au contrat actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeCh"
   Tamp = Screen.ActiveControl
   mode = 0
   i = OuvertureFormM("chrono")

   If Not IsNull(i) Then

      DoCmd GoToControl "codeCh"
      DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvreInst1 ()

' Cette fonction ouvre le formulaire institutions sur l'enregistrement
' lié au contrat actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeCo"
   Tamp = Screen.ActiveControl
   ValCo = Tamp

   Dim Texte As String
   Texte = "[codeCo] = " & ValCo
   x = DLookup("[idInstitution]", "contrat", Texte) 'retourne le idInstitution lié au codeCo

   If IsNull(x) Then
     MsgBox "Aucune institution n'est liée à ce contrat! Vous devez en créer une."
     i = OuvertureFormM("Nouveau institution")
     i = NouvIns()

   Else

     i = OuvertureFormM("institutions")
     DoCmd GoToControl "idInstitution"
     DoCmd FindRecord x, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function


Function OuvrePub ()

' Cette fonction ouvre le formulaire publications sur l'enregistrement
' lié au contrat actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider
```

```
Else

    DoCmd GoToControl "codeCo"
    Tamp = Screen.ActiveControl
    ValCo = Tamp


    DoCmd GoToControl "codeRe"
    Tamp = Screen.ActiveControl
    ValCodeR = Tamp

    Dim Texte As String
    Texte = "[codeCo] = " & """" & ValCo & """" & "And [codeRe]=" & ValCodeR
    x = DLookup("[codePu]", "associe", Texte) 'retourne le codePu lié au codeCo

    If IsNull(x) Then
      MsgBox "Aucune publication n'est liée à ce contrat! Vous devez en créer une."
      i = OuvertureFormM("publications")
      i = NouvPub()

    Else

      i = OuvertureFormM("publications")
      DoCmd GoToControl "codePu"
      DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
    End If
End If

End Function
```

```
Function OuvreTheme1 ()

' Cette fonction ouvre le formulaire themes sur l'enregistrement
' lié au contrat actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeCo"
   Tamp = Screen.ActiveControl
   ValCo = Tamp
   Dim Texte As String
   Texte = "[codeCo] = " & ValCo
   x = DLookup("[codeTh]", "associe", Texte) 'retourne le codeTh lié au codeCo

   mode = 0
   If IsNull(x) Then
     MsgBox "Il n'existe aucun thème lié a ce contrat!Vous devez en créer un."
     i = OuvertureFormM("themes")
     i = NouvTheme()
   Else

     i = OuvertureFormM("themes")
     DoCmd GoToControl "codeTh"
     DoCmd FindRecord ValTh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvrirReal ()

' Cette fonction ouvre le formulaire realisations sur l'enregistrement
' lié au contrat actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeRe"
   Tamp = Screen.ActiveControl

   i = OuvertureFormM("realisations")
   If Not IsNull(i) Then
     DoCmd GoToControl "CodeRe"
     DoCmd FindRecord Tamp, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function
```

```
Function SupprimeCo ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!contrat
    ValCo = Forms!contrat![codeCo]

    DoCmd RunSQL "DELETE FROM contrat WHERE contrat.codeCo =" & """" & ValCo
& """" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If
End If

End Function
```

```
Function VerifContrat (Uncont As String)

' Cette fonction retourne 1 si le code contrat est correcte, 0 sinon

'*****************************************************************************
**********

On Error Resume Next


If Uncont = "?" Then
   MsgBox "Impossible d'avoir comme codeCo '?' "
   VerifContrat = 0
   Exit Function
End If

If Uncont Like "CO-#" Or Uncont Like "CO-##" Or Uncont Like "CO-###" Then
   VerifContrat = 1
Else
    MsgBox "Le codeCo doit commencer par les lettres 'CO-'suivies de 1 à 3 chiffres; exemple
:CO-1 "
    VerifContrat = 0
    Exit Function
End If


If Len(Uncont) > 6 Or Len(Uncont) < 4 Then
   MsgBox "Le codeCo doit commencer par les lettres 'CO-'suivies de 1 à 3 chiffres; exemple
:CO-1 "
   VerifContrat = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codeCo] = " & """" & Uncont & """"

x = DLookup("[codeCo]", "contrat", Texte)

If IsNull(x) Then
   ValCo = Uncont
   VerifContrat = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme codeCo" & Uncont
   VerifContrat = 0
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

'ModCOrganise contrôle le formulaire organise


Function CreerNvOrganise ()

' Cette fonction crée un nouvel enregistrement avec les codes saisis Retourne 1
' si tout c'est bien passé, 0 sinon

'*************************************************************************
On Error Resume Next

Dim MaBD As Database
Dim MD As Dynaset

Set MaBD = CurrentDB()
Set MD = MaBD.CreateDynaset("organise")
MD.AddNew
MD.[idResponsable] = ValRes
MD.[codeRe] = ValCodeR
MD.[idInstitution] = ValIns
MD.Update
MD.Close

If Err Then
  CreerNvOrganise = 0
Else
   CreerNvOrganise = 1
End If

End Function
```

```
Function NouvOrganise ()

' Cette fonction ouvre le formulaire organise dans lequel vont être saisies les
' différentes clés par l'utilisateur

'*****************************************************************************
On Error Resume Next

If mode = 0 Then
  Reponse = ConfirmationNv(" de organise")

  If Reponse = 1 Then

    DoCmd GoToControl "idInstitution"
    Tamp = Screen.ActiveControl
    ValIns = Tamp

    DoCmd GoToControl "idResponsable"
    Tamp = Screen.ActiveControl
    ValRes = Tamp

    DoCmd GoToControl "codeRe"
    Tamp = Screen.ActiveControl
    ValCodeR = Tamp
    mode=1
    i = OuvertureFormM("Nouvel organise")

  End If

End If

End Function
```

```
Function OkOrganise ()

' Cette fonction permet de passer en mode consultation sur le nouvel enregistrement

'*****************************************************************************
If mode = 0 Then
   i = OkSauve()

Else
  mode = 0
  i = OuvertureFormM("organise")

  If i <> 0 Then

    DoCmd GoToControl "codeRe"
    DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True

    DoCmd GoToControl "idInstitution"
    If Screen.ActiveControl <> ValIns Then
      DoCmd FindRecord ValIns, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
    End If

    DoCmd GoToControl "idResponsable"
    If Screen.ActiveControl <> ValRes Then
      DoCmd FindRecord ValRes, A_ENTIRE, False, A_DOWN, False, A_CURRENT, False
    End If

  End If

End If

End Function
```

```
Function OkSaisiOrganise ()

' Cette fonction ouvre le formulaire organise avec les codes saisis
' valides en mode 1

'*********************************************************************

On Error Resume Next

If IsNull(ValCodeR) Or ValCodeR = 0 Then
     MsgBox ("Vous devez saisir le code réalisation")
     Exit Function
   End If
If IsNull(ValIns) Or ValIns = 0 Then
     MsgBox ("Vous devez saisir le code institution")
     Exit Function
   End If
If IsNull(ValRes) Or ValRes = 0 Then
     MsgBox ("Vous devez saisir le code responsable")
     Exit Function
   End If

Dim Texte As String
Texte = "[idResponsable]=" & ValRes & " And [idInstitution]=" & ValIns
X = DLookup("[idResponsable]", "responsables", Texte)
If IsNull(X) Then
  MsgBox " Il n'existe aucune responsable" & ValRes & "lié à l'institution" & ValIns
  Exit Function
End If

Dim Texte1 As String
Texte1 = "[CodeRe]=" & ValCodeR & " And [idInstitution]=" & ValIns & " And
[idResponsable]=" & ValRes

X = DLookup("[codeRe]", "organise", Texte1)
If IsNull(X) Then
  i = CreerNvOrganise()

  If i = 0 Then
    Call MessErreurCrea
    j = OuvertureFormM("organise")
  Else
    mode = 1
    i = FermerFormActif()
    DoCmd OpenForm "organise", A_NORMAL, , Texte1, A_EDIT, A_NORMAL
  End If
Else
  MsgBox "Cet enregistrement existe déjà !"
End If
End Function
```

```
Function RetourCodeRe ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValCodeR

'*************************************************************************
On Error Resume Next

ValCodeR = Forms![Nouvel organise]!ListeRealisations.Column(0)
DoCmd Requery "ListeRealisations"

End Function


Function RetourValIns ()

' Cette fonction recupère la valeur du contrôle actif dans la variable ValIns

'*************************************************************************
**************

On Error Resume Next

 ValIns = Forms![Nouvel organise]!ListeInstitutions.Column(0)
 DoCmd Requery "ListeInstitutions"

End Function


Function RetourValRes ()

' Cette fonction recupère la valeur du contrôle actif dans la variable
' ValRes

'*************************************************************************
***************

On Error Resume Next

ValRes = Forms![Nouvel organise]!ListeResponsables.Column(0)
DoCmd Requery "ListeResponsables"

End Function
```

```
Function SupprimOrganise ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
**************

On Error Resume Next

Dim LeForm As Form
If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Screen.ActiveForm
    ValCodeR = Forms!organise![codeRe]
    ValRes = Forms!organise![idResponsable]
    ValIns = Forms!organise![idInstitution]

    DoCmd RunSQL "DELETE FROM organise WHERE [codeRe]=" & ValCodeR & "
AND [idResponsable]=" & ValRes & " AND [idInstitution]=" & ValIns & ";"

    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function VerifOrganise ()

' Cette fonction ouvre le formulaire organise après la saisie des
' codes valides

'*********************************************************************

On Error Resume Next

If IsNull(ValCodeR) Or ValCodeR = 0 Then
    Reponse = MsgBox("Vous devez sélectionner le code réalisation", 48,
"AVERTISSEMENT")
    Exit Function
End If

If IsNull(ValIns) Or ValIns = 0 Then
  Reponse = MsgBox("Vous devez sélectionner le code institution", 48,
"AVERTISSEMENT")
  Exit Function
End If

If IsNull(ValRes) Or ValRes = 0 Then
    Reponse = MsgBox("Vous devez sélectionner le code responsable", 48,
"AVERTISSEMENT")
  Exit Function
End If

Dim Texte As String
Texte = "[codeRe]=" & ValCodeR & "AND [idResponsable]=" & ValRes & ";"

X = DLookup("[codeRe]", "responsables", Texte)

If IsNull(X) Then
  Msg = " Il n'existe aucune réalisation " & ValCodeR & "liée au contrat " & ValCo
  Reponse = MsgBox(Msg, 48, "AVERTISSEMENT")
  Exit Function
End If

Dim Texte1 As String
Texte1 = "[codeRe]=" & ValCodeR & " AND [idResponsable]=" & ValRes & " AND
[idInstitution]=" & ValIns

Y = DLookup("[codeRe]", "organise", Texte1)

If IsNull(Y) Then
  i = CreerNvOrganise()
  If i = 0 Then
    Call MessErreurCrea
    j = OuvertureFormM("organise")
  Else
    i = FermerFormActif()
```

```
    If i <> 0 Then
      mode = 0
      DoCmd OpenForm "organise", A_NORMAL, , Texte1, A_EDIT, A_NORMAL
    End If
  End If
Else
  Msg = "Cet enregistrement existe déjà !"
  Reponse = MsgBox(Msg, 48, "AVERTISSEMENT")
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCPays contrôle le formulaire pays

Function AnnulePa ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table pays
' et ouvre le formulaire pays

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau pays]
ValPa = Forms![LeForm]![codePays]

DoCmd RunSQL "DELETE FROM pays WHERE [codePays] =" & """" & ValPa & """" &
";"
mode = 0
i = OuvertureFormM("pays")

End Function


Function CreerNvPays () As Integer

' Cette fonction crée un enregistrement avec le codePays saisi par l'utilisateur.
' Retourne 1 si tout c'est bien passé, 0 sinon
'*************************************************************************

On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset

   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("pays")
   MD.AddNew
   MD.[codePays] = ValPa
   MD.Update
   MD.Close

   If Err Then

      CreerNvPays = 0
   Else
      CreerNvPays = 1
   End If
End Function
```

```
Function Nouvpays ()

' Cette fonction permet de créer un nouvel enregistrement pour
' le formulaire pays.
' Après confirmation, un enregistrement est crée avec le codePays entré par l'utilisateur
' dans la boite de dialogue.
'*****************************************************************************
On Error Resume Next

If mode = 0 Then
  Dim mess As String
  Reponse = ConfirmationNv("de pays")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = " Veuillez saisir le code pays. Il ne doit pas être vide"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'un nouveau pays", DefValeur)
      If Reponse = " " Then Exit Function
        Dim unpays As String
        unpays = Reponse
        i = verifPays(unpays)

     If i <> 0 Then
        i = CreerNvPays()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codePays] = " & """" & ValPa & """"
          DoCmd OpenForm "Nouveau pays", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
       Else
         Call MessErreurCrea
       End If

     Else
        DefValeur = unpays
    End If

  Wend
End If
End If

End Function
```

```
Function OKPays ()

' Cette fonction permet de repasser en mode 0 (Consultation) sur le dernier enregistrement
' créé. Le contrôle "codePays" est verrouillé.

'*********************************************************************

On Error Resume Next

If mode = 0 Then
    i = OkSauve()
Else
   mode = 0
   i = OuvertureFormM("pays")
   If i <> 0 Then

      DoCmd GoToControl "codePays"
      DoCmd FindRecord ValPa, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If

End If

End Function


Function OuvreInst ()

' Ouvre le formulaire institutions sur le même codePays que celui du
' formulaire actif, s'il existe
'*******************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider
Else
   mode = 1
   DoCmd GoToControl "codePays"
   Tamp = Screen.ActiveControl
   ValPa = Tamp
   mode = 0
   If Not IsNull(ValPa) Then
      i = OuvertureFormM("institutions")
      DoCmd GoToControl "idInstitution"
      DoCmd FindRecord ValIns, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If
End Function
```

```
Function SupprimePa ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!pays
    ValPa = Forms!pays![codePays]

    DoCmd RunSQL "DELETE FROM pays WHERE pays.codePays =" & """" & ValPa &
""" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function verifPays (unpays As String)

' Cette fonction retourne 1 si le code pays est correcte, ' 0 sinon

'*********************************************************************

On Error Resume Next


If unpays = "?" Then
   MsgBox "Impossible d'avoir comme codepays '?' "
   verifPays = 0
   Exit Function
End If

If unpays Like "???" Then
   verifPays = 0
Else
   MsgBox "Le codepays est composé de trois lettres, qui sont généralement les trois
premières lettres du nom du pays; exemple : fra pour France "
   verifPays = 0
   Exit Function
End If


If Len(unpays) > 3 Or Len(unpays) < 3 Then
   MsgBox "Le codepays est composé de trois lettres, qui sont généralement les trois première
lettres du nom du pays; exemple : fra pour France "
   verifPays = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codePays] = " & """" & unpays & """"

X = DLookup("[codePays]", "pays", Texte)

If IsNull(X) Then
   ValPa = unpays
   verifPays = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme codePays" & unpays
   verifPays = 0
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCPub contrôle le formulaire publications


Function AnnulePub ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table
' publication et ouvre le formulaire publications

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouvelle publication]
ValPu = Forms![LeForm]![codePu]

DoCmd RunSQL "DELETE FROM publication WHERE [codePu] =" & """" & ValPu & """"
& ";"
mode = 0
i = OuvertureFormM("publications")

End Function


Function CreerNvPub () As Integer

' Cette fonction crée un enregistrement avec le codePu  saisi par l'utilisateur.
' Retourne 1 si tout c'est bien passé, 0 sinon

'*************************************************************************

On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset

   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("publication")
   MD.AddNew
   MD.[codePu] = ValPu
   MD.Update
   MD.Close

   If Err Then

      CreerNvPub = 0
   Else
      CreerNvPub = 1
   End If
End Function
```

```
Function NouvPub ()

' Cette fonction permet de créer un nouvel enregistrement pour
' le formulaire publications.
' Après confirmation, un enregistrement est crée avec le codePu
' entré par l'utilisateur dans la boite de dialogue.
'*****************************************************************************
On Error Resume Next

If mode = 0 Then

  Reponse = ConfirmationNv("de publication")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = " Veuillez saisir le code publication. Il ne doit pas être vide"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'une nouvelle publication", DefValeur)
      If Reponse = " " Then Exit Function
        Dim Unpub As String
        Unpub = Reponse
        i = VerifPub(Unpub)

     If i <> 0 Then
        i = CreerNvPub()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codePu] = " & """" & ValPu & """"
          DoCmd OpenForm "Nouvelle publication", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
       Else
         Call MessErreurCrea
       End If

     Else
        DefValeur = Unpub
    End If

  Wend
End If
End If

End Function
```

```
Function OKPub ()

' Cette fonction permet de repasser en mode consultation (mode 0) sur le nouvel
' enregistrement

'*****************************************************************

Dim ok As Integer
ok = 1
On Error Resume Next

If mode = 0 Then
    i = OkSauve()

Else
   If IsNull(Forms!publications![titrePu]) Then
     MsgBox "Vous devez décrire la publication que vous venez de créer"
     ok = 0
   End If
   If ok = 1 Then
     mode = 0
     i = OuvertureFormM("publications")
      If i <> 0 Then
       DoCmd GoToControl "codePu"
       DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
      End If
    End If
End If
ok = 1

End Function
```

```
Function OuvreChrono2 ()

' Cette fonction ouvre le formulaire chrono sur l'enregistrement
' lié à la publication active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeCh"
   Tamp = Screen.ActiveControl
   ValCh = Tamp
   Dim Texte As String
   Texte = "[codeCh] = " & ValCh
   X = DLookup("[codeCh]", "chrono", Texte) 'retourne le codeCh lié au codePu

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("chrono")
     DoCmd GoToControl "codeCh"
     DoCmd FindRecord X, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvreContrat ()

' Cette fonction ouvre le formulaire contrat sur l'enregistrement
' lié à la publication active lors de l'appel

'*********************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codePu"
   Tamp = Screen.ActiveControl
   ValPu = Tamp
   Dim Texte As String
   Texte = "[codePu] = " & ValPu
   X = DLookup("[codeCo]", "associe", Texte) 'retourne le codeCo lié au codePu

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("contrat")
     DoCmd GoToControl "codeCo"
     DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvreDiapo ()

' Cette fonction ouvre le formulaire diapositives sur l'enregistrement
' lié à la publication active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codePu"
   Tamp = Screen.ActiveControl
   ValPu = Tamp
   Dim Texte As String
   Texte = "[codePu] = " & ValPu
   X = DLookup("[codeDi]", "contient", Texte) 'retourne le codeDi lié au codePu

   mode = 0


   If Not IsNull(X) Then
     i = OuvertureFormM("diapositives")
     DoCmd GoToControl "codeDi"
     DoCmd FindRecord ValDi, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvreRea ()

' Cette fonction ouvre le formulaire réalisations sur l'enregistrement
' lié à la publication active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codePu"
   Tamp = Screen.ActiveControl
   ValPu = Tamp
   Dim Texte As String
   Texte = "[codePu] = " & ValPu
   X = DLookup("[codeRe]", "associe", Texte) 'retourne le codeRe lié au codePu

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("realisations")
     DoCmd GoToControl "codeRe"
     DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True

   End If
End If

End Function
```

```
Function OuvreTheme2 ()

' Cette fonction ouvre le formulaire thèmes sur l'enregistrement
' lié à la publication active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codePu"
   Tamp = Screen.ActiveControl
   ValPu = Tamp
   Dim Texte As String
   Texte = "[codePu] = " & ValPu
   X = DLookup("[codeTh]", "associe", Texte) 'retourne le codeTh lié au codePu

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("themes")
     DoCmd GoToControl "codeTh"
     DoCmd FindRecord ValTh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function SupprimePub ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!publications
    ValPu = Forms!publications![codePu]

    DoCmd RunSQL "DELETE FROM publication WHERE publication.codePu=" & """" &
ValPu & """" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord A_FORM, LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function VerifPub (Unpub As String)

' Cette fonction retourne 1 si le code publication est correcte, 0 sinon

'*****************************************************************************
On Error Resume Next


If Unpub = "?" Then
   MsgBox "Impossible d'avoir comme codePu '?' "
   VerifPub = 0
   Exit Function
End If

If Unpub Like "PU#" Or Unpub Like "PU##" Or Unpub Like "PU###" Or Unpub Like
"PU####" Then
   VerifCode = 1
Else
   MsgBox "Le codePu doit commencer par les lettres 'PU' suivies de 1 à 4 chiffres; exemple
:PU225  "
   VerifPub = 0
   Exit Function
End If

If Len(Unpub) > 6 Or Len(Unpub) < 3 Then
   MsgBox "Le codePu doit commencer par les lettres 'PU' suivies de 1 à 4 chiffres; exemple
:PU225  "
   VerifPub = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codePu] = " & """" & Unpub & """"

X = DLookup("[codePu]", "publication", Texte)

If IsNull(X) Then
   ValPu = Unpub
   VerifPub = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme  " & """" & Unpub & """"
   VerifPub = 0
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCRea contrôle le formulaire realisations


Function AnnuleRea ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table
' realisations et ouvre le formulaire realisations

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouvelle realisation]
ValCodeR = Forms![LeForm]![codeRe]

DoCmd RunSQL "DELETE FROM realisations WHERE [codeRe] =" & ValCodeR & ";"
mode = 0
i = OuvertureFormM("realisations")

End Function


Function CreerNvRea ()

' Cette fonction crée un nouvel enregistrement avec un n° de CodeRe calculé (Le max + 1)
' Retourne le code Réalisation du nouvel enregistrement s'il n'y a pas eu d'erreurs,
' 0 sinon
'*********************************************************************

On Error Resume Next

   Dim MABD As Database
   Dim MD As Dynaset
   Dim X As Integer
   X = CalculNvNum("REALISATIONS", "codeRe")
   ValCodeR = X
   Set MABD = CurrentDB()
   Set MD = MABD.CreateDynaset("REALISATIONS")
   MD.AddNew
   MD.[codeRe] = X
   MD.Update
   MD.Close
   If Err Then
      CreerNvRea = 0
   Else
      CreerNvRea = X
   End If

End Function
```

```
Function NouvRea ()

' Cette fonction permet de créer un nouvel enregistrement pour
' la table REALISATIONS, et d'ouvrir le formuaire afin qu'il soit complété

'*****************************************************************

On Error Resume Next
If mode = 0 Then
Dim UnEnt As Integer
Dim UnNum As Integer

UnEnt = ConfirmationNv(" de réalisation")

If UnEnt = 1 Then

   UnNum = CreerNvRea()
   If UnNum = 0 Then
     Call MessErreurCrea

   Else
     ValCodeR = UnNum
     i = fermerformactif()
     If i <> 0 Then
       mode = 1
       Dim Texte As String
       Texte = "[codeRe] = " & ValCodeR
       DoCmd OpenForm "Nouvelle realisation", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
     End If

   End If

End If
End If

End Function
```

```
Function OKRea ()

' Cette fonction permet de repasser en mode consultation (mode 0) sur le nouvel
' enregistrement

'*****************************************************************

Dim ok As Integer
ok = 1
On Error Resume Next

If mode = 0 Then
   i = OkSauve()

Else
    mode = 0
    i = OuvertureFormM("Realisations")
     If i <> 0 Then
      DoCmd GoToControl "codeRe"
     DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True
     End If

End If
ok = 1
End Function
```

```
Function OuvreCo1 ()

' Cette fonction ouvre le formulaire contrat sur l'enregistrement
' lié à la réalisation active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else
   DoCmd GoToControl "codeRe"
   Tamp = Screen.ActiveControl
   ValCodeR = Tamp

   Dim Texte As String
   Texte = "[codeRe]=" & ValCodeR
   X = DLookup("[codeCo]", "contrat", Texte)

   If IsNull(X) Then
     MsgBox "Aucun contrat n'est encore enregistré pour cette réalisation!Vous devez en créer
un. "
     i = OuvertureFormM("contrat")
     i = Nouvcontrat()

   Else
     i = OuvertureFormM("contrat")
     DoCmd GoToControl "codeCo"
     DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvreIns ()

' Cette fonction ouvre le formulaire institutions sur l'enregistrement
' lié à la réalisation active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeRe"
   Tamp = Screen.ActiveControl
   ValCodeR = Tamp
   Dim Texte As String
   Texte = "[codeRe] = " & ValCodeR
   X = DLookup("[idInstitution]", "organise", Texte) 'retourne l'idInstitution lié au codeRe

   mode = 0

   If Not IsNull(X) Then
     i = OuvertureFormM("institutions")
     DoCmd GoToControl "idInstitution"
     DoCmd FindRecord ValIns, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function
```

```
Function OuvrePub2 ()

' Cette fonction ouvre le formulaire publications sur l'enregistrement
' lié à la réalisation active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeRe"
   Tamp = Screen.ActiveControl
   ValCodeR = Tamp
   Dim Texte As String
   Texte = "[codeRe] = " & ValCodeR
   X = DLookup("[codePu]", "associe", Texte) 'retourne le codePu lié au codeRe

   mode = 0
   i = OuvertureFormM("publications")

   If Not IsNull(X) Then

      DoCmd GoToControl "codePu"
      DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

   End If
End If

End Function


Function OuvreRes1 ()

' Cette fonction ouvre le formulaire responsables sur l'enregistrement
' lié à la réalisation active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else
```

```
    DoCmd GoToControl "codeRe"
    Tamp = Screen.ActiveControl
    ValCodeRe = Tamp
    Dim Texte As String
    Texte = "[codeRe] = " & ValCodeR
    X = DLookup("[idResponsable]", "organise", Texte) 'retourne l'idResponsable lié au codeRe

    mode = 0
    i = OuvertureFormM("responsables")

    If Not IsNull(X) Then

      DoCmd GoToControl "idResponsable"
      DoCmd FindRecord ValRes, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

    End If
End If

End Function

Function OuvreTheme ()

' Cette fonction ouvre le formulaire themes sur l'enregistrement
' lié à la réalisation active lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

    Call MessValider
Else
    DoCmd GoToControl "codeRe"
    Tamp = Screen.ActiveControl
    ValCodeR = Tamp
    Dim Texte As String
    Texte = "[codeRe] = " & ValCodeR
    X = DLookup("[codeTh]", "associe", Texte) 'retourne le codeTh lié au codeRe
    mode = 0
    i = OuvertureFormM("themes")
    If Not IsNull(X) Then
      DoCmd GoToControl "codeTh"
      DoCmd FindRecord ValTh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True

    End If
End If

End Function
```

```
Function SupprimeRea ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************

On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!realisations
    ValCodeR = Forms!realisations![codeRe]

    DoCmd RunSQL "DELETE FROM realisations WHERE realisations.codeRe =" &
ValCodeR & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCRes contrôle le formulaire responsable


Function AnnuleRes ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table diapositives
' et ouvre le formulaire diapositive

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau responsable]
ValRes = Forms![LeForm]![idResponsable]

DoCmd RunSQL "DELETE FROM responsables WHERE [idResponsable] =" & ValRes &
";"
mode = 0
i = OuvertureFormM("responsables")

End Function


Function CreerNvResponsable ()
' Cette fonction permet de créer un nouveau responsable en attribuant
' à ce dernier automatiquement un nouveau code
' Retourne le nouveau IdResponsable si tout s'est bien passé, 0 sinon
'*********************************************************************

On Error Resume Next
Dim MABD As Database
Dim MD As Dynaset

X = CalculNvNum("RESPONSABLES", "IDRESPONSABLE")
ValRes = X
Set MABD = CurrentDB()
Set MD = MABD.CreateDynaset("RESPONSABLES")

MD.AddNew
MD.[idResponsable] = X
MD.Update
MD.Close

If Err Then
   CreerNvResponsable = 0
Else
   CreerNvResponsable = X
End If
End Function
```

```
Function NouvResponsable ()

' Cette fonction crée après confirmation un nouvel enregistrement et ouvre le formulaire
' en mode consultation et modification(mode 0)

'*************************************************************************
On Error Resume Next
If mode = 0 Then
i = ConfirmationNv(" de responsable")

If i = 1 Then

   i = CreerNvResponsable()
   If i <> 0 Then

      mode = 1
      ValRes = i
      i = FermerFormActif()
      DoCmd OpenForm "Nouveau responsable", A_NORMAL, , "[iDResponsable] = " &
ValRes, A_EDIT, A_NORMAL

   Else
      Call MessErreurCrea
      i = OuvertureFormM("Responsables")
   End If

End If
End If

End Function


Function OkResponsable ()

' Cette fonction permet de repasser en mode 0 après saisi du nouveau responsable
'*************************************************************************
Dim ok As Integer
ok = 1
If mode = 0 Then
   i = OkSauve()

Else
   mode = 0
   i = OuvertureFormM("responsables")
   If i <> 0 Then
     DoCmd GoToControl "idResponsable"
     DoCmd FindRecord ValRes, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function
```

```
Function OuvRea ()

' Cette fonction ouvre le formulaire realisations sur l'enregistrement
' lié au responsable actif lors de l'appel

'*********************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "idResponsable"
   Tamp = Screen.ActiveControl
   ValRes = Tamp
   Dim Texte As String
   Texte = "[idResponsable] = " & ValRes
   X = DLookup("[codeRe]", "organise", Texte) 'retourne le codeRe lié à l'idResponsable

   If IsNull(X) Then
     MsgBox "Aucune réalisation n'est liée à ce responsable! Vous devez en créer une."
     i = OuvertureFormM("realisations")
     i = NouvRea()

   Else

     i = OuvertureFormM("realisations")
     DoCmd GoToControl "codeRe"
     DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True
   End If
End If

End Function
```

```
Function OuvrIns ()

' Cette fonction ouvre le formulaire realisations sur l'enregistrement
' lié au responsable actif lors de l'appel

'*********************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "idResponsable"
   Tamp = Screen.ActiveControl
   ValRes = Tamp
   Dim Texte As String
   Texte = "[idResponsable] = " & ValRes
   X = DLookup("[idInstitution]", "responsables", Texte) 'retourne l'idInstitution au codeCo

   If IsNull(X) Then
     MsgBox "Aucune institution n'est liée à ce responsable! Vous devez en créer une."
     i = OuvertureFormM("institutions")
     i = NouvPub()

   Else

     i = OuvertureFormM("institutions")
     DoCmd GoToControl "idInstitution"
     DoCmd FindRecord X, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If


End Function
```

```
Function SupprimeRes ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? Ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!responsables
    ValRes = Forms!responsables![idResponsable]

    DoCmd RunSQL "DELETE FROM responsables WHERE [idResponsable] =" & ValRes
& ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCTheme contrôle le formulaire theme


Function AnnuleTh ()

' Cette fonction annule la création d'un nouvel enregistrement dans la table theme
' et ouvre le formulaire themes

'*************************************************************************
On Error Resume Next

Dim LeForm As Form
Set LeForm = Forms![Nouveau theme]
ValTh = Forms![LeForm]![codeTh]

DoCmd RunSQL "DELETE FROM theme WHERE [codeTh] =" & """" & ValTh & """" &
";"
mode = 0
i = OuvertureFormM("themes")

End Function


Function CreerNvTheme () As Integer

' Cette fonction crée un enregistrement avec le code theme saisi par l'utilisateur.
' Retourne 1 si tout c'est bien passé, 0 sinon

'*************************************************************************

On Error Resume Next

   Dim MaBD As Database
   Dim MD As Dynaset

   Set MaBD = CurrentDB()
   Set MD = MaBD.CreateDynaset("theme")
   MD.AddNew
   MD.[codeTh] = ValTh
   MD.Update
   MD.Close

   If Err Then

      CreerNvTheme = 0
   Else
      CreerNvTheme = 1
   End If
End Function
```

```
Function NouvTheme ()

' Cette fonction permet de créer un nouvel enregistrement pour la table contrat.
' Après confirmation, un enregistrement est crée avec le codeCo  entré par l'utilisateur
' dans la boite de dialogue.

'*****************************************************************************
On Error Resume Next

If mode = 0 Then
  Dim mess As String
  Reponse = ConfirmationNv("de Theme")
  If Reponse = 1 Then

    Dim Msg As String
    Msg = " Veuillez saisir le code thème. Il ne doit pas être vide"
    DefValeur = "?"

    While True

      Reponse = InputBox(Msg, "Saisie d'un nouveau thème", DefValeur)
      If Reponse = " " Then Exit Function
        Dim Untheme As String
        Untheme = Reponse
        i = VerifTheme(Untheme)

     If i <> 0 Then
        i = CreerNvTheme()
        If i <> 0 Then

          i = FermerFormActif()
          mode = 1
          Dim Texte As String
          Texte = "[codeTh] = " & """" & ValTh & """"
          DoCmd OpenForm "Nouveau theme", A_NORMAL, , Texte, A_EDIT,
A_NORMAL
          Exit Function
        Else
          Call MessErreurCrea
        End If

     Else
        DefValeur = Untheme
    End If

  Wend
End If
End If
End Function
```

```
Function OKTheme ()

' Cette fonction permet de repasser en mode consultation (mode 0) sur le nouvel
' enregistrement

'*****************************************************************************
Dim ok As Integer
ok = 1
On Error Resume Next

If mode = 0 Then
    i = OkSauve()

Else
   If IsNull(Forms!themes![titreTh]) Then
     MsgBox "Vous devez décrire le thème que vous venez de créer"
     ok = 0
   End If
   If ok = 1 Then
     mode = 0
     i = OuvertureFormM("themes")
      If i <> 0 Then
       DoCmd GoToControl "codeTh"
       DoCmd FindRecord ValTh, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
      End If
    End If
End If
ok = 1

End Function
```

```
Function OuvreCo2 ()

' Cette fonction ouvre le formulaire contrat sur l'enregistrement
' lié au theme actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeTh"
   Tamp = Screen.ActiveControl
   ValTh = Tamp
   Dim Texte As String
   Texte = "[codeCo] = " & ValTh
   X = DLookup("[codeCo]", "associe", Texte) 'retourne le codeCo lié au codeTh

   mode = 0
   i = OuvertureFormM("contrat")
   If Not IsNull(X) Then

     DoCmd GoToControl "codeCo"
     DoCmd FindRecord ValCo, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function
```

```
Function OuvrePub3 ()

' Cette fonction ouvre le formulaire publications sur l'enregistrement
' lié au thème actif lors de l'appel

'*********************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeTh"
   Tamp = Screen.ActiveControl
   ValCo = Tamp
   Dim Texte As String
   Texte = "[codeTh] = " & ValTh
   X = DLookup("[codePu]", "associe", Texte) 'retourne le codePu lié au codeTh
   i = OuvertureFormM("publications")

   mode = 0
   If Not IsNull(X) Then
     DoCmd GoToControl "codePu"
     DoCmd FindRecord ValPu, A_ENTIRE, False, A_DOWN, False, A_CURRENT, True
   End If
End If

End Function
```

```
Function OuvreRea1 ()

' Cette fonction ouvre le formulaire réalisation sur l'enregistrement
' lié au theme actif lors de l'appel

'*************************************************************************

On Error Resume Next

If mode = 1 Then

   Call MessValider

Else

   DoCmd GoToControl "codeTh"
   Tamp = Screen.ActiveControl
   ValTh = Tamp
   Dim Texte As String
   Texte = "[codeCo] = " & ValTh
   X = DLookup("[codeRe]", "associe", Texte) 'retourne le codeRe lié au codeTh

   If IsNull(X) Then
     MsgBox "Aucune réalisation n'est liée à ce contrat! Vous devez en créer une."
     i = OuvertureFormM("realisations")
     i = NouvRea()

   Else
     ValRe = X
     i = OuvertureFormM("realisations")
     DoCmd GoToControl "codeRe"
     DoCmd FindRecord ValCodeR, A_ENTIRE, False, A_DOWN, False, A_CURRENT,
True
   End If
End If


End Function
```

```
Function SupprimeTh ()

' Cette fonction supprime l'enregistrement actif après confirmation et ouvre le formulaire
' actif sur l'enregistrement précédent.Si l'enregistrement supprimé est le premier alors,
' l'enregistrement suivant devient l'enregistrement actif

'*****************************************************************************
On Error Resume Next

Dim LeForm As Form

If mode = 1 Then
  Call MessValider
Else
  Dim Msg As String
  Msg = "Etes-vous sûr de vouloir supprimer l'enregistrement actif ? ces données seront
perdues à jamais!"
  Reponse = MsgBox(Msg, 36, "ATTENTION")

  If Reponse = 7 Then
    Exit Function
  Else
    Set LeForm = Forms!themes
    ValTh = Forms!themes![codeTh]

    DoCmd RunSQL "DELETE FROM theme WHERE theme.codeTh =" & """" & ValTh &
"""" & ";"
    DoCmd GoToRecord A_FORM, LeForm.FormName, A_PREVIOUS
    If Err Then
        MsgBox "Vous venez de supprimer le premier enregistrement!"
        DoCmd GoToRecord LeForm.FormName, A_NEXT
    End If
  End If

End If

End Function
```

```
Function VerifTheme (Untheme As String)

' Cette fonction retourne 1 si le code theme est correcte, 0 sinon

'*********************************************************************

On Error Resume Next


If Untheme = "?" Then
   MsgBox "Impossible d'avoir comme codeTh '?' "
   VerifTheme = 0
   Exit Function
End If

If Untheme Like "TH#" Or Untheme Like "TH##" Or Untheme Like "TH##" Or Untheme
Like "TH####" Or Untheme Like "TH#####" Or Untheme Like "TH######" Then
   VerifTheme = 1
Else
   MsgBox "Le codeTh doit commencer par les lettres 'TH' suivies de 1 à 6 chiffres; exemples
:TH222, TH1 "
   VerifTheme = 0
   Exit Function
End If

If Len(Untheme) > 8 Or Len(Untheme) < 3 Then
   MsgBox "Le codeTh doit commencer par les lettres 'TH' suivies de 1 à 6 chiffres; exemples
:TH222, TH1 "
   VerifTheme = 0
   Exit Function
End If

Dim Texte As String
Texte = "[codeTh] = " & """" & Untheme & """"

X = DLookup("[codeTh]", "theme", Texte)

If IsNull(X) Then
   ValTh = Untheme
   VerifTheme = 1

Else

   MsgBox "Il existe déjà un enregistrement ayant comme codeTh" & Uncont
   VerifTheme = 0
End If

End Function
```

```
Option Compare Database   'Utilise l'ordre de la base pour les comparaisons de chaînes

' ModCStat, ce module contrôle le formulaire stattables


Function stat ()

' Cette fonction compte le nombre d'enregistrement dans chaque table et affecte le résultat
' dans le champ correspondant du formulaire stattables

'*****************************************************************************
On Error Resume Next

Dim valeur As Long
Dim MaDB As Database
Dim MTB As Dynaset

Set MaDB = CurrentDB()

i = OuvertureFormM("stattables")

  Set MTB = MaDB.CreateDynaset("publication")
  MTB.MoveLast
  forms!stattables!NBPUB = MTB.RecordCount
  MTB.Close

  Set MTB = MaDB.CreateDynaset("diapositives")
  MTB.MoveLast
  forms!stattables!NBDIA = MTB.RecordCount
  MTB.Close

  Set MTB = MaDB.CreateDynaset("chrono")
  MTB.MoveLast
  forms!stattables!NBCHR = MTB.RecordCount
  MTB.Close

  Set MTB = MaDB.CreateDynaset("associe")
  MTB.MoveLast
  forms!stattables!NBASS = MTB.RecordCount
  MTB.Close

  Set MTB = MaDB.CreateDynaset("pays")
  MTB.MoveLast
  forms!stattables!NBPAS = MTB.RecordCount
  MTB.Close

  Set MTB = MaDB.CreateDynaset("realisations")
  MTB.MoveLast
  forms!stattables!NBREA = MTB.RecordCount
  MTB.Close
```

```
    Set MTB = MaDB.CreateDynaset("responsables")
    MTB.MoveLast
    forms!stattables!NBRES = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("organise")
    MTB.MoveLast
    forms!stattables!NBORG = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("institutions")
    MTB.MoveLast
    forms!stattables!NBINS = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("theme")
    MTB.MoveLast
    forms!stattables!NBTHM = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("contient")
    MTB.MoveLast
    forms!stattables!NBCNT = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("contrat")
    MTB.MoveLast
    forms!stattables!NBCTR = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("groupeTravail")
    MTB.MoveLast
    forms!stattables!NBGTR = MTB.RecordCount
    MTB.Close

    Set MTB = MaDB.CreateDynaset("CHEMIN DE LA BASE ATTACHEE")
    forms!stattables!NOMBASE = MTB.Chemin
    Open MTB.Chemin For Binary As #1
    lataille = LOF(1)
    Close #1
    lataille = lataille \ 1000
    forms!stattables!taille = lataille

End Function
```