TD/TP Bases de Données N°3 : VBA (suite) : Les données

Exercice N°1 : Modification de table

1) Il s'agit d'accéder à une table de la base données courante depuis une procédure afin de la modifier (la table LOC).

Pour cela :

- Dans un nouveau module *module2*, créer une nouvelle procédure de type **Sub** nommée *Modifierbase*
- Déclarer les variables *MaBase*, *MaTableDef*, *MonChamp* avec les types adéquats.
- Affecter à la variable *MaBase* la base courante (**CurrentDb**) et à la variable *MaTableDef* la table **LOC** de la base.
- Changer le nom de la table LOC en LOCATAIRES (Propriété Name d'un objet TableDef).
- Modifier le nom du champ *Numéro* en *Num_loc* (Propriété **Name** d'un contrôle)
- Ajouter le champ *Pnom_loc* de type Texte et de taille 25 à cette table. (Méthodes **CreateField** et **Append**)
- Afficher le nombre de champs de la table LOCATAIRES (Fonction **MsgBox** et Propriété **Count** de la collection **Fields**).

Après exécution de la procédure, vérifier le résultat sous ACCESS (onglet Tables).

2) Vérifier que le formulaire LOCATAIRES, créé précédemment, fonctionne toujours malgré les modifications faites sur la table LOCATAIRES.

Créer une nouvelle procédure de type sub, nommée *TransfomeFormulaire()* chargée de faire quelques modifications sur le formulaire LOCATAIRES, c'est-à-dire qu'il devra :

- Ouvrir le formulaire LOCATAIRES en mode création (Méthode **OpenForm** de l'objet **DoCmd**),
- Modifier la légende du formulaire : Mettre « LOCATAIRES » au lieu de « LOC » (propriété **Caption** d'un formulaire),
- Modifier la légende et la largeur d'un contrôle de type étiquette : changer « Nom_Loc » en « Locataire » et augmenter sa largeur de 1 cm (propriété **Caption** et **Width** d'un contrôle).
- Créer une Zone de Texte pour pouvoir afficher (ou saisir) le prénom du locataire (**CreateControl**) : Même alignement, même hauteur mais à 2 cm à gauche du contrôle Nom_Loc et de largeur 3 cm.

E LOCATAIRES								
•	Numéro Locataire adr loc	CARDONNE	CLIQUEZ	1				

- Fermer le formulaire LOCATAIRES en enregistrant les modifications (Méthode **Close** de l'objet **DoCmd**).

Après exécution de la procédure, vérifier le résultat sous ACCESS.

Modifier la procédure *ModifierFormulaire()* créée précédemment (TP02) pour obtenir le résultat suivant lorsqu'on clique sur le bouton :

8	LOCATAIRES			
1	NUMÉRO	1		
	LOCATAIRE	CARDONNE		
	ADR	15 bis, avenue des Champs Elysées		
	CP	75000		
	VILLE	PARIS	Fermer	
			L	
Enr	ra 🔣 🔍	1 • • • • • sur 23		

Utiliser le formulaire pour saisir plusieurs prénoms.

Exercice N°2 : Création de tables et de relations

Il s'agit maintenant de créer les tables **SEMAINES**, **TARIFS** et **LOCATIONS** manquantes dans la base de données fournie.



Pour cela :

- Dans le module2, ajouter une nouvelle procédure de type **Sub** nommée *AjoutTables*.
- Déclarer les variables nécessaires *MaBase, MaTableDef, MonChamp, MonIndex* avec les types adéquats.
- Créer 3 tables **SEMAINES**, **TARIFS** et **LOCATIONS** (utiliser **CreateTableDef** et **Append**) et leurs champs.

Table **SEMAINES** :

	NumSemaine	Entier	
	DateLundi	Date	
Table TARIFS :			
	NumApp	Entier Long	
	NumSemaine	Entier	
	Prix	Monétaire	
Table LOCATIONS :			
	NumLocation	Entier Long	NuméroAuto
	NumApp	Entier Long	
	Numsemaine	Entier	
	NumLoc	Entier Long	

- Ex. :Set Mabase = CurrentDbSet MaTableDef = Mabase.CreateTableDef("SEMAINES").... ' Creation des champs et des index de la tableMabase.TableDefs.AppendMaTableDef
- Créer les index de clés primaires de chaque table : NumSemaine (Table SEMAINES), NumLocation (Table LOCATIONS), NumMeu et NumSemaine (Table TARIFS).

Pour créer les index de clés primaires, utiliser les méthodes **CreateIndex**, **CreateField**, **Append** et la propriété **Primary**.

- Ajouter également un index NomPrénomLoc <u>unique</u> composé de Nom_Loc et Pnom_loc sur la table LOCATAIRE (utiliser les méthodes CreateIndex, CreateField, Append et la propriété Unique)
- Créer les relations entre les tables (utiliser la méthode **CreateRelation**).
- Enfin, affichez le message "Base de données Modifiée".

Une fois la procédure *AjoutTables* créée, compiler le module et corriger les erreurs éventuelles. Exécuter ensuite la procédure *AjoutTables*.

Fermer le module en l'enregistrant.

Vérifier les modifications apportées à la base de base de données :

- Tables TARIFS, LOCATIONS et SEMAINES créées
- Vérifier les caractéristiques des champs des tables
- Vérifier les caractéristiques des index de ces 2 tables (cliquez dans la barre d'outils)
- Vérifier les caractéristiques de la relation entre ces 2 tables

Exercice N°3 : Utiliser la bibliothèque ACCESS pour importer des données

Les tables **SEMAINES**, **TARIFS** et **LOCATIONS** ont été créées et ajoutées à la base existante OFTOUR.

Dans le module2, écrire une nouvelle procédure *Importer_données* qui sera chargée d'importer dans ces 3 tables, des données à partir des fichiers Texte disponibles dans le répertoire */home/bib/sandron/S3* : LOCATIONS.TXT, TARIFS.TXT et SEMAINES.TXT.

Pour cela, utiliser la méthode **TransferText** de l'objet **DoCmd**.

On désire également charger la table LOCATAIRES avec de nouvelles données : Inclure à la procédure le code qui supprime les données de la table LOCATAIRES puis qui importe les données à partir du fichier LOCATAIRES.TXT.

Pour cela, utiliser les méthodes RunSql puis TransferText de l'objet DoCmd

Attention : Afin d'éviter les erreurs d'importation, ne pas oublier

- de vérifier le type des champs que vous avez créés
- de respecter l'intégrité référentielle

Vérifier que toutes les tables ont été correctement chargées.