



Cours d'informatique

Réaliser une base de données avec Access

Patrice Buche
Département OMIP
INA Paris Grignon



1. Introduction

1.1 Introduction générale

Cet enseignement d'informatique a un double objectif :

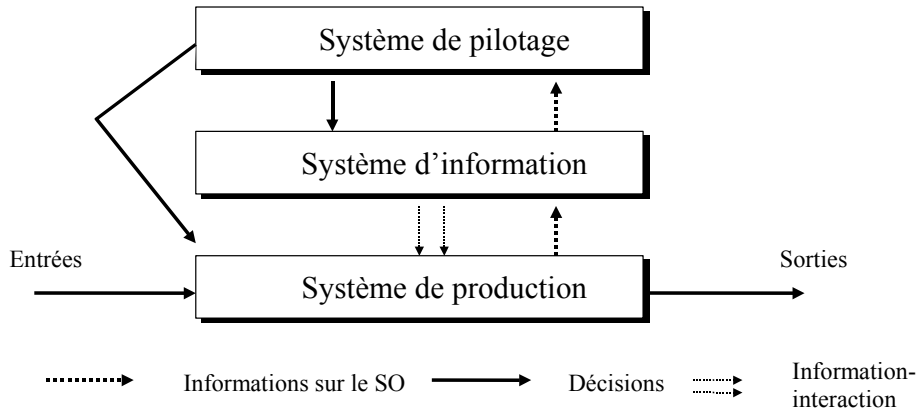
1) Il complète les enseignements de début d'année par la présentation d'un nouvel outil de bureautique largement diffusé dans les entreprises : Access.

- Il fait partie de la suite de produits bureautiques vendus par l'éditeur de logiciels Microsoft, au même titre que Word et Excel.
- Word permet de faire du traitement de texte et Excel des calculs dans des tableaux de données.
- Access permet de réaliser des bases de données, partie automatisée des systèmes d'information des entreprises, sous la forme d'ensembles de tableaux de données reliés entre eux par des liaisons sémantiques .

2) Il introduit de manière concrète, précise et intuitive, par des exemples et des définitions, la notion de base de données et les concepts informatiques essentiels permettant leur réalisation. Son utilité pour un ingénieur agronome est illustrée par l'étude d'exemples de bases de données issus des domaines de la zootechnie et de l'agronomie.

La méthode pédagogique choisie est une alternance entre des présentations de concepts nécessaires à la bonne compréhension du fonctionnement du logiciel et des manipulations dirigées sur ordinateurs.

Systeme d'information, une definition



1.2 Système d'information

Un **système de production** est un ensemble d'éléments matériels ou immatériels (êtres vivants, machines, méthodes, règles, ...) en interaction transformant par un processus des éléments (les entrées) en d'autres éléments (les sorties).


Exemple : Dans un haras, le poulain est le résultat de la fécondation d'une jument par un étalon.

Un système de production est, en général, contrôlé par un **système de pilotage**. Le système de pilotage reçoit des informations sur l'état du **système de production** (statistiques de vente, par exemple), mesure l'écart aux objectifs et réagit éventuellement par des décisions influençant le processus du système opérant (lancement d'un nouveau produit, modification du prix de vente, ...).

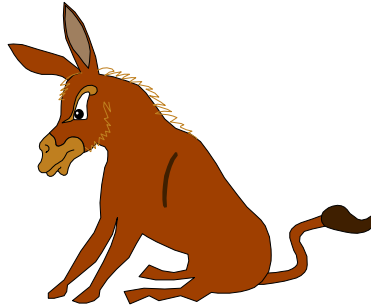
Pour obtenir ces informations, les organisations se dotent d'un **système d'information** qui est l'interface entre le système de pilotage et le système de production.


Système d'information, une définition :

- Il rassemble les données nécessaires au fonctionnement d'une activité de production de l'entreprise
- Il fournit les outils pour mettre à jour et exploiter ces données



Systeme d'information, un premier exemple
la gestion d'un haras



- ◆ Gestion des données liées à la reproduction
 - ◆ Un volume important de données (800 juments traitées par saison)
 - ◆ Entreprise de petite taille (5 postes de travail)
- 

1.3 L'exemple de la gestion d'un haras

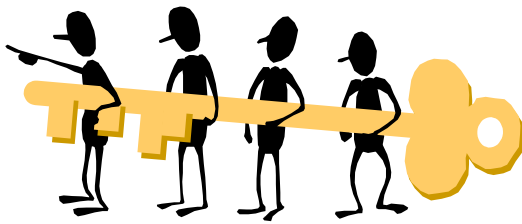
Le Haras des Cruchettes est une entreprise de services de petite taille, spécialisée dans la gestion de carrières de monte d'étalons. Son activité consiste principalement à récolter les étalons et à suivre les juments confiées au Haras pour leur mise à la reproduction.

Le Haras gère une dizaine d'étalons et plus de 800 juments pendant chaque saison (de fin février à début août). Le volume important de données manipulées pour cette activité a justifié la mise en place d'un système d'information automatisé (SAI).

Nous utiliserons des exemples issus de ce cas réel tout au long de la formation pour illustrer les concepts informatiques qui seront présentés.

Vous allez commencer par manipuler une application complètement réalisée sous Access permettant de gérer ce haras.

Un système d'information, pour quoi faire ?



- ◆ Toute l'équipe utilise et enrichit en même temps cet ensemble d'informations
- ◆ Avoir très rapidement et en permanence sur son poste de travail l'information la plus récente

1.4 Utilité d'un système d'information automatisé

Partage et mise à jour de l'information par toute une équipe, voire plusieurs services d'une même organisation. Ceci permet à chaque usager du système d'accéder à l'information la plus récente.

Exemple: Les services des grandes mairies de France partagent le même système d'information pour gérer les réseaux de canalisation d'eau, de câblage électrique et le plan d'occupation du sol. Dès que quelque chose change dans l'un de ces réseaux, les autres services sont immédiatement au courant.

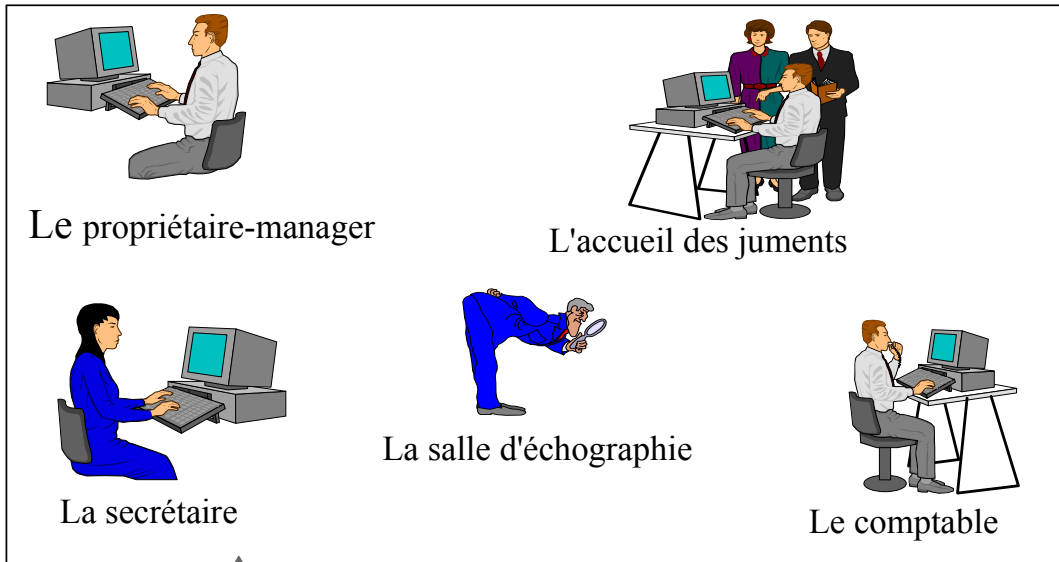
L'aide à la décision : si la décision revient à l'homme, la machine est capable de mobiliser et de synthétiser rapidement une grande masse d'informations utile à la prise de décision.

Exemple: Afin de réaliser une promotion en tête de gondole, le responsable du rayon d'un supermarché recherche dans les statistiques de vente les produits qui sont souvent achetés ensemble.

La simplification et l'amélioration du travail par l'automatisation des tâches répétitives et fastidieuses.

Exemple: la généralisation de l'utilisation des lecteurs de codes barres a considérablement allégé les tâches fastidieuses de saisie.

Un système d'information commun enrichi par toute une équipe : la gestion d'un haras



1.5 Un système d'information enrichi par toute une équipe

Le propriétaire-manager du Haras est en même temps le commercial de la PME. Il vend les saillies de ses **étalons**. Il négocie les **contrats** de saillies avec les **clients**. Clients et contrats sont enregistrés dans la base de données.

Le comptable reçoit les contrats signés accompagnés d'un chèque pour la réservation d'une saillie (le haras ne peut utiliser que 100 saillies d'un même étalon par saison). Les réservations confirmées sont enregistrées dans la base de données.

Lorsque la **jument** arrive au haras, on l'enregistre dans la base de données.

Les juments sont suivies par échographie. Le résultat de chaque examen est enregistré dans sa **fiche de suivi** informatisée. Après l'examen, on enregistre la date de prochaine visite, voire d'insémination.


Les clients désirent souvent avoir des informations sur leur jument lorsqu'elle est en pension au haras. La secrétaire leur répond en consultant la fiche de suivi de la jument dans la base de données.

Le logiciel utilisé pour réaliser le SIA doit permettre les accès concurrents aux informations. Plus précisément, il doit gérer si nécessaire les conflits d'accès simultanés en écriture sur la même information.



Le haras des Cruchettes

Présentation d'une maquette réalisée
avec Access de Microsoft



1.6 Prise en main de l'application Gestion de Haras

Manipulations :

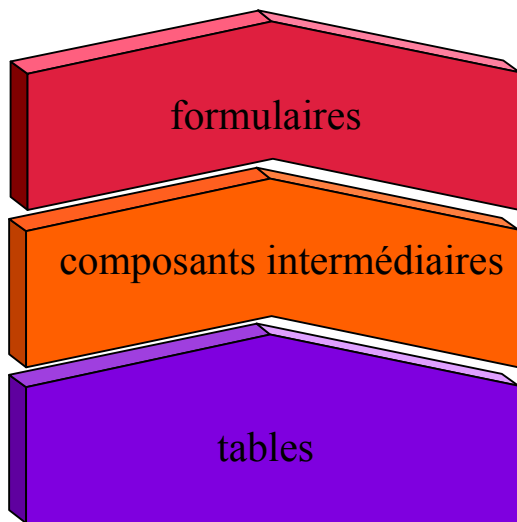
- *Manipuler les listes déroulantes des trois formulaires d'accès aux fiches client, étalon et jument.*
- *Accéder à la fiche d'un client existant*
- *Créer la fiche d'un nouveau client*
- *Accéder à la liste des contrats d'un client (par exemple : Bernadou)*
- *Créer un nouveau contrat*
- *Ouvrir la fiche d'une jument déjà enregistrée (par exemple : Bagatelle de Mai)*
- *Créer la fiche d'une nouvelle jument (saisir la date du jour comme date à revoir)*
- *Editer la liste des juments à voir dans la journée*
- *Accéder à la fiche de suivi d'une jument (par exemple : Bagatelle de Mai)*
- *Enregistrer le résultat d'un examen échographique et une insémination dans la fiche de suivi d'une jument*
- *Accéder à la fiche d'un étalon (par exemple Hurgo)*
- *Sélectionner les juments affectées à l'étalon depuis sa fiche*
- *Ouvrir la fiche de suivi d'une jument sélectionnée dans la liste des juments affectées à l'étalon (double-clic sur le nom de la jument)*
- *Depuis le menu Etalon, calculer le bilan de fertilité des étalons (global et par chaleur)*

Tester la réaction d'Access lorsque deux postes mettent à jour en même temps la même donnée. Pour cela, deux postes de travail voisins exécutent la même application.

Regarder la fenêtre *Base de données*.

Les données de la base sont physiquement rangées dans les tables (onglet Tables).

Programmer une base de données avec Access, c'est :



- ◆ paramétrer des composants logiciels préexistants :
 - les tables, supports de stockage des données
 - des composants intermédiaires
 - les formulaires présentés à l'utilisateur
- ◆ emboîter ces composants

1.7 Programmer une base de données Access

Les étapes à respecter lorsque l'on crée une base de données avec Access sont :

- Définir les informations que l'on stocke dans les tables de la base de données (en paramétrant les composants Table à partir de la fenêtre Base de Données) et les liaisons existant entre ces tables.
- Définir la façon dont on va présenter les informations stockées dans les tables à l'utilisateur. Pour cela, on crée des formulaires (en paramétrant les composants Formulaire à partir de la fenêtre Base de Données). On aura besoin également de créer d'autres composants que l'on associera aux formulaires (requêtes, macro-commandes que l'on définira plus tard).
- Le principe de construction d'une application est comparable aux poupées russes. On emboîte les composants les uns dans les autres (les tables dans des requêtes, les requêtes et les macros dans des formulaires).

Pour illustrer ces propos, nous allons ébaucher la construction de la fiche Personne :

1. on va d'abord créer la **table** *Personne* dans laquelle Access stocke les données sur les personnes enregistrées dans la base,
2. on va ensuite fabriquer le **formulaire** *FichePersonne* qui va permettre de visualiser sous forme de fiche les données stockées dans la table Personne.

Microsoft Access - [Personne : Table]

Fichier Edition Affichage Insertion Format Enregistrements Outils Fenêtre ?

| | Id_Personne | Nom_Pers | Prénom | Adresse | Ville | CodePostal | Téléphone | |
|---|--------------------|-----------------|---------------|-------------------------------|------------------|-------------------|------------------|-------|
| | 1 | Bernadou | Lucien | 10, place de la Mairie | BOURNAIZE | 65480 | 02-55-43-67-34 | |
| | 2 | Gallois | Patrick | 27, route de Ponthieu | CRECY EN VILAINE | 35426 | 03-39-43-67-34 | |
| | 8 | Dibango | Manu | Chemin du Cap Vert | PARIS | 75020 | 01-44-49-78-56 | 01444 |
| | 9 | Letourneur | André | 16, rue des Roses | L'Haye les roses | 94700 | 01-44-48-12-23 | 01444 |
| | 10 | Huard | Mireille | 16, rue Claude Bernard | PARIS CÉDEX 5 | 75231 | 01-44-08-18-05 | |
| | 11 | Crésus | Onassis | 15, avenue des Champs Elysées | PARIS | 75016 | 01-02-03-01-02 | |
| | 12 | Dupont | Gilles | 11, rue de la Mousson | PARIS | 75000 | 01-05-04-78-45 | |
| * | (NuméroAuto) | | | | | | | |

Clé primaire : numéro différent pour chaque personne

Propriétés

La table Personne

Enr: 1 sur 7

Mode Feuille de données

Démarrer Eudora - [In] Explorateur - Bd1A Microsoft Word - ... Microsoft PowerP... Microsoft Acc... 11:46

1.8 Création d'une table

1.8.1 Définitions

Définition 1 : Une **table** permet de modéliser un **objet** que l'on désire représenter dans la base de données. Les objets de l'univers modélisé sont encore appelés les **entités** du système d'information.

Exemple: Dans l'application Haras, la table Personne permet de stocker toutes les informations nécessaires à la gestion des clients.

Un objet, modélisé par une table, peut être caractérisé par un ensemble de propriétés élémentaires.

Exemples: On désire caractériser l'objet client par son adresse, son numéro de téléphone,

Définition 2 : Chaque **propriété** de l'entité modélisée est représentée dans une **colonne** de la table modélisant l'entité.

Définition 3 : On associe à chaque colonne, un **type élémentaire** prédéfini. Il détermine le format de la donnée enregistrée dans la colonne.

Parmi les types prédéfinis, les types texte et numérique sont les plus utilisés.

Exemple : Le nom est de type texte de longueur 30 caractères, l'âge est de type numérique, réel.

Définition 4 : Dans chaque table, une colonne particulière, appelée la **clé primaire** de la table, permet d'identifier de manière unique chaque occurrence de l'entité représentée par un enregistrement dans la table.

Cela est particulièrement nécessaire pour lever les ambiguïtés potentielles engendrées par les homonymies.

Exemple: Gérard Dupont dont l'IdPersonne est 14 est l'oncle de Gérard Dupont dont l'IdPersonne est 56.

Remarque 1 : Par commodité, on utilise en général le type **NuméroAuto** pour la clé primaire de la table. Il correspond à un entier numérique de type entier long. Access génère automatiquement dans cette colonne un entier (incrémenté à chaque fois de 1) lorsque l'on crée un nouvel enregistrement.

Exemple: IdPersonne de type NuméroAuto est clé primaire de la table Personne.

Remarque 2 : Si une propriété de l'entité peut naturellement jouer ce rôle, on la choisit comme clé primaire de la table.

Exemple: Si dans la table Personne, on désire mémoriser le numéro de sécurité sociale pour des besoins de gestion, alors on choisit cette propriété comme clé primaire de la table.

1.8.2 Manipulations : Création d'une version simplifiée de la table Personne

Lancez Microsoft Access. Vous allez créer une nouvelle base de données. Choisissez l'option **Créer une nouvelle base de données** et cliquez sur ok. Dans la boîte de dialogue **Fichier nouvelle base de données** :

- *sélectionnez votre répertoire de travail (ie N:\),*
- *créez une nouvelle base portant le nom haras.mdb.*

Dans cette nouvelle base, vous allez créer la table *Personne*. Cliquez sur l'onglet **Tables** de la fenêtre **Base de données**. Cliquez sur le bouton **Nouveau**, puis sélectionnez dans la liste l'option **Mode Création**.

Entrez les noms des 3 colonnes de la table *Personne* dans la colonne **Nom du Champ** et sélectionnez leur type dans la colonne **Type de données**. Les 3 colonnes sont :

- *IdPersonne* (la clé de type **NuméroAuto**),
- *NomPersonne* (de type **Texte de 30 caractères**),
- *PrénomPersonne* (de type **Texte de 30 caractères**).

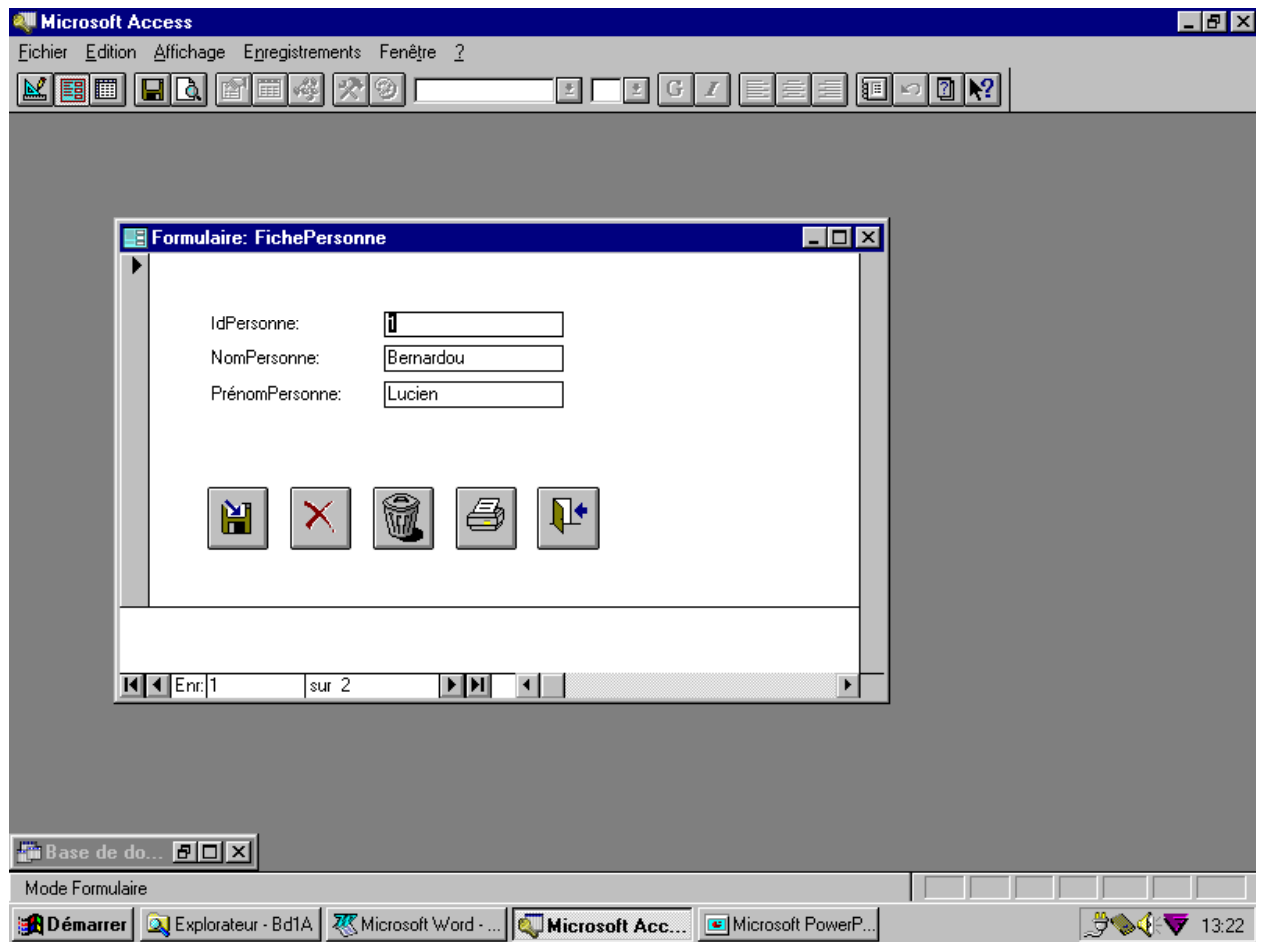
Définissez la colonne *IdPersonne* en tant que clé primaire de la table. Pour cela, sélectionnez la ligne contenant la définition de *IdPersonne* et cliquez sur le bouton **Clé primaire** de la barre d'outils.

Terminez la saisie en enregistrant la table. Sélectionnez dans le menu **Fichier**, l'option **Enregistrer Sous**. Dans le champ de saisie **Nouveau nom**, tapez *Personne*. Cliquez sur OK. Revenez à la fenêtre **Base de données** en sélectionnant l'option **Fermer** du menu **Fichier**. Vous devez voir apparaître la table *Personne* sous l'onglet **Tables**.

Identifiez les deux modes d'ouverture d'une table (mode **Création** et mode **feuille de données**) en cliquant sur les boutons **Modifier** et **Ouvrir**.

Vous allez maintenant saisir quelques occurrences de personnes dans la table. Sélectionnez l'onglet **Table** dans la fenêtre **Base de données**, double-cliquez avec le bouton gauche sur le nom de la table *Personne*.

Saisissez dans la table *Personne*, les personnes *Bernadou Lucien* et *Gallois Patrick*. Terminez la saisie de la table en sélectionnant l'option **Fermer** du menu **Fichier**.



1.9 Création d'un formulaire

1.9.1 Un formulaire, pour quoi faire ?

Pour pouvoir présenter les informations de manière plus agréable et plus pratique que lorsque l'on utilise la table : présentation en mode fiche.

Pour que l'utilisateur puisse accéder aux données qui l'intéressent sans avoir besoin de connaître la structure des tables de la base.

Pour pouvoir réaliser des contrôles sur la façon dont sont saisies les données ou pour protéger les données visualisées lorsque l'on ne fait que de la consultation.

1.9.2 Un formulaire, définitions

Définition 1 : Un formulaire est une fenêtre Windows dans laquelle on affiche une source de données (table ou requête). Il est composé de **contrôles graphiques**.

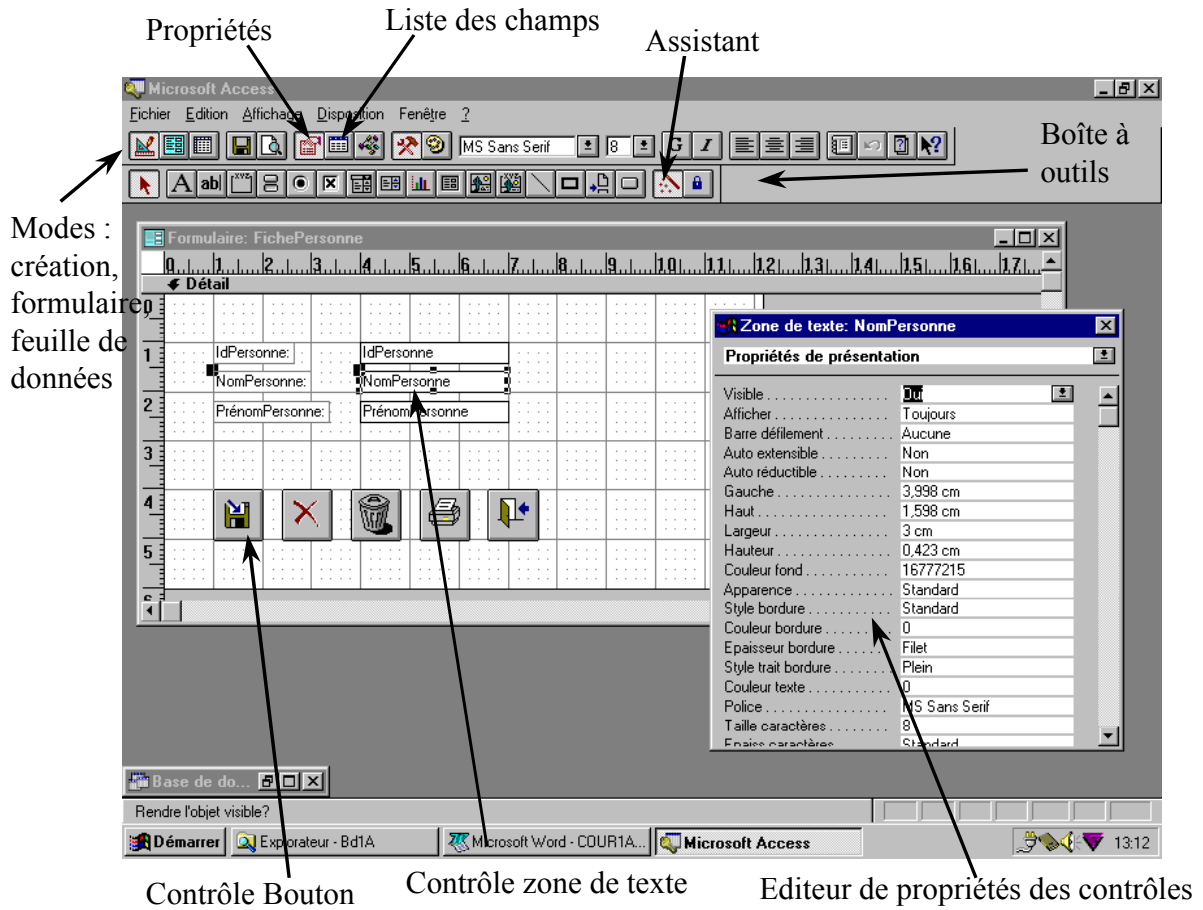
Définition 2 : Un contrôle graphique est un composant logiciel paramétrable destiné à être affiché sur le formulaire.

Exemple : dans le formulaire, chaque colonne de la table Personne peut être affichée dans un contrôle graphique de type zone de texte.

Remarque 1 : On parle de contrôle graphique car on peut contrôler le comportement du composant logiciel en le paramétrant.

Par exemple, on peut contrôler le mode d'affichage de la donnée (affichage en gras, italique, ...).

Remarque 2 : Le formulaire, lui-même, est un contrôle. Ce contrôle permet de gérer le comportement de la fenêtre Windows (affichage en mode agrandissement, ...)



1.9.3 Manipulations : création du formulaire Personne avec l'aide d'un assistant

Dans ce formulaire, nous allons présenter les données contenues dans la table *Personne*

Cliquez sur l'onglet *Formulaires* de la fenêtre *Base de données*.

Cliquez sur le bouton *Nouveau*.

Sélectionnez la table *Personne* dans la liste déroulante.

Sélectionnez dans la liste des assistants, l'assistant *Formulaire instantané : Colonnes*

Nous allons partir de cette première mise en forme pour obtenir le formulaire présenté sur la page de gauche.

1.9.4 Manipulations : Personnalisation du formulaire FichePersonne

Identifiez les trois modes d'utilisation d'un formulaire :

- mode formulaire : les données d'une même ligne de la table sont présentées dans le formulaire,
- mode création : ce mode permet de modifier la présentation des données sur le formulaire,
- mode feuille de données : les données sont affichées en mode tableau.

On peut passer d'un mode à un autre à partir du bouton *Affichage* dans la barre d'outils. Testez ces trois modes.

En mode création, élargissez votre formulaire sur la droite. Déplacez les contrôles zones de texte comme sur l'exemple. Sauvegardez (icône Disquette) en nommant le formulaire *FichePersonne*. Exécutez-le (icône Affichage, mode Formulaire dans la barre d'outil).

Nous allons ajouter 5 boutons de commande sur le formulaire. Affichez la boîte à outils (cliquez sur l'icône boîte à outils). Activez les assistants contrôle (bouton *baguette magique* dans la boîte à outils).

Cliquez sur le bouton *bouton de commandes* dans la boîte à outils. Placez-le sur le formulaire. Sous l'assistant, sélectionnez dans la liste *Catégories*, l'option *Opérations sur enregistrements*, puis dans la liste *Actions* l'option *Sauvegarder un enregistrement*. Cliquez sur le bouton *Suivant*. Choisissez une icône qui apparaîtra sur le bouton. Puis cliquez sur le bouton *Terminer*. Recommencez l'opération pour placer un bouton *Annuler* et un bouton *Supprimer*. Sauvegardez et exécutez le formulaire pour vérifier le fonctionnement des trois boutons.

En suivant la même procédure, ajoutez un bouton pour imprimer le formulaire (catégorie *Opérations sur formulaire*, action *imprimer le formulaire*) et pour fermer le formulaire (catégorie *Opérations sur formulaire*, action *Fermer un formulaire*). Sauvegardez et exécutez.

1.9.5 Paramétrage des contrôles du formulaire

Les contrôles sont des composants prédéfinis paramétrables. On programme un contrôle en modifiant la valeur de ses propriétés avec **l'éditeur de propriétés**.

Les propriétés sont nombreuses. Elles sont regroupées en 4 ensembles :

Propriétés des données : Elles permettent d'indiquer quelles données provenant des tables et des requêtes de la base de données sont attachées au contrôle.

*Exemples : Le contrôle Formulaire FichePersonne a pour propriété **Source** le nom de la table attachée au formulaire, à savoir Personne.*

*Le contrôle Zone de texte NomPersonne a pour propriété **Source contrôle** la colonne NomPersonne incluse dans la table Personne.*

Propriétés de format : Elles permettent de spécifier comment le contrôle est affiché à l'écran.

*Exemples : La propriété **Police** du contrôle Zone de texte NomPersonne est MS Sans Serif. La propriété **Taille caractères** est à 8.*

Propriétés des événements : Elles permettent de définir le comportement de l'application lorsque survient un événement qui affecte le contrôle.

Exemple : Lorsque l'on clique sur le bouton de commande Fermer un formulaire, Access ferme le formulaire Fiche Personne.

Autres propriétés : Elles regroupent des propriétés diverses.

*Exemple : La propriété **Index tabulation** permet de déterminer l'ordre dans lequel sont parcourues les zones de saisie.*

1.9.6 Récapitulation sur la création d'un premier formulaire

Nous avons tout d'abord créé le composant table *Personne*.

Puis, nous avons saisi des données dans la table.

A quoi sert la clé primaire dans la table *Personne* ?

Le formulaire a été créé avec un assistant. Un **assistant** aide l'utilisateur d'Access à paramétrer les composants. Il facilite la prise en main de l'outil. Mais, progressivement, nous allons prendre l'habitude de configurer nous-mêmes les composants car les assistants ne permettent pas de tout faire.

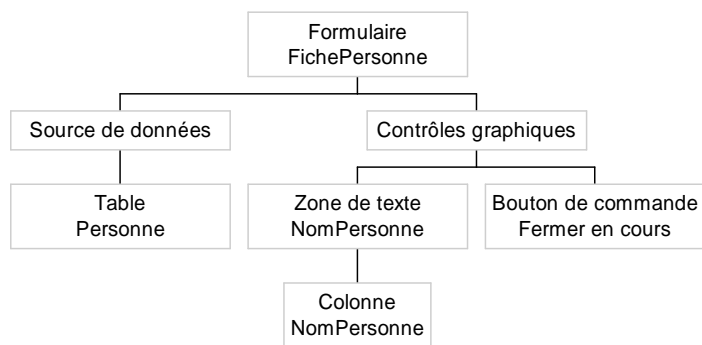
Vérifiez avec l'éditeur de propriétés que la table *Personne* est attachée au formulaire.

Regardez avec l'éditeur de propriétés comment l'assistant a codé les actions exécutées lorsque l'on clique sur les boutons.

Avec Access, on peut définir les actions de deux manières :

- *soit en programmant en Access Basic, le langage de programmation d'Access. C'est la formule choisie par les assistants intégrés dans Access,*
- *soit en créant des macrocommandes, c'est la formule que nous choisirons car, en général, elle est rapide et ne demande pas d'apprendre Access Basic.*

L'emboîtement des composants



1.10 Pour en finir avec cette introduction

Dans cette introduction, nous avons défini la notion de système d'information et plus particulièrement abordé le sous-système d'information automatisé qu'est une base de données. Nous avons expliqué la logique de construction d'un tel sous-système avec un outil comme Access :

- La construction d'une application base de données avec Access suit une logique d'emboîtement de composants logiciels préprogrammés et paramétrables (tables, requêtes, formulaires).
- Le programmeur Access conçoit la structure des tables.
- Il réalise des formulaires en leur associant des données (sous la forme d'une table ou d'une requête attachée) et des contrôles graphiques.
- Les contrôles graphiques permettent de visualiser les données dans les formulaires et de déclencher des traitements.

1.11 Questions de révision sur l'introduction

Répondre par oui ou non aux questions suivantes :

1. Une base de données peut être utilisée simultanément par plusieurs personnes connectées au même réseau ?
2. Si deux personnes modifient en même temps le même enregistrement d'une table, le premier qui a commencé à modifier l'information a la priorité ?
3. Sur le disque dur, chaque table de la base de données est enregistrée dans un fichier séparé ?
4. On peut modifier la valeur enregistrée dans une colonne de type NuméroAuto ?
5. Un formulaire permet d'afficher les données enregistrées dans une table ?

La relation

EstLaPropriétéDe entre les tables Etalon et Personne

| Etalon | |
|---------|----------------|
| Nom | IdPropriétaire |
| Achille | 2 |
| Biésolo | 8 |
| Hurgo | 8 |

Achille est la propriété
de Mr Gallois

| Personne | |
|------------|---------|
| IdPersonne | Nom |
| 2 | Gallois |
| 8 | Dibango |

Biésolo est la
propriété de
Mr Dibango

2. Structuration des données

2.1 Rappels sur les tables

Chaque entité du système d'information est modélisée par une table (par exemple, la table Personne). Ses propriétés sont définies par les colonnes de la table (par exemple, NomPersonne). On distingue une colonne particulière, la clé primaire, qui permet d'identifier de manière unique chaque occurrence de l'entité (par exemple, IdPersonne).

2.2 Objectif du chapitre

On veut maintenant pouvoir représenter les relations existant entre les entités du système d'information. Par exemple, on veut pouvoir représenter la relation *EstLaPropriétéDe* qui associe une occurrence de l'entité Etalon à une occurrence de l'entité Personne.

La manière de représenter cette relation dépend du type de la relation qui associe les deux entités. On distingue deux types de relation :

- la relation entre entités de type un à plusieurs
- la relation entre entités de type plusieurs à plusieurs

2.3 Représentation des relations de type un à plusieurs entre entités

2.3.1 Introduction

On peut être amené à représenter des relations un à plusieurs entre les entités.

*Par exemple, on veut représenter que l'étalon Achille **est la propriété de** la personne Gallois. On suppose qu'un étalon est la propriété d'une seule personne. Par contre, une même personne peut être propriétaire de plusieurs étalons. On dit dans ce cas qu'il existe une relation asymétrique **de type 1 à plusieurs** (que l'on peut nommer EstLaPropriétéDe) entre l'entité Personne et l'entité Etalon.*

2.3.2 Notion de relation un à plusieurs

Définition : Il existe une relation asymétrique **de type un à plusieurs** (que l'on peut nommer Relie) entre l'entité A et l'entité B si toute occurrence de l'entité B est reliée **au maximum à une et une seule occurrence** de l'entité A. Par contre, les occurrences de l'entité A peuvent être reliées à un nombre quelconque d'occurrences de l'entité B.

2.3.3 Représentation de la relation un à plusieurs

Pour représenter ce type de liaison, on **ajoute une colonne** dans la table du côté **plusieurs**.

Dans l'exemple, on ajoute la colonne IdPropriétaire dans la table Etalon.

On appelle cette nouvelle colonne **clé étrangère** car elle contient la valeur de la clé primaire identifiant l'entité du côté **un** de la relation.

Dans l'exemple, il s'agit de la colonne IdPersonne de la table Personne.

2.3.4 Notion de clé étrangère

Définition : Dans une table B, une colonne **clé étrangère** permet de référencer une ligne d'une table A. De cette manière, la clé étrangère matérialise la relation de type un à plusieurs existant entre la table A et la table B. La clé étrangère contient la valeur de la clé primaire de la ligne référencée dans la table A.

2.3.5 Manipulations : l'exemple de la table Etalon

La table *Etalon* contient toutes les informations que l'on désire conserver sur les étalons (nom, caractéristiques, tarif, ...).

Créez la table *Etalon* depuis la fenêtre **Base de données**. Entrez les noms de 3 colonnes de la table *Etalon*, à savoir:

- IdEtalon (la clé de type **NuméroAuto**),
- NomEtalon (de type **Texte de 30 caractères**),
- IdPropriétaire (de type **Numérique, entier long**).

Définissez la colonne IdEtalon en tant que clé primaire de la table.

Remarque : Comme la colonne IdPersonne de la table Personne est de type **NuméroAuto**, il faut définir la colonne IdPropriétaire (qui la référence) par un type compatible. Il s'agit du type **numérique, entier long**.

Sauvegardez la table *Etalon*.

2.4 Maintien de la cohérence des relations entre les tables

2.4.1 Le problème de la cohérence

Si dans la colonne `IdPropriétaire` de la table `Etalon`, l'utilisateur enregistre la valeur 3 et que cette valeur ne correspond pas à un numéro de `Personne` enregistrée dans la table `Personne`, on introduit une incohérence dans la base.

Access peut contrôler automatiquement que les valeurs prises dans la colonne `IdPropriétaire` de la table `Etalon` correspondent bien à des valeurs existantes dans la colonne `IdPersonne` de la table `Personne`.

2.4.2 Notion de contrôle d'intégrité référentielle

Définition : On appelle contrôle **d'intégrité référentielle** la vérification automatique suivante, réalisée par le moteur de la base de données : toute valeur enregistrée dans une colonne de type clé étrangère doit correspondre à une valeur stockée dans la colonne de type clé primaire référencée par la clé étrangère.

2.4.3 Manipulations

Par le menu **Outils**, option **Relations**, sélectionnez les deux tables dans la fenêtre **Ajouter une table**.

Cliquez sur la colonne `IdPersonne`, clé primaire de la table `Personne` et glissez vers la colonne `IdPropriétaire` de la table `Etalon`.

Attention : Le sens dans lequel on établit la relation est important car il s'agit d'une relation asymétrique de type un à plusieurs (ie à un propriétaire peut correspondre plusieurs étalons, la réciproque étant fausse).

La boîte de dialogue Relation doit être ouverte. Si ce n'est pas le cas, double-cliquez sur la relation inter-tables. Dans cette boîte, il est possible de définir plusieurs propriétés sur la relation (contrôle de l'intégrité référentielle, cascades de mise à jour et suppression, type de jointure).

Validez l'option ***Appliquer l'intégrité référentielle*** et cliquez sur le bouton ***Créer***. Fermez la fenêtre des relations.

Ouvrez la table *Étalon* en mode ***Feuille de données***.

Créez deux étalons *Achille* et *Biésolo* ayant pour propriétaire *Bernadou Lucien*. Essayez de modifier le propriétaire d'un étalon en tapant un numéro non existant dans la table *Personne*.

2.4.4 Maintien de la cohérence en cas de modification ou de suppression dans les tables référencées

Essayez de supprimer une personne de la table *Personne* référencée comme propriétaire d'étalon dans la table *Etalon*.

Essayez de supprimer un étalon ayant un propriétaire dans la table *Etalon*.

Que faut-il en conclure ?

Par le menu **Outils**, option **Relations**, double-cliquez sur la relation inter-tables *EstLaPropriétéDe* entre les tables *Personne* et *Etalon*.

Inspectez les propriétés qui peuvent être définies sur la relation (contrôle de l'intégrité référentielle, cascades de mise à jour et suppression).

Rôle de l'option **Mettre à jour en cascade** : Lorsque l'on modifie l'identifiant d'un enregistrement de la table du côté 1 alors Access modifie automatiquement toutes les références à cet enregistrement dans la table du côté plusieurs.

Remarque : Dans le cas de la relation entre la table *Personne* et la table *Etalon*, cette option n'a pas d'intérêt car l'identifiant de la table *Personne* est de type **NuméroAuto**. Par définition, une fois attribué, il ne peut pas être modifié.

Rôle de l'option **Effacer en cascade** : Lorsque l'on supprime un enregistrement de la table du côté 1 alors Access supprime automatiquement tous les enregistrements de la table du côté plusieurs qui lui sont associés par la relation.

Faut-il valider cette option pour la relation qui lie *Personne* à *Etalon* ?

La relation EstLaPropriétéDe entre les tables Etalon et Personne

| Etalon | |
|----------|---------|
| IdEtalon | Nom |
| 1 | Biésolo |
| 2 | Achille |

| Personne | |
|------------|---------|
| IdPersonne | Nom |
| 2 | Gallois |
| 8 | Dibango |

| EstLaPropriétéDe | |
|------------------|----------------|
| IdCheval | IdPropriétaire |
| 1 | 8 |
| 2 | 2 |
| 2 | 8 |

Biésolo est la propriété de Mr Dibango →

← *Achille est la propriété de Mr Gallois et de Mr Dibango*

2.5 Représentation des relations de type plusieurs à plusieurs entre entités

2.5.1 Introduction

Reprenons l'exemple de la relation *EstLaPropriétéDe* entre la table Etalon et la table Personne. Supposons cette fois-ci qu'un étalon peut être la propriété de plusieurs personnes. Dans ce cas, la relation *EstLaPropriétéDe* est de type **plusieurs à plusieurs**.

La solution proposée précédemment pour représenter la relation n'est pas suffisante. La colonne IdPropriétaire de la table Etalon ne permet de stocker qu'un seul numéro de personne.

2.5.2 Notion de relation plusieurs à plusieurs

Définition : Il existe une relation symétrique dite **de type plusieurs à plusieurs** (que l'on peut nommer Relie) entre l'entité A et l'entité B s'il existe au moins une occurrence de l'entité B qui peut être reliée à plusieurs occurrences de l'entité A et réciproquement.

2.5.3 Représentation d'une relation plusieurs à plusieurs

On matérialise une relation plusieurs à plusieurs **en créant une table supplémentaire**. Cette table contient deux colonnes qui référencent les clés primaires des deux tables participant à la relation. Ces deux colonnes forment **la clé primaire** de la nouvelle table et sont clés étrangères vis-à-vis des deux tables qu'elles référencent.

2.5.4 Manipulations : création de la table *EstLaPropriétéDe*

Créer une nouvelle base *haras2.mdb* avec une table *Personne* (avec 2 colonnes *IdPersonne*, clé primaire et *NomPersonne*) et une table *Etalon* (avec 2 colonnes *IdEtalon*, clé primaire et *NomEtalon*).

Créez une nouvelle table. Nommez-la *EstLaPropriétéDe*. Créez deux colonnes :

- *IdCheval*, de type numérique, entier long,
- *IdPropriétaire*, de type numérique, entier long.

Définissez ces deux colonnes en tant que clé primaire de la table *EstLaPropriétéDe*. Puis sauvegardez la table.

Par le menu **Outils**, option **Relations**, paramétrez le contrôle d'intégrité référentielle entre clés primaires et clés étrangères :

- Ajoutez les tables *Personne*, *Etalon* et *EstLaPropriétéDe* dans le panneau **Relations**.
- Cliquez sur la colonne *IdPersonne*, clé primaire de la table *Personne* et glissez vers la colonne *IdPropriétaire* de la table *EstLaPropriétéDe*.
- Validez l'option **Appliquer l'intégrité référentielle**.
- Faites de même pour les colonnes *IdEtalon* et *IdCheval*.

Mémorisez dans les tables de la base le fait que *Achille* a pour propriétaires *Gallois* et *Dibango* et que *Biésolo* a pour propriétaire *Dibango*.

Essayez de modifier le propriétaire d'un étalon en tapant un numéro non existant dans la table *Personne*.

Fermez la base *haras2.mdb*.

2.5.5 Nouvelle définition d'une clé primaire


Définition : La clé primaire d'une table est le **plus petit sous-ensemble** de colonnes de cette table qui détermine **de manière unique** les autres colonnes de la table.

Exemple 1 : La colonne IdPersonne est clé primaire de la table Personne.

Exemple 2 : Si nous voulons représenter la date d'acquisition par un copropriétaire d'une part d'un étalon, il suffit d'ajouter dans la table EstLaPropriétéDe une colonne supplémentaire DateAchat. Une occurrence de DateAchat est déterminée de manière unique si l'on connaît à la fois la valeur associée à IdCheval et celle associée à IdPropriétaire. Le couple (IdCheval, IdPropriétaire) est donc bien clé primaire de la table EstLaPropriétéDe.



Récapitulation

- ◆ Chaque entité du système d'information est représentée par une table.
 - ◆ Chaque table contient obligatoirement une colonne particulière (appelée clé primaire) qui identifie de manière unique chaque exemplaire de l'entité modélisée.
 - ◆ Les tables peuvent être liées entre elles par des relations.
- 

2.6 Récapitulation

Les données sont stockées dans des tables correspondant aux entités du système d'information modélisé.

Access est une **base de données relationnelle**. On peut créer des relations entre les entités modélisées.

On distingue deux types de relation : les relations de type un à plusieurs et les relations de type plusieurs à plusieurs.

Règle 1: Lorsque l'on veut définir une relation asymétrique entre deux entités de type **un à plusieurs**, on ajoute une colonne de liaison dans la table se trouvant du côté **plusieurs**.

Exemple : On suppose que la relation entre la table *Personne* et la table *Etalon* est de type **un à plusieurs**. Dans ce cas, à une occurrence de personne, on peut associer plusieurs occurrences d'étalons. L'inverse n'est pas vrai. Il s'agit donc d'une relation asymétrique. On ajoute la colonne *IdPropriétaire* dans la table *Etalon* pour matérialiser la relation *EstLaPropriétéDe*.

Règle 2: Lorsque l'on veut définir une relation symétrique entre deux entités de type **plusieurs à plusieurs**, on crée une nouvelle table contenant au moins deux colonnes. Ces colonnes référencent les clés primaires des deux tables mises en relation. Elles constituent la clé primaire de la table représentant la relation.

Exemple : On suppose que la relation entre la table *Personne* et la table *Etalon* est de type **plusieurs à plusieurs**. Dans ce cas, à une occurrence de personne, on peut associer plusieurs occurrences d'étalons et réciproquement. Il s'agit donc d'une relation symétrique. Pour matérialiser la relation, on crée une table *EstLaPropriétéDe* constituée de deux colonnes référencant les clés primaires des tables *Etalon* et *Personne*.

2.7 Exercice : la gestion d'un ensemble d'exploitations agricoles

On désire modéliser dans une base de données des informations sur des exploitations agricoles, leur propriétaire, les employés et les parcelles. Dans une nouvelle base Access (agro.mdb par exemple), définissez la structure des tables permettant de modéliser ces entités.

On décrit :

1. l'exploitation par un numéro, un nom, une adresse, un code postal et un bureau distributeur,
2. un employé par un numéro de sécurité sociale, un nom, un prénom, une adresse, un numéro de téléphone,
3. un propriétaire par un numéro de sécurité sociale, un nom, un prénom, une adresse et un numéro de téléphone,
4. une parcelle par un numéro, un nom, une superficie, une teneur en nitrate, phosphate, potassium, calcium, magnésium, et sulfure.

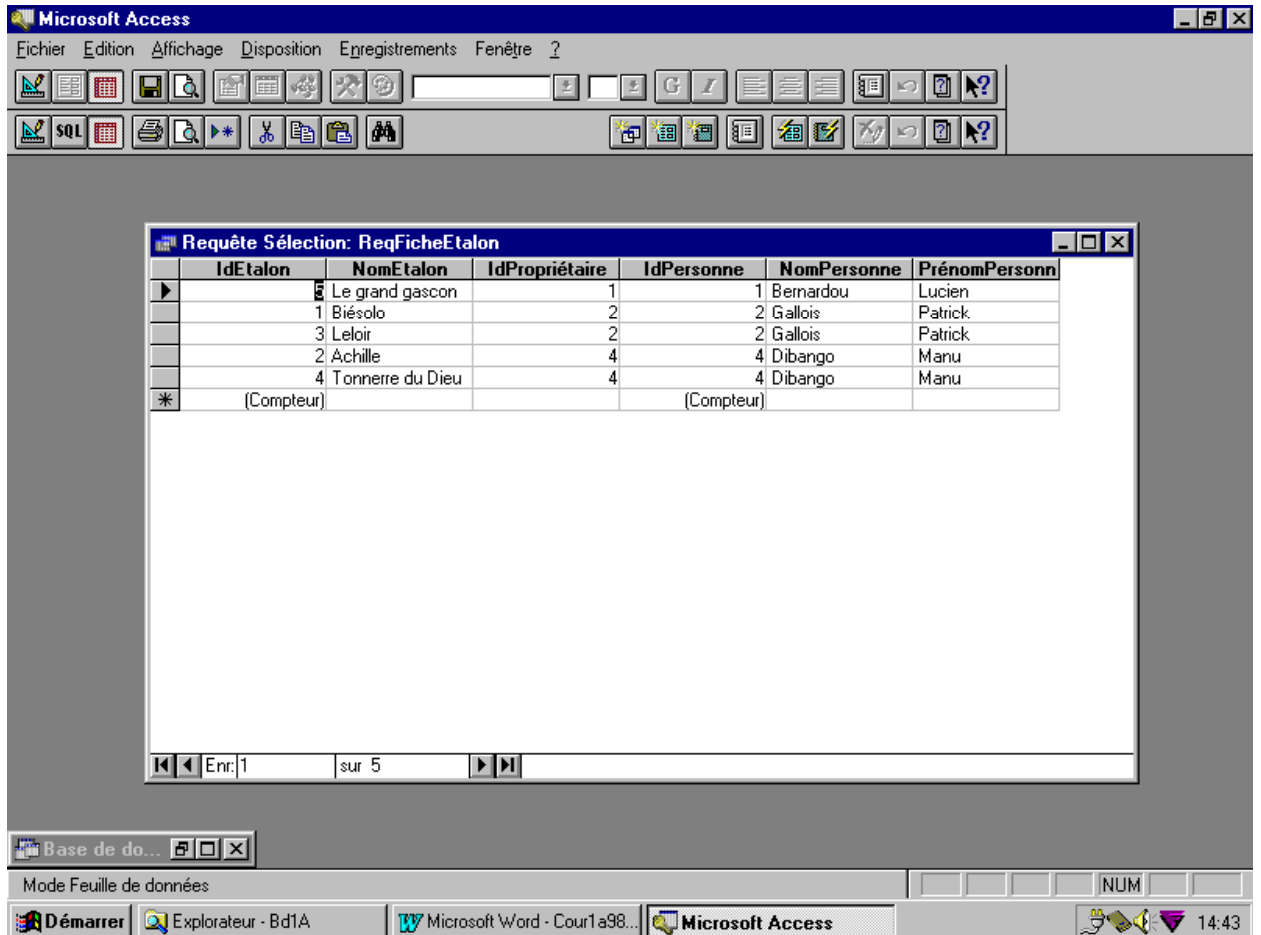
On veut également représenter les relations suivantes :

1. Une parcelle appartient à un ou plusieurs propriétaires et une personne peut être propriétaire d'une ou plusieurs parcelles.
2. Un employé travaille sur une ou plusieurs exploitations, et pour chaque exploitation, on mémorise le temps travaillé (en nombre d'heures) et la date d'embauche. Une exploitation peut employer une ou plusieurs personnes.
3. Une parcelle fait partie d'une et d'une seule exploitation.

2.8 Questions de révision sur la structuration des données

Répondre aux questions suivantes par oui ou non :

1. Deux enregistrements d'une même table peuvent avoir même valeur pour la clé primaire ?
2. Access rend obligatoire la définition d'une clé primaire lorsque l'on crée une table ?
3. Une clé primaire peut être constituée de plusieurs colonnes de la table ?
4. Soient les tables Client et Produit, la relation EstCommandéPar est représentée par l'ajout d'une colonne IdClient dans la table Produit référençant la clé primaire de la table Client ?
5. Soient les tables Personne et Etalon reliées par la relation de type un à plusieurs avec contrôle de l'intégrité référentielle, Access accepte-t-il la suppression d'un propriétaire d'étalons ?



3. Les requêtes

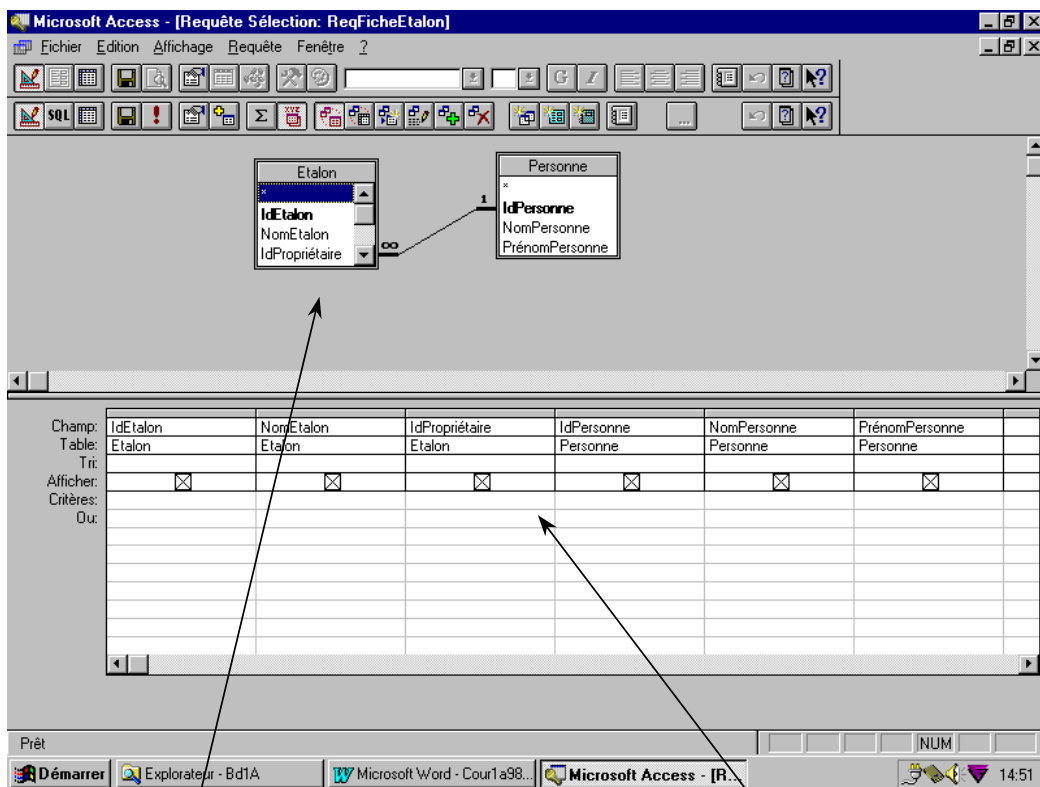
3.1 Vues sur la base de données

Dans une application base de données, on définit deux points de vue sur les données :

- Celui de l'administrateur des données : il voit la base de données dans toute sa complexité ; il connaît l'intégralité de la structure de la base de données.
- Celui de l'utilisateur de la base de données : il ne voit que les données auxquelles il a besoin d'accéder. Pour cela, l'administrateur lui construit une **vue simplifiée** sur la base de données avec une **requête de sélection**.

Par exemple, le client du haras consulte la base de données pour connaître les caractéristiques des étalons. Pour cela, l'administrateur construit une vue sur la base de données dans laquelle il sélectionne et rassemble les informations disponibles dans la base concernant l'étalon qu'il désire rendre accessible au client.

Pour construire une vue, on crée une **requête de sélection** qui rassemble dans une table virtuelle les données stockées dans des tables de la base. On parle de table **virtuelle** car les informations obtenues par une requête de sélection sont présentées sous la forme d'une table.



Liste des tables rassemblées dans la requête

Liste des colonnes regroupées dans la requête

3.2 Définition d'une requête de sélection

Les moteurs base de données sont capables d'exécuter plusieurs types de requêtes (sélection, mise à jour, création de table, ajout dans une table, suppression dans une table avec Access). Dans ce cours, seules les requêtes de sélection sont abordées.

Définition 1 : Une **requête de sélection** permet de sélectionner, rassembler, trier des colonnes provenant de plusieurs tables dans une table virtuelle. Le résultat de l'exécution d'une requête de sélection se présente sous la forme d'une table virtuelle constituée d'un ensemble de lignes. Chaque ligne est composée d'un ensemble de valeurs, chaque valeur correspondant à une colonne de la requête. Une requête peut également permettre d'effectuer des regroupements de lignes ayant même valeur pour une colonne donnée afin d'appliquer une fonction de calcul sur chacun des groupes de lignes constitués.

Exemple : la requête ReqFicheEtalon rassemble les colonnes des tables Etalon et Personne.

Définition 2 : Le rassemblement des valeurs associées aux colonnes provenant des tables faisant partie de la requête est obtenu par une **opération de jointure**. Pour cela, il faut définir une **colonne de jointure** dans chacune des tables. La jointure rassemble sur une même ligne du résultat de la requête les lignes des tables participantes ayant même valeur dans leurs colonnes de jointure. L'opération de jointure est effectuée au cours de l'exécution de la requête.

Exemple : Dans une jointure entre les tables Etalon et Personne, on définit comme colonnes de jointure IdPersonne dans la table Personne et IdPropriétaire dans la table Etalon. La ligne (1, Biésolo, 2) de la table Etalon est jointe à la ligne (2, Gallois) de la table Personne par l'opération de jointure.

3.3 Manipulations : Création de la requête ReqFicheEtalon

Une **requête de sélection** est un composant Access accessible à partir de la fenêtre base de données.

Dans cette manipulation, on suppose que la relation *EstLaPropriétéDe* entre *Etalon* et *Personne* est de type **un à plusieurs**. Ouvrez la base *haras.mdb*.

Depuis la fenêtre **Base de données**, cliquez sur l'onglet **Requêtes**. Puis cliquez sur le bouton **Nouveau**. Sélectionnez dans la liste le **mode Création**.

Sélectionnez dans la fenêtre **Ajouter une Table** les tables auxquelles vous avez besoin d'accéder, à savoir, *Personne* et *Etalon*.

Remarquez qu'Access a automatiquement placé les relations entre les tables. En effet, si Access trouve des définitions de contrôle d'intégrité entre les tables rassemblées dans la requête, il s'en sert pour relier les tables.

Sélectionnez les attributs qui apparaîtront à l'exécution de la requête.

Sélectionnez tous les attributs de la table *Etalon*. Pour cela, cliquez sur le titre de la table. Toutes les colonnes de la table sont alors sélectionnées. Glissez-les dans le tableau se trouvant dans la partie inférieure de la fenêtre.

Sélectionnez et glissez dans le tableau tous les attributs de la table *Personne*.

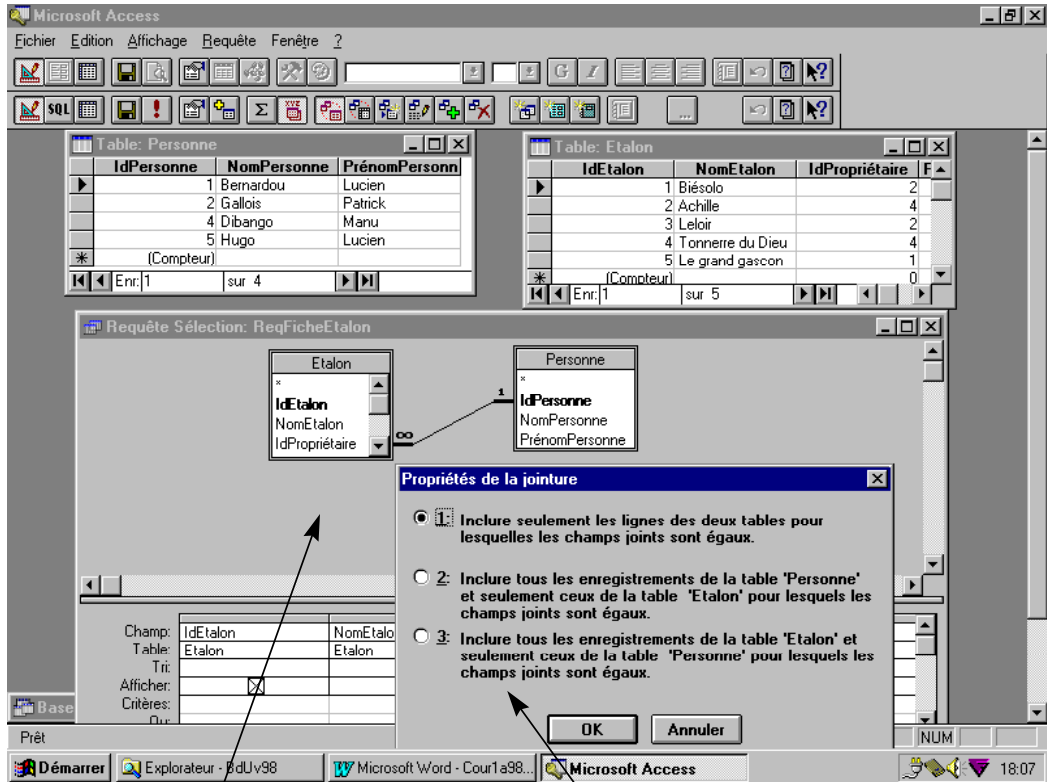
Sauvegardez votre requête par l'option **Enregistrer sous** du menu **Fichier** sous le nom *ReqFicheEtalon*.

Exécutez la requête en cliquant sur l'icône **Point d'exclamation** de la barre d'outil (**Exécuter** dans la bulle d'aide). Vous devez voir apparaître la liste de tous les couples étalons/propriétaires de la base.

Access a regroupé sur la même ligne de la table résultat de la requête les lignes des tables qui ont même valeur pour les colonnes de liaison.

Par exemple : Biésolo est relié à Dibango car Access a relié la ligne (1, Biésolo, 4) de la table Etalon à la ligne (4, Dibango) de la table Personne.

Remarquez qu'il est possible de modifier les valeurs affichées dans la requête. Vérifiez que ces modifications sont prises en compte dans les tables.



Liste des tables rassemblées dans la requête

Liste des types de jointure possibles entre les tables

3.4 Plusieurs types de jointures possibles entre les tables

Définition 1 : Une jointure de type 1 est symétrique : une ligne du résultat de la jointure contient un couple de lignes des tables jointes si celles-ci portent la même valeur dans les colonnes jointes.

Exemple : Dans une jointure de type 1 entre Etalon et Personne (1, Biésolo, 2) est joint à (2, Gallois) par contre (5, Le Grand Gascon, null) ne figure pas dans la table résultat de la jointure car la colonne IdPropriétaire n'est pas renseignée (null veut dire que la colonne n'est pas renseignée).

Définition 2 : Une jointure de type 2 est asymétrique ; elle privilégie la table du **côté un** : toutes les lignes de la table du côté un font partie au moins d'une ligne du résultat de la jointure. Une ligne de la table résultat contient éventuellement une ligne de la table du côté plusieurs si celle-ci porte dans sa colonne jointe la même valeur que la colonne jointe de la ligne provenant de la table du côté un.

Exemple : Une jointure de type 2 entre Personne et Etalon affiche **toutes** les personnes, même celles qui ne possèdent pas d'étalon. (2, Gallois) est joint à (1, Biésolo, 2); (5, Dupont) figure également dans la table résultat de la jointure même si aucune des lignes de la table Etalon ne fait référence à la personne n°5.

Définition 3 : Une jointure de type 3 est asymétrique ; elle privilégie la table du **côté plusieurs**: toutes les lignes de la table du côté plusieurs font partie au moins d'une ligne du résultat de la jointure. Une ligne de la table résultat contient éventuellement une ligne de la table du côté un si celle-ci porte dans sa colonne jointe la même valeur que la colonne jointe de la ligne provenant de la table du côté plusieurs.

Exemple : Une jointure de type 3 entre Etalon et Personne affiche **tous** les étalons, même ceux pour lesquels la colonne IdPropriétaire n'est pas renseignée. (1, Biésolo, 2) est joint à (2, Gallois); (5, Le Grand Gascon, null) figure également dans la table résultat de la jointure alors que la colonne IdPropriétaire n'est pas renseignée.

3.5 Manipulations : paramétrer le type de jointure

Double-cliquez sur la liaison entre les tables. Vous constatez que trois types de liaisons (encore appelées jointures) sont possibles entre les tables lorsque l'on exécute la requête.

Exécutez la requête en modifiant à chaque fois le type de jointure. Vérifiez les différences entre les résultats obtenus. Les valeurs stockées dans les tables Etalon et Personne présentées sur la copie d'écran de la page précédente permettent d'obtenir un résultat différent pour chaque type de liaison.

3.6 Possibilités de sélection et d'interrogation multi-critères

On peut spécifier des critères de sélection dans une requête. Pour cela, il suffit d'indiquer la valeur recherchée sur la ligne Critères de la colonne sur laquelle on veut opérer une sélection.

*Exemple : Pour lister les étalons dont Dibango est propriétaire, on tape sur la ligne Critères de la colonne NomPersonne la valeur **Dibango**.*

Le caractère * est un joker qui remplace une suite quelconque de caractères.

*Exemple : Pour lister les étalons dont le nom commence par un B, on tape sur la ligne Critères de la colonne NomEtalon la valeur **B***.*

Si l'on veut faire une interrogation multi-critères séparés par des **et**, on place les critères de sélection sur la **même** ligne Critères.

*Exemple : Pour lister les étalons dont le nom commence par un B **et** dont Dibango est propriétaire, on tape sur la ligne Critères de la colonne NomPersonne la valeur Dibango et sur la **même** ligne Critères de la colonne NomEtalon la valeur B*.*

Si l'on veut faire une interrogation multi-critères séparés par des **ou**, on place les critères de sélection sur deux lignes Critères **différentes**.

*Exemple : Pour lister les étalons dont le nom commence par un B **ou** dont Dibango est propriétaire, on tape sur la ligne Critères de la colonne NomPersonne la valeur Dibango et sur la ligne Critères **suivante** dans la colonne NomEtalon la valeur B*.*

3.7 Possibilités de tri

Pour trier le tableau résultat de la requête sur une colonne, il suffit de sélectionner l'attribut Croissant ou Décroissant sur la ligne Tri correspondant à la colonne à trier.

Remarque : Si plusieurs colonnes sont triées, l'ordre de tri se fait de la gauche vers la droite.

3.8 Possibilités de calcul

Il est possible de réaliser des calculs simples dans une requête (dénombrement, somme, moyenne). Le résultat de la requête de sélection est une valeur calculée sur l'ensemble des lignes.

3.9 Possibilités de calcul sur regroupement de lignes

Un regroupement de lignes est constitué par l'ensemble des lignes qui portent la même valeur dans toutes les colonnes participant au regroupement. Le résultat de la requête de sélection est constitué par l'ensemble des calculs effectués sur les lignes de chaque regroupement. Il y a autant de résultat de calcul que de valeurs différentes pour le critère de regroupement.

3.10 Manipulation

On définit une colonne de regroupement en sélectionnant l'opération **Regroupement** sur la ligne **Opération**. On obtient la ligne Opération par le menu Affichage, action Opération.

Par exemple : on désire connaître le nombre de chevaux par propriétaire

1. Sélectionnez uniquement les colonnes NomPersonne et IdEtalon dans la requête,
2. sélectionnez l'opération Regroupement pour la colonne NomPersonne.
3. sélectionnez l'opération Compte pour la colonne IdEtalon.

3.11 Exercices

Recopiez la base *agro.mdb* se trouvant dans le répertoire indiqué par l'enseignant. Sur cette base d'exploitations agricoles, programmez les requêtes suivantes :

1. Liste triée de tous les employés travaillant dans l'exploitation Boldevitz depuis le 1/4/98,
2. Liste triée de tous les employés avec le nom de l'exploitation qui les emploie,
3. Liste triée de toutes les employés avec le cas échéant le nombre total d'heures travaillées,
4. Liste des exploitations triées sur le nom avec superficie totale,
5. Liste des couples de parcelles de même superficie,
6. Liste des propriétaires triés sur le cumul décroissant des superficies totales possédées.

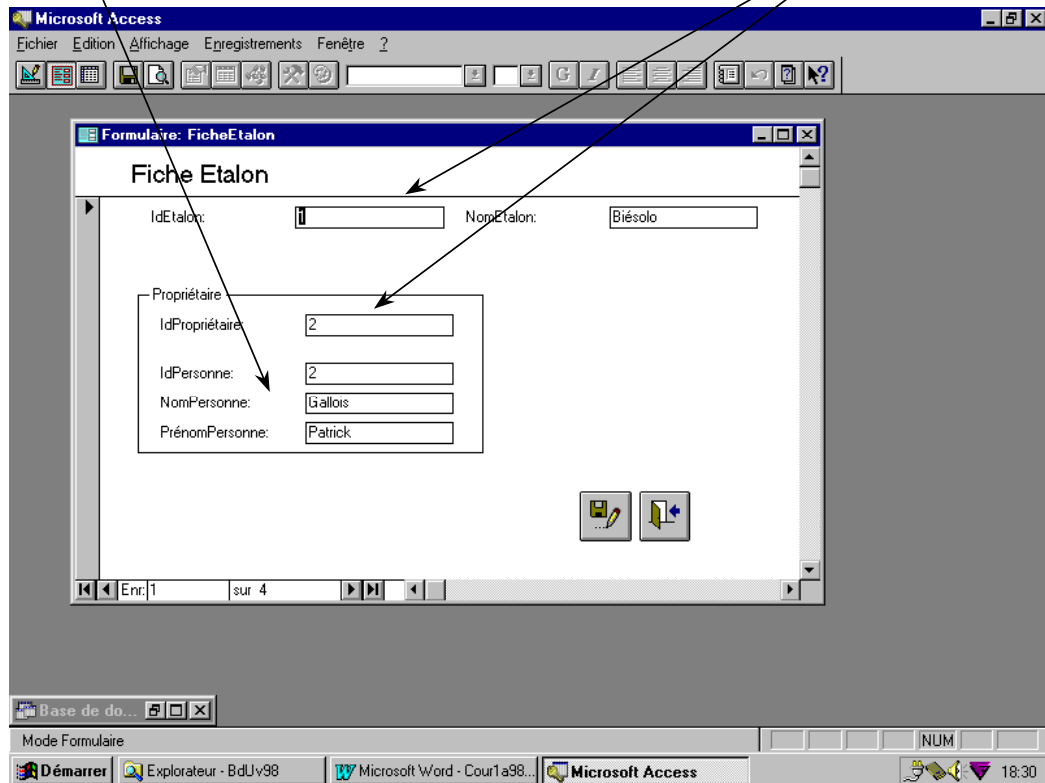
3.12 Questions de révision sur les requêtes

Répondez par oui ou non aux questions suivantes :

1. Une requête de sélection permet de regrouper dans une table virtuelle des colonnes provenant de tables différentes ?
2. Une liaison de type 1 entre les tables Personne et Etalon permet d'afficher uniquement les propriétaires d'étalons ?
3. Une liaison de type 2 entre les tables Personne et Etalon permet d'afficher toutes les personnes enregistrées dans la table Personne et les étalons dont elles sont propriétaires le cas échéant ?
4. Dans une requête, on ne peut établir une jointure entre deux tables que sur des colonnes liées par une contrainte d'intégrité référentielle ?
5. Lorsque l'on demande un tri sur trois colonnes de la requête, le tri n'est effectué que sur la colonne la plus à gauche ?

Table Personne

Table Etalon



4. Formulaire

4.1 Rappels sur les formulaires

Un formulaire est une fenêtre Windows dans laquelle on affiche **une source de données** (table ou requête).

Il est composé de **contrôles graphiques**. Un contrôle graphique est un composant paramétré qui est affiché sur le formulaire.

Exemple : dans le formulaire FichePersonne, chaque colonne de la table Personne est affichée dans un contrôle graphique de type zone de texte.

4.2 Source de données

Définition 1 : La source de données est la table ou la requête de sélection qui contient les données affichées dans le formulaire.

Les premières questions que l'on doit se poser lorsque l'on fabrique un formulaire sont :

- *Quelles informations veut-on afficher ?*
- *Dans quelles tables se trouvent-elles ?*

Définition 2 : Si toutes les informations que l'on désire afficher se trouvent dans une seule table, la source de données du formulaire est cette table.

Exemple : la source de données du formulaire FichePersonne est la table Personne

Définition 3 : Si les informations que l'on désire afficher sont réparties dans plusieurs tables, la source de données du formulaire est une **requête de sélection** regroupant ces informations.

Exemple : la source de données du formulaire FicheEtalon est la requête ReqFicheEtalon

Microsoft Access - [Requête Sélection: ReqFicheEtalon]

Fichier Edition Affichage Requête Fenêtre ?

| Champ: | IdEtalon | NomEtalon | IdPropriétaire | IdPersonne | NomPersonne | PrénomPersonne | |
|-----------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|--|
| Table: | Etalon | Etalon | Etalon | Personne | Personne | Personne | |
| Tri: | | | | | | | |
| Afficher: | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | |
| Critères: | | | | | | | |
| Où: | | | | | | | |

Prêt

Démarrer Explorateur - EnC... Microsoft Word - ... Eudora - [In] Microsoft Acc... Microsoft PowerP... 19:25

4.2.1 Manipulations : création du formulaire FicheEtalon

Le formulaire *FicheEtalon* permet d'afficher dans une fenêtre Windows les données regroupées dans la requête *ReqFicheEtalon*.

Cliquez sur l'onglet **Formulaires** de la fenêtre **Base de données**.

Cliquez sur le bouton **Nouveau**.

Sélectionnez la requête *ReqFicheEtalon* dans la liste déroulante et sélectionnez dans la liste le **mode création**.

En mode création, cliquez sur le bouton Liste des champs de la barre d'outils. Sélectionnez tous les noms de colonnes s'y trouvant et glissez-les sur le formulaire.

Identifiez les trois modes de visualisation d'un formulaire dans la barre d'outils (bouton **affichage**)

- *mode création* : il permet de concevoir le formulaire,
- *mode formulaire* : il permet de visualiser les données d'une même ligne en mode fiche,
- *mode feuille de données* : il permet de visualiser les données en mode tableau.

Sauvegardez (icône Disquette) en nommant le formulaire *FicheEtalon*.
Exécutez-le (icône Formulaire dans la barre d'outil).

Nous allons ajouter deux boutons de commande sur le formulaire. Affichez la boîte à outils (cliquez sur l'icône boîte à outils). Activez les assistants contrôle (bouton *baguette magique* dans la boîte à outils). Ajoutez un bouton **Enregistrer** et **Fermer le formulaire**.

Enregistrez un nouvel étalon et son propriétaire.

Quelle amélioration devrait-on apporter pour faciliter la saisie ?

4.2.2 Manipulations : accès à la source de données avec l'éditeur de propriétés

Dans les manipulations de création de formulaires, nous avons utilisé l'option **mode création**. Dans le dialogue avec l'assistant, nous avons indiqué la source de données associée au formulaire.

Lorsque le formulaire est déjà construit, on peut accéder à sa source de données avec l'éditeur de propriétés.

Le formulaire doit être ouvert en mode création. L'éditeur de propriétés (bouton Propriétés) doit être positionné sur les propriétés du formulaire. Pour cela, il faut cliquer dans le carré blanc (à gauche de la règle horizontale). Dans la fenêtre de l'éditeur, sélectionner l'onglet *toutes*, rechercher la propriété *Source*. Elle contient le nom de la source de données.

La colonne affichée dans le contrôle est la première colonne de la requête

Origine/Source : Requête

Contenu : Liste Personnes

Source contrôle : IdPropriétaire

Colonne liée : 3

| Propriétaire | | |
|--------------|----------|----|
| Crésus | | |
| Bernadou | Lucien | 1 |
| Crésus | Onassis | 11 |
| Dibango | Manu | 8 |
| Dupont | Gilles | 12 |
| Gallot | Patrick | 2 |
| Huard | Mireille | 10 |
| Letourneur | André | 9 |

Mode Formulaire

Démarrer | Explorateur - C:\MSOffice\... | Microsoft Access | Microsoft PowerPoint | 09:05

Largeur des colonnes : 2,5cm; 2,5cm; 1cm

Nbre colonnes : 3

4.3 Contrôles listes modifiables

4.3.1 Définition

Les contrôles de type liste modifiable permettent :

- d'afficher dans une liste déroulante le résultat d'une requête de sélection
- de sélectionner une ligne dans cette liste
- d'affecter au contrôle, vu comme une variable d'un langage de programmation, la valeur de l'une des colonnes de la ligne sélectionnée.
- d'associer cette valeur à une colonne de la source de données du formulaire.

4.3.2 Un premier exemple d'utilisation du contrôle liste modifiable

On veut faciliter le report dans une colonne de la source de données d'un numéro identifiant un objet. Pour cela, on sélectionne cet objet par son nom dans la liste déroulante d'un contrôle liste modifiable. Le report automatique du numéro est obtenu par paramétrage du contrôle liste modifiable.

*Exemple : Dans la version actuelle du formulaire FicheEtalon, lorsque l'on veut affecter un propriétaire à l'étalon, on doit reporter dans la colonne IdPropriétaire de la table Etalon l'identifiant du propriétaire (i.e. son numéro se trouvant dans la table Personne). Pour faciliter cette opération, nous allons utiliser un contrôle graphique de type **Liste modifiable**. Ce contrôle permet d'afficher la liste de tous les noms de personnes enregistrées dans la table Personne. Il suffira de sélectionner un nom dans la liste. Automatiquement, Access enregistrera dans la colonne IdPersonne de la table Etalon le numéro de la personne sélectionnée.*

4.3.3 Manipulations : création d'une liste modifiable

Créez une requête à partir de la table *Personne* dans laquelle vous sélectionnez les colonnes *NomPersonne*, *PrénomPersonne* et *IdPersonne*. Triez noms et prénoms par ordre croissant. Pour cela, sélectionnez dans le tableau résultat de la requête dans les deux colonnes correspondant au nom et au prénom l'option **croissant** sur la ligne **Tri**. Sauvegardez cette requête sous le nom *ListePersonnes*. Exécutez-la pour vérifier qu'elle vous donne bien le résultat désiré.

Dans le formulaire *FicheEtalon* (en mode création), désélectionnez la baguette magique dans la boîte à outils et sélectionnez un contrôle de type **zone de liste modifiable**. Placez-le à côté du contrôle zone de texte *IdPropriétaire*.

Pour paramétrer le contrôle il faut renseigner les 6 propriétés suivantes :

1) Origine/Source : Indique que les valeurs affichées dans la liste proviennent d'une requête.

Sélectionnez Table/Requête

2) Contenu : C'est le nom de cette requête.

Sélectionnez ListePersonnes

3) Nbre Colonnes : C'est le nombre de colonnes de cette requête.

Tapez 3 (car 3 colonnes dans ListePersonnes)

4) Source contrôle : C'est la colonne de la requête attachée au formulaire qui recevra la valeur sélectionnée dans la liste.

Sélectionnez IdPropriétaire

5) Colonne liée : C'est le numéro de la colonne de la requête définie dans **Contenu** dont la valeur sera recopiée dans **Source Contrôle**.

Tapez 3 (correspond à IdPersonne dans ListePersonnes)

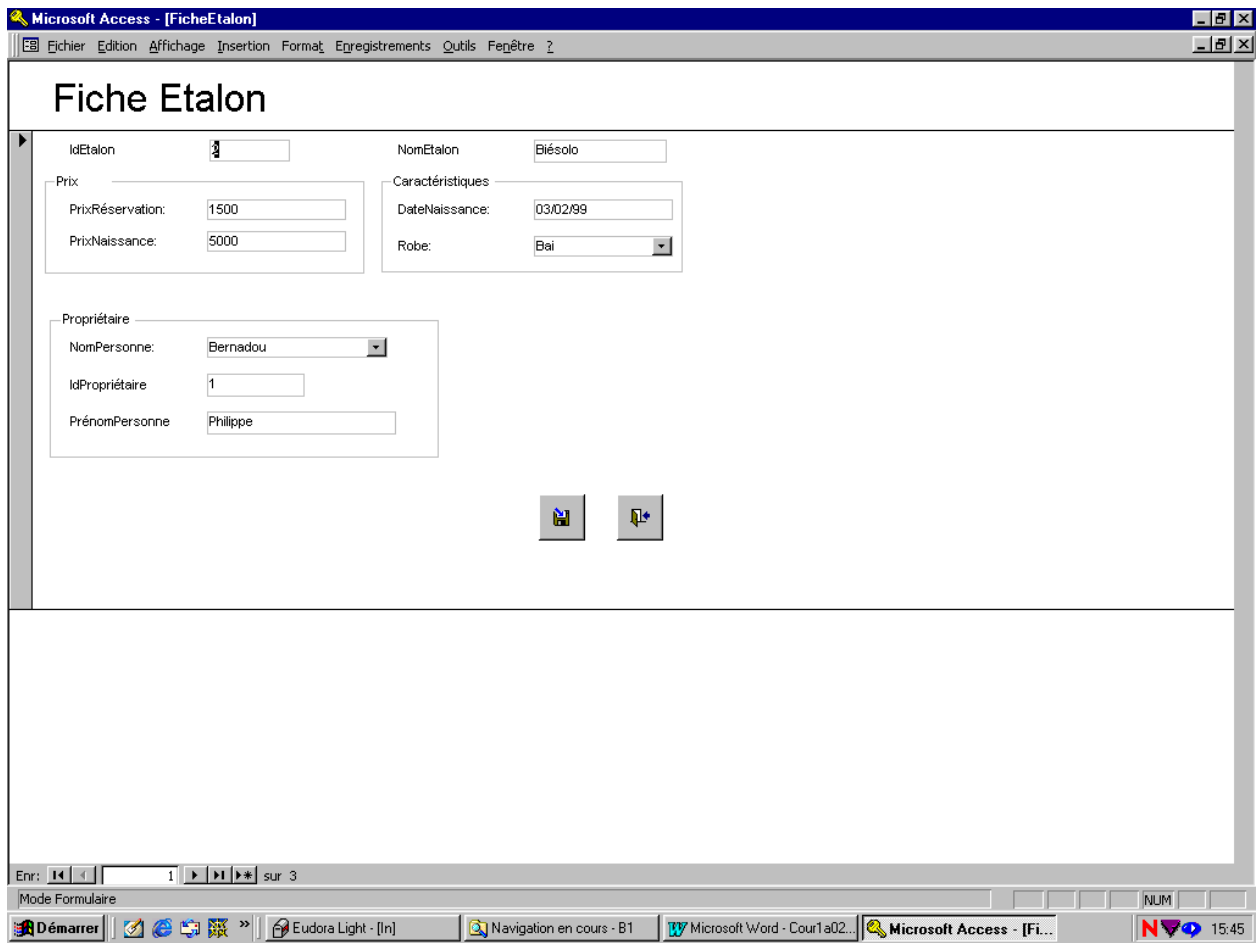
6) Largeurs de colonnes : C'est la largeur d'affichage de chacune des colonnes.

Tapez 2,5cm; 2,5cm; 0cm. Le troisième paramètre correspond à la colonne IdPersonne. Cette colonne est donc présente dans la requête mais non affichée dans la liste.

Testez le contrôle en exécutant le formulaire. Le choix dans un contrôle **Liste modifiable** peut se faire de deux manières :

- 1. On ouvre la liste et on se déplace avec l'ascenseur pour trouver le nom désiré,*
- 2. On tape dans le contrôle le ou les premiers caractère(s) du nom désiré, Access se positionne alors sur le premier nom commençant par le caractère tapé. Il faut, dans ce cas, avoir préalablement trié la liste des noms.*

Vous pouvez ensuite supprimer les contrôles zone de texte *IdPropriétaire* et *NomPersonne* qui font maintenant double emploi.



4.4 Impact de la modification de la structure de données sur l'application

Comment doit-on procéder pour pouvoir enregistrer une nouvelle information dans la base de données lorsque l'application (tables, requêtes, formulaires) est déjà construite ?

- Etape 1 : Il faut modifier la structure de la table qui va accueillir la nouvelle information dans une nouvelle colonne.
- Etape 2 : Si la source de données du formulaire est une requête, il faut ajouter la nouvelle colonne dans la requête.
- Etape 3 : Pour afficher la nouvelle colonne dans le formulaire, il faut utiliser le bouton **Liste des champs** de la barre d'outils en mode création.

4.4.1 Manipulation : modification de la structure de la table Etalon

Ouvrez la table Etalon en modification. Complétez la structure de la table *Etalon* avec 4 colonnes supplémentaires :

- PrixRéservation (*de type Numérique, réel simple*),
- PrixNaissance (*de type Numérique, réel simple*),
- DateNaissance (*de type Date/heure*),
- Robe (*de type Texte de 30 caractères*).

4.4.2 Manipulation : un deuxième exemple d'utilisation du contrôle liste modifiable dans la définition d'une table

Nous allons compléter la définition de la colonne Robe en indiquant une **liste de valeurs** possibles pour cette colonne.

Pour cela, positionnez le curseur sur la ligne correspondant à la colonne Robe. Dans les propriétés du champ (en bas de la fenêtre), cliquez sur l'onglet **Liste de choix**, afin d'indiquer un choix de valeurs possibles pour ce champ. Pour paramétrer le contrôle il faut renseigner les 3 propriétés suivantes :

1. pour la propriété *Afficher le contrôle*, choisissez l'affichage **Zone de liste modifiable** : cela signifie que le champ Robe apparaîtra dans les formulaires sous la forme d'une zone de liste modifiable (au lieu d'une zone de texte, qui est l'option par défaut),
2. pour la propriété *Origine source*, choisissez l'option **Liste valeurs** : cela signifie que la liste de valeurs proposées va être saisie à la main, à la ligne suivante, dans la propriété *Contenu*.
3. dans la propriété *Contenu*, saisissez les valeurs suivantes séparées par des point-virgules :
alezan ; bai ; bai-brun ; bai foncé ; gris ; rouan

Sauvegardez la table *Etalon*.

4.4.3 Manipulation : modification de la requête ReqFicheEtalon

Dans la fenêtre *Base de données*, sélectionnez l'onglet *Requêtes*, la requête *ReqFicheEtalon* puis cliquez sur le bouton *Modifier*.

Complétez la requête avec les attributs que vous avez ajoutés dans la table *Etalon*. Pour cela, sélectionnez les 4 nouvelles colonnes et glissez-les dans le tableau se trouvant dans la partie inférieure de la fenêtre.

Sauvegardez votre requête par l'option *Enregistrer* du menu.

Exécutez la requête en cliquant sur l'icône *Point d'exclamation* de la barre d'outil (*Exécuter* dans la bulle d'aide). Vous devez voir apparaître la liste de tous les couples Etalons/propriétaires de la base.

4.4.4 Manipulation : modification du formulaire FicheEtalon

Nous allons ajouter les nouvelles colonnes dans le formulaire.

Cliquez sur l'onglet **Formulaires** de la fenêtre **Base de données**. Sélectionnez le formulaire *FicheEtalon*. Cliquez sur le bouton **Modifier**.

Pour ajouter sur le formulaire les colonnes de la requête qui ne s'y trouvent pas déjà, cliquez sur le bouton **Liste des champs** de la barre d'outil. Elle fait apparaître la liste de toutes les colonnes de la requête associée au formulaire. Cliquez sur les colonnes à ajouter et glissez-les sur le formulaire.

Pour encadrer les caractéristiques de l'étalon, les tarifs de saillies et les informations sur le propriétaire, placez trois contrôles graphiques de type **Groupe d'options**.

Sauvegardez le formulaire et exécutez-le.

Remarquez l'effet, sur le formulaire, de la liste de choix que nous avons établie pour le champ Robe, dans la table Etalon, avec un affichage de type *Zone de liste modifiable*.

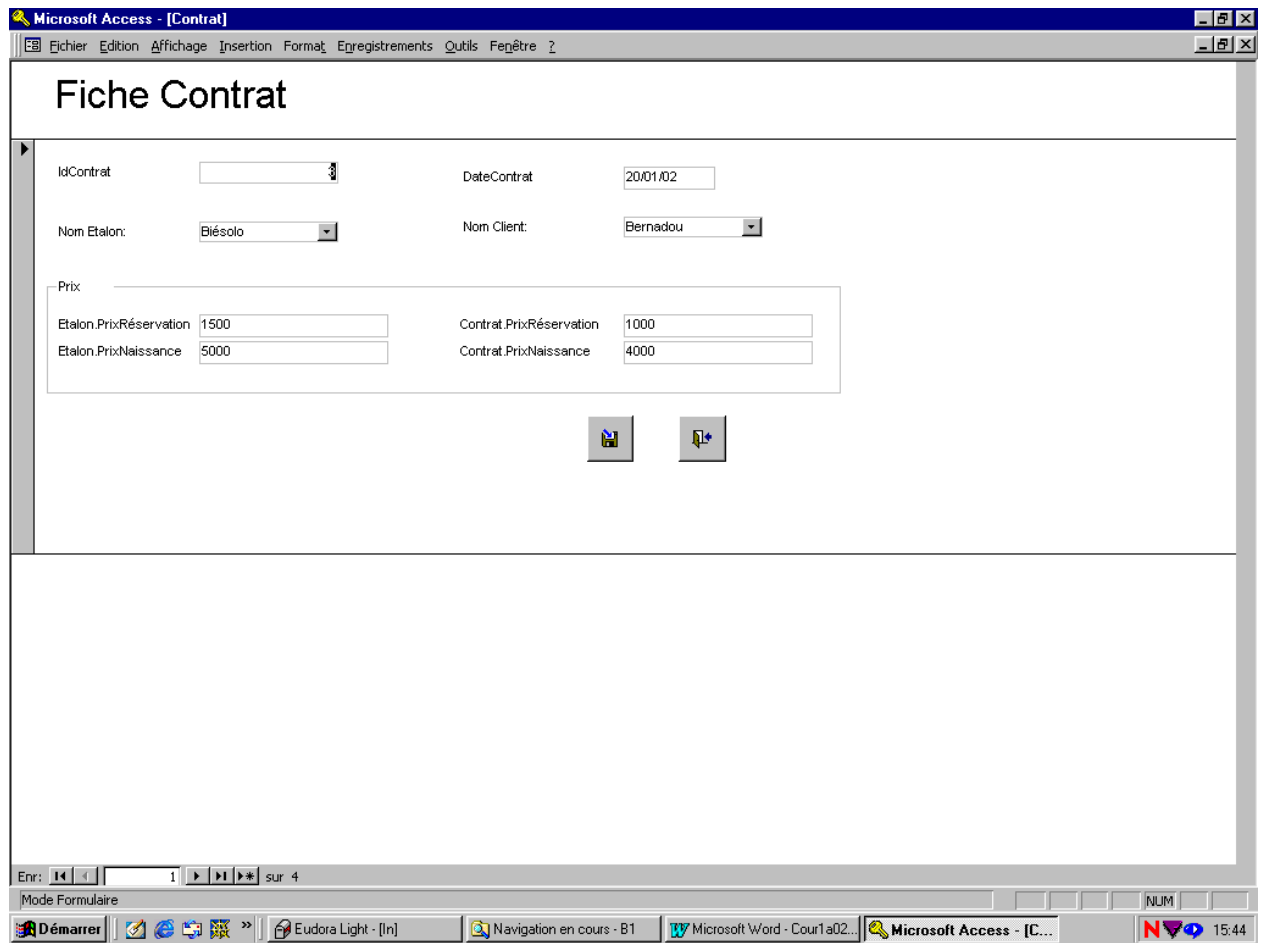
4.5 Verrouillage des contrôles zone de texte

Il est parfois intéressant d'empêcher la modification de certaines données affichées dans les contrôles du formulaire.

*Par exemple, dans le formulaire **FicheEtalon**, on désire empêcher la modification des informations concernant les propriétaires (nom, prénom). Seules les colonnes de la table Etalon peuvent être modifiables.*

Nous allons donc verrouiller les contrôles zone de texte associés aux colonnes de la table *Personne*. Sélectionnez ces contrôles (Shift-clic), ouvrez l'éditeur de propriétés, dans la liste des *propriétés des données*, positionnez la propriété **Verrouillé** à **oui**.

Pour différencier visuellement les zones verrouillées des zones de saisie, modifiez la couleur d'affichage du contrôle zone de texte verrouillé (**Propriété couleur du fond** à gris).



4.6 Exercices

4.6.1 Création du formulaire *FicheContrat*

Pour l'application de gestion du haras, vous allez créer une table *Contrat* qui contient les informations suivantes :

- *IdContrat* de type *NuméroAuto*,
- *DateContrat* de type *date*,
- *NoClient* de type numérique entier long,
- *NoEtalon* de type numérique entier long,
- *PrixRéservation* de type numérique réel double,
- *PrixNaissance* de type numérique réel double.

On veut pouvoir accéder dans le formulaire *FicheContrat* :

- aux informations stockées dans la table *Contrat*,
- aux prix catalogue à la réservation et la naissance se trouvant dans la table *Etalon*.

Dans ce formulaire, on dispose de deux listes modifiables qui permettent d'affecter :

- à la colonne *NoEtalon* de la table *Contrat* le numéro de l'étalon correspondant à l'étalon sélectionné dans la liste déroulante. La liste affiche tous les étalons de la base triés sur leur nom,
- à la colonne *NoClient* le numéro du client correspondant au client sélectionné dans la liste déroulante. La liste affiche tous les clients de la base triés sur leur nom et leur prénom.

Les prix catalogue ne sont accessibles qu'en lecture dans ce formulaire.

Les prix à la réservation et à la naissance du contrat peuvent avoir été négociés par rapport aux prix catalogue.

Données stockées dans la table Parcelle

The screenshot shows the Microsoft Access interface with a form titled 'Formulaire: Parcelle'. The form contains the following fields and data:

| Parcelle | | | |
|---------------------|--|----------|--------------------------------|
| NoParcelle: | <input type="text" value="2"/> | | |
| NomParcelle: | <input type="text" value="Aue"/> | | |
| Superficie: | <input type="text" value="100"/> | | |
| Caractéristiques | | | |
| Nitrate: | <input type="text" value="A"/> | Kalium: | <input type="text" value="A"/> |
| Phosphate: | <input type="text" value="B"/> | Calcium: | <input type="text" value="C"/> |
| Magnésium: | <input type="text" value="B"/> | Sulfure: | <input type="text" value="A"/> |
| Exploitation | | | |
| Nom: | <input type="text" value="Boldevitz"/> | | |
| BureauDistributeur: | <input type="text" value="Zurich"/> | | |

At the bottom of the form, there is a status bar showing 'Enr: 1 sur 5'. The Windows taskbar at the bottom shows the Start button, Explorer - Temp, Eudora, Microsoft Word, and Microsoft Access. The system tray shows the time as 18:17.

Données stockées dans la table Exploitation

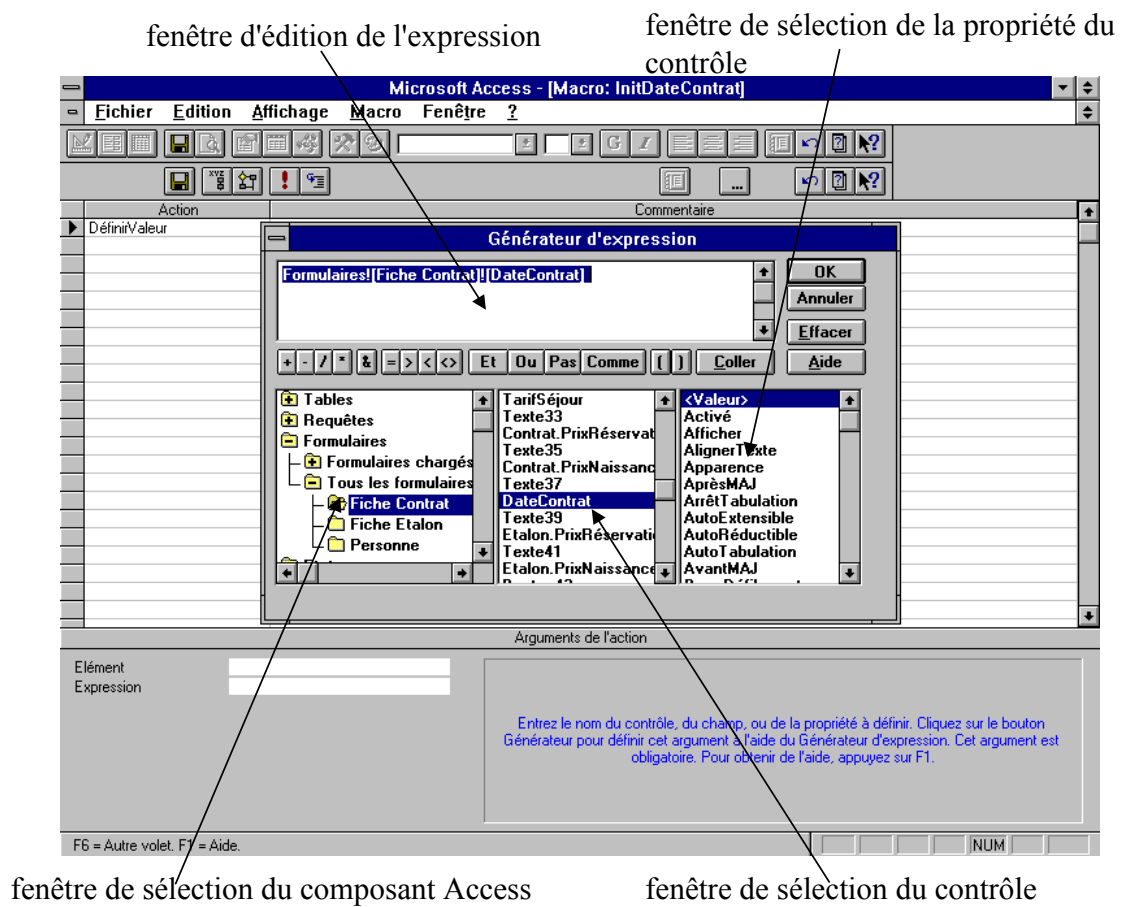
Liste modifiable pour sélection d'une exploitation

4.6.2 Création du formulaire Parcelle

Dans la base de données Gestion des exploitations, vous allez créer un formulaire permettant de gérer des parcelles.

Ce formulaire inclut les données de la table Parcelle ainsi que le nom de l'exploitation et le bureau distributeur dont elle dépend.

Ces deux dernières informations ne sont accessibles qu'en lecture dans le formulaire. Pour affecter la parcelle à une exploitation, on dispose d'une liste modifiable dans laquelle Access affiche la liste de toutes les exploitations de la base triées par nom et bureau distributeur.



4.7 Événements et macro-commandes

4.7.1 Exemple d'introduction

On aimerait améliorer l'ergonomie du formulaire *FicheContrat* de deux manières :

- lorsque l'on insère un nouveau contrat, la zone de texte *DateContrat* est remplie automatiquement avec la date du jour gérée par l'ordinateur,
- lorsque l'on sélectionne un étalon, les prix catalogue provenant de la table *Etalon* sont recopiés automatiquement dans les zones de texte associées aux prix négociés stockés dans la table *Contrat*.

Pour cela, nous allons utiliser les possibilités qu'offre Access de programmer des **actions automatiques** (ici définir la valeur de zones de texte) lorsqu'un **événement** intervient (ici insertion d'un nouvel enregistrement dans la table *Contrat* ou choix d'une valeur dans une liste modifiable par exemple). Nous allons apprendre à programmer un nouveau type de composant d'Access : la **macro-commande** (ou plus simplement la macro).

4.7.2 Définitions

Définition 1 : Une **macro-commande** est un ensemble d'actions paramétrées exécutées séquentiellement par Access lorsque l'on invoque cette macro.

Définition 2 : A chaque composant graphique du formulaire, est associé un ensemble prédéfini d'**événements** auxquels il est possible d'associer une macro-commande ou une fonction écrite en langage Visual Basic.

Par exemple, un bouton de commande peut réagir à l'événement surClic. La réaction se traduit par l'exécution de la macro-commande associée à cet événement.

4.7.3 Manipulation : insertion automatique de la date du contrat

Lorsque l'on programme une macro, on fait deux choses :

1. on définit l'ensemble des actions qu'elle doit réaliser automatiquement en utilisant le générateur de macro,
2. on définit sur quel événement elle doit être invoquée en utilisant l'éditeur de propriétés.

Ouvrez le formulaire *FicheContrat* en mode création. Editez les propriétés du contrôle formulaire (double-clic dans le carré gris au coin supérieur gauche du formulaire). Placez-vous sur l'onglet **événement**. Remarquez qu'à chaque fois que vous sélectionnez un événement, Access affiche un commentaire explicatif dans la barre d'état.

Sélectionnez l'événement **Avant Insertion**, nous allons lui associer une macro qui va initialiser la date du contrat lorsque l'on saisit un nouveau contrat.

Cliquez sur les trois petits points de suite en bout de ligne. Dans la boîte de dialogue **Choisir Générateur**, sélectionnez **Générateur de macro**.

Nommez la macro *InitDateContrat*.

Sur la première ligne de la macro, dans la colonne **Action**, sélectionnez l'action **Définir Valeur**.

Dans la partie *Arguments* de l'action (partie inférieure de la fenêtre), deux lignes sont à paramétrer :

1. la ligne *Elément* : on y indique le contrôle dans lequel on veut affecter une valeur,
2. la ligne *Expression* : on y indique la valeur que l'on veut affecter.

On va utiliser le générateur d'expression pour saisir ces deux arguments. Cet outil permet d'éviter les erreurs de syntaxe.

Saisie de l'argument *Elément* :

Cliquez sur les trois petits points de suite au bout de la ligne *Elément*.

Sous le générateur d'expression, dans la colonne inférieure gauche (sélection d'un composant Access), ouvrez le dossier *Formulaires*, sous-dossier *Tous les formulaires* et le formulaire *FicheContrat*.

Dans la deuxième colonne (sélection d'un contrôle), apparaissent tous les noms de contrôles graphiques du formulaire sélectionné. Double-cliquez sur *DateContrat*. Le nom complet du contrôle apparaît dans la fenêtre supérieure :

```
Formulaires![FicheContrat]![DateContrat]
```

Saisie de l'argument *Expression* :

Cliquez sur les trois petits points de suite au bout de la ligne *Expression*.

Sous le générateur d'expression, dans la colonne inférieure gauche (sélection d'un composant Access), ouvrez le dossier *Fonctions*, sous-dossier *Fonctions intégrées*.

Dans la deuxième colonne (sélection d'une catégorie de fonctions), cliquez sur *Date/Heure*. Dans la troisième colonne, double-cliquez sur *Date*. Le nom complet de la fonction apparaît dans la fenêtre supérieure :

```
Date()
```

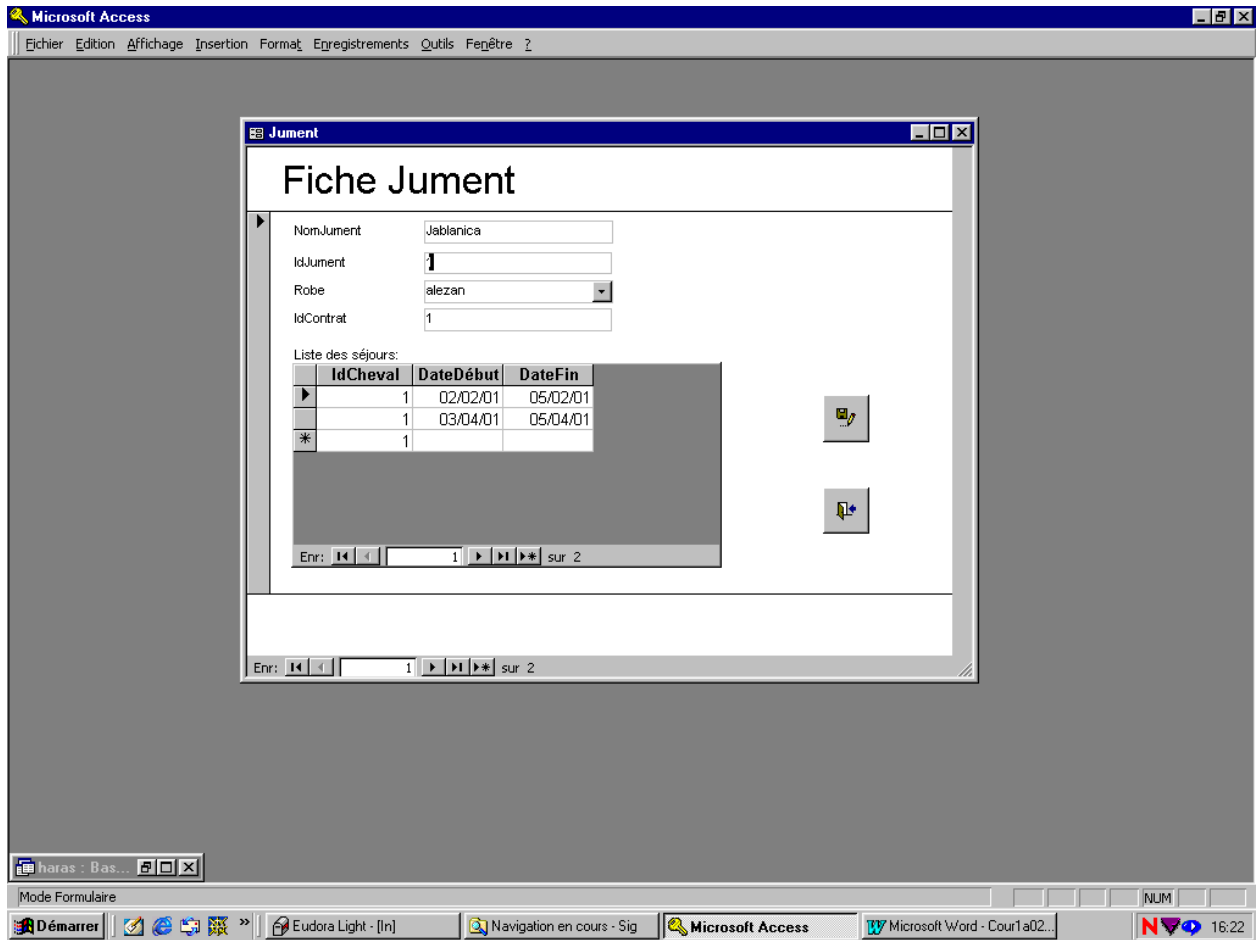
Sortez du générateur d'expression, sauvez la macro et testez le formulaire en exécution pour vérifier que la macro réalise bien le traitement désiré.

4.7.4 Manipulation : initialisation des prix négociés

Nous allons créer une macrocommande *InitPrixNégociés* qui, lorsque l'on sélectionne un étalon dans la liste modifiable du formulaire *FicheContrat*, recopie le prix à la réservation « catalogue » de l'étalon sélectionné dans les contrôles zone de texte correspondant au prix à la réservation « négociés » (celui de la table *Contrat*). Vérifiez que les contrôles zone de texte correspondant aux prix catalogue sont initialisés lorsque l'on sélectionne un nouvel étalon dans le contrôle de type liste modifiable. Vous allez donc associer la macro *InitPrixNégociés* à l'événement *Après MAJ* du contrôle de type liste modifiable.

4.7.5 Exercice

Complétez la macro *InitPrixNégociés* pour obtenir la recopie automatique du prix à la naissance catalogue dans le prix à la naissance négocié (table *Contrat*).



4.8 Contrôle sous-formulaire

4.8.1 Définition

Un **contrôle sous-formulaire** permet d'afficher dans un formulaire (appelé formulaire principal) un autre formulaire (appelé sous-formulaire) permettant d'afficher une source de données, notée B, **synchronisée** à la source de données, notée A, du formulaire principal. La source B est actualisée automatiquement à chaque fois que l'on change de fiche dans le formulaire principal, c'est à dire à chaque fois que l'on change de ligne dans la source de données A associée au formulaire principal.

Par exemple, dans le formulaire FicheJument, la source de données associée au formulaire est la table Jument. Dans le sous-formulaire, on affiche la liste des séjours qu'une jument a effectués au haras.

4.8.2 Relation entre les sources de données associées au formulaire et au sous-formulaire

Les sources de données associées respectivement à un formulaire et son sous-formulaire doivent avoir une ou plusieurs colonne(s) de liaison.

Définition : Il existe une relation de type un à plusieurs entre la source de données associée au formulaire principal et la source de données associée au sous-formulaire.

4.8.3 Exemple

Analysons les sources de données utilisées dans le formulaire **FicheJument**. On y affiche deux objets du système d'information :

- les juments représentées par la table **Jument**,
- les séjours représentés par la table **Séjour**.

La table *Jument* contient les informations caractérisant la jument :

- IdJument (type NuméroAuto),
- NomJument (type Texte),
- Robe (type Texte, avec la liste de choix suivante : **alezan ; bai ; bai-brun ; bai foncé ; gris ; rouan**),
- IdContrat (type Numérique, entier long).

La relation A Séjourné entre les tables Jument et Séjour

| Jument | |
|-----------|----------|
| Nom | IdJument |
| Jablanica | 2 |
| Cubanita | 8 |

Jablanica a séjourné deux fois au Haras. Le deuxième séjour n'est pas terminé.

| Séjour | | |
|----------|-----------|----------|
| IdCheval | DateDébut | DateFin |
| 2 | 2/4/1997 | 9/4/1997 |
| 2 | 4/5/1997 | |
| 8 | 2/2/1997 | 7/3/1997 |

Cubanita a effectué un seul séjour au Haras.

La table *Séjour* contient pour chaque jument les informations caractérisant chacun de ses séjours au haras, à savoir la date de début et de fin du séjour.

La table *Séjour* contient 3 colonnes :

- *IdCheval* (numérique entier long): cette colonne permet de faire la liaison avec la table Jument,
- *DateDébut* (type date) : date de début du séjour,
- *DateFin* (type date) : date de fin du séjour.

Quelle est la clé primaire de la table *Séjour* ?

Pourquoi n'enregistre-t-on pas directement les séjours dans la table Jument ?

Quel est le type de liaison entre les tables Jument et Séjour ?

4.8.4 Manipulations : création du formulaire FicheJument

Créez les tables *Jument* et *Séjour* telles que définies dans le paragraphe précédent. Etablissez un contrôle d'intégrité référentielle entre ces deux tables sur la colonne de liaison.

Créez avec l'option **mode création** le formulaire *FicheJument* ayant pour source de données la table *Jument*. Utilisez la liste des champs pour poser les colonnes de la table sur le formulaire. Ajoutez avec l'assistant deux boutons de commande pour enregistrer et fermer le formulaire.

Créez un deuxième formulaire en lui associant la table *Séjour*. Utilisez l'option **mode création** et nommez ce formulaire *SfFicheJument/Séjour*. Utilisez la liste des champs pour poser les colonnes de la table sur le formulaire. Configurez la propriété **Affich par Défaut** du formulaire *SfFicheJument/Séjour* à **Feuille données**.

Puis ouvrez en mode création le formulaire *FicheJument*. Placez **sans l'assistant** un contrôle sous-formulaire sur le formulaire *FicheJument* en utilisant la boîte à outils. Nommez ce contrôle *CtrlSfFicheJument/Séjour* (propriété **Nom** dans l'éditeur de propriétés).

Paramétrez le contrôle sous-formulaire de la manière suivante :

- Sélectionnez le formulaire *SfFicheJument/Séjour* dans la propriété **Objet Source** du contrôle.
- Tapez le nom des colonnes de liaison entre les deux sources de données (*Jument* et *Séjour*), à savoir *IdJument* dans la propriété **Champ Père** (colonne de liaison de la source de données du formulaire principal) et *IdCheval* dans la propriété **Champ Fils** (colonne de liaison de la source de données du sous-formulaire).

Testez le bon fonctionnement du contrôle en exécutant le formulaire *FicheJument*.

Microsoft Access - [Formulaire: FicheSuivi]

Echier Edition Affichage Enregistrements Fenêtre ?

Fiche Suivi Jument

IdJument:

NomJument:

Robe:

Liste des examens, soins et inséminations :

| IdCheval | DateExamen | NoChaleur | OvaireGauche | OvaireDroit | DiversOG | DiversOD | DiversUtérus | DiversCol |
|----------|------------|-----------|--------------|-------------|----------|----------|--------------|-----------|
| 1 | 21/04/98 | 1 | 0 | 0 | ras | ras | ras | ras |
| 1 | 25/04/98 | 1 | 20 | 0 | | | | |
| 1 | 29/04/98 | 1 | 35 | 0 | | | | |
| 1 | 05/05/98 | 1 | 40 | 0 | | | | |
| 1 | | 0 | 0 | 0 | | | | |

Enc: 1 sur 4

Mode Formulaire

NUM

Démarrer Explorateur - Bd1A Microsoft Word - Cour1a98... Microsoft Access - [F... 19:01

4.8.5 Conclusion sur la structuration d'un formulaire

Règle 1 : Lorsque l'on veut afficher dans un formulaire une ligne d'une table A **jointe à une seule** ligne d'une table B, alors la source de données du formulaire est une requête de sélection reliant les tables A et B.

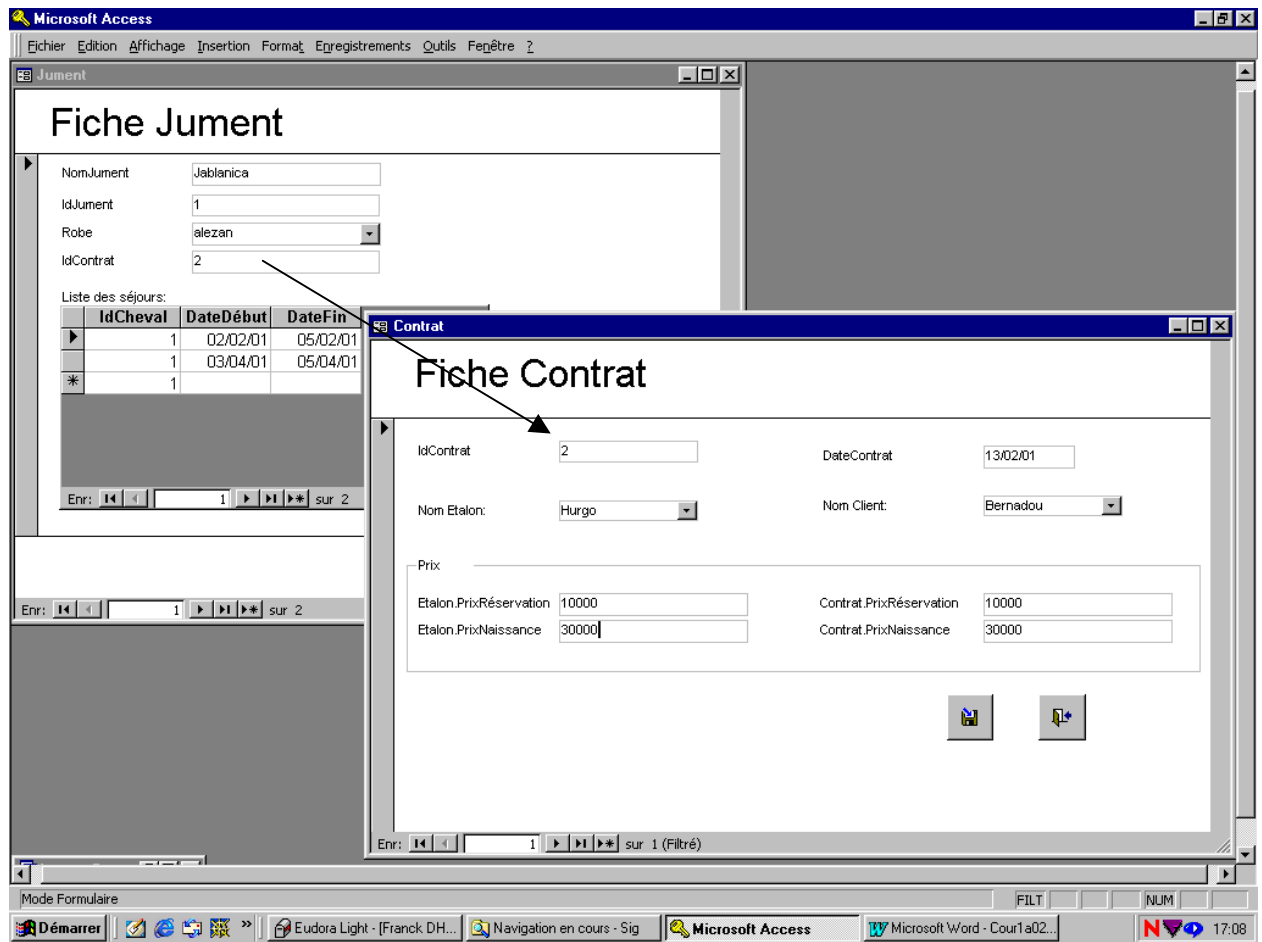
Exemple dans le poly : le formulaire FicheEtalon

Règle 2 : Lorsque l'on veut afficher dans un formulaire une ligne d'une table A **jointe à plusieurs lignes** d'une table B, alors : (1) le formulaire inclut un sous-formulaire, (2) la source de données du formulaire est la table A, (3) la source de données du sous-formulaire est la table B.

Exemple dans le poly : le formulaire FicheJument

4.8.6 Exercice : création de la fiche de suivi d'une jument

Vous allez créer un nouveau formulaire qui permet de gérer le suivi des juments. La source de données associée au formulaire *FicheSuivi* est la table *Jument*. Celle associée au sous-formulaire est une nouvelle table contenant le résultat des examens (Date de l'examen, no de chaleur de la jument, taille du follicule dans l'ovaire droit et dans l'ovaire gauche, etc ...). Vous appellerez cette table *SuiviJument*.



4.9 Navigation dans la base de données

L'ouverture d'un formulaire sur un enregistrement présélectionné dans sa source de données est une fonctionnalité très utile. Elle permet d'obtenir rapidement un complément d'information et plus généralement de naviguer d'une information à une autre. C'est une opération comparable à la navigation d'une page HTML à une autre par l'intermédiaire de liens hypertextes dans un navigateur WEB.

Par exemple, lorsque l'on visualise la fiche d'une jument, on veut avoir des précisions sur le contrat auquel elle émerge (nom de l'étalon, prix de vente, ...). Pour cela, on programme depuis le formulaire FicheJument l'ouverture automatique du formulaire FicheContrat positionné sur le contrat auquel émerge la jument.

4.9.1 Paramétrer l'action OuvrirFormulaire

Pour programmer cette fonctionnalité, il faut créer une macro et y sélectionner l'action **OuvrirFormulaire**. Puis il faut paramétrer deux arguments de cette action :

- le nom du formulaire à ouvrir (argument **Nom du formulaire**),
- l'enregistrement sur lequel il doit être ouvert (argument **Condition Where**).

Le deuxième argument doit être de la forme :

ColonneDestination = ContrôleSource

où

- ColonneDestination = Nom de la colonne de la source de données associée au formulaire à ouvrir sur laquelle on filtre l'enregistrement recherché,
- ContrôleSource = Nom du contrôle graphique (en général zone de texte ou liste modifiable) dans le formulaire d'appel contenant l'identifiant de l'enregistrement recherché.

4.9.2 Manipulation : Ouvrir le formulaire FicheContrat sur le contrat sélectionné

Nous allons programmer dans le formulaire *FicheJument* l'ouverture du formulaire *FicheContrat* sur le contrat sélectionné lorsque l'on double-clique sur le contrôle zone de texte *NoContrat*.

Ouvrez le formulaire *FicheJument* en mode création. Sélectionnez le contrôle zone de texte *IdContrat*. Ouvrez l'éditeur de propriétés.

Sur l'événement **Sur double-clic**, appelez le générateur de macros. Nommez la macro *OuvrirFicheContrat*. Sélectionnez l'action **OuvrirFormulaire**.

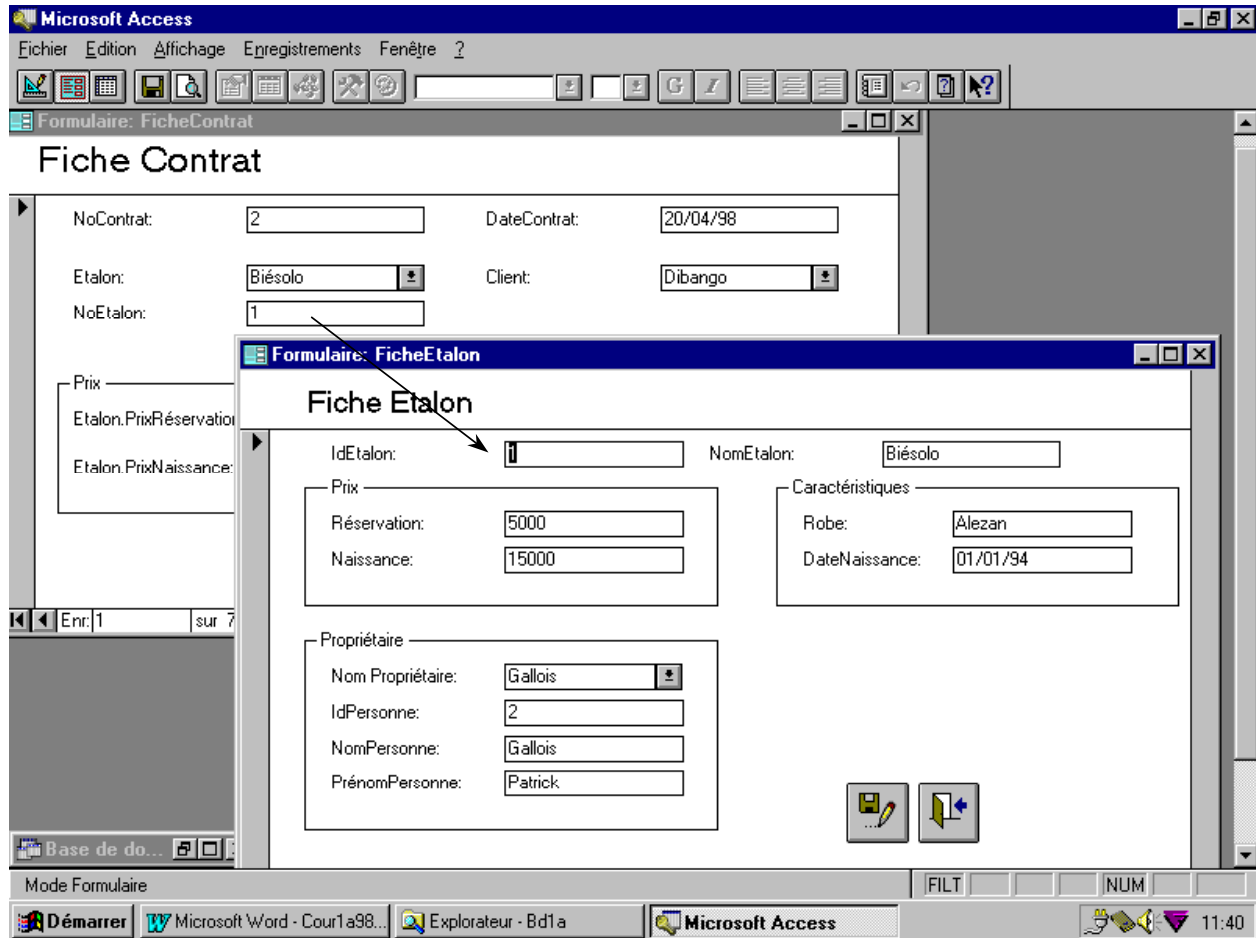
Dans l'argument **Nom Formulaire**, sélectionnez *FicheContrat*.

Dans l'argument **Condition Where**, tapez l'expression :

```
[ReqFicheContrat]![IdContrat]=[Formulaires]![FicheJument]![IdContrat]
```

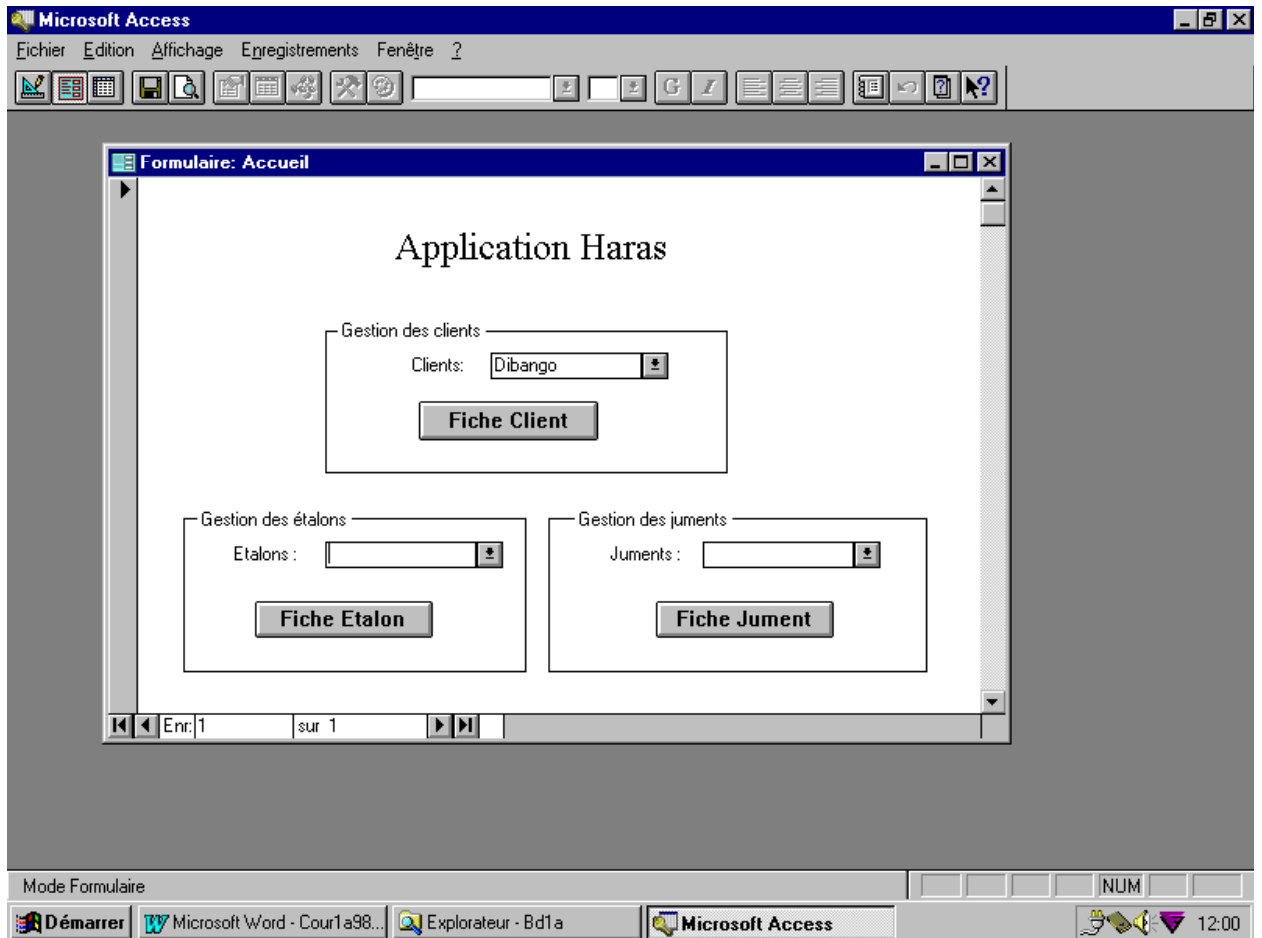
Rappel : On spécifie dans l'argument **Condition Where** sur quel enregistrement de la requête sous-jacente au formulaire *FicheContrat* on effectue l'ouverture du formulaire (partie gauche de l'affectation). La partie droite de l'affectation référence le contrôle qui contient le numéro de contrat sélectionné dans le formulaire appelant.

Testez l'ouverture du formulaire *FicheContrat* depuis le formulaire *FicheJument*.



4.9.3 Exercice : Ouverture de la fiche Etalon depuis la fiche Contrat

A partir de *la liste des champs*, ajoutez sur le formulaire *FicheContrat* un contrôle zone de texte affichant le numéro d'étalon. Ensuite, programmez l'ouverture automatique du formulaire *FicheEtalon* sur l'étalon sélectionné dans le formulaire *FicheContrat* lorsque l'on double-clique sur le numéro de l'étalon affiché.



4.10 Un formulaire libre comme formulaire d'accueil de l'application

Définition : Un formulaire libre est un formulaire sans *source de données*.

Nous allons nous servir de cette possibilité pour créer un formulaire d'accueil de l'application.

4.10.1 Manipulation

Le formulaire d'accueil est divisé en trois parties :

1. gestion des juments,
2. gestion des étalons,
3. gestion des clients.

Dans chaque partie, le principe est le même. On sélectionne dans la liste modifiable, l'occurrence d'entité par son nom (exemple, Dibango dans la liste modifiable des noms de clients). Puis on clique sur le bouton pour ouvrir la fiche sur l'occurrence d'entité sélectionnée.

Créez un nouveau formulaire (formulaire vierge) *FicheAccueil* sans lui associer de source de données.

Sur le formulaire, placez une liste modifiable, un bouton de commande et un cadre pour réaliser l'accès aux fiches client.

Affichez dans la liste déroulante associée à la liste modifiable la requête contenant la liste triée des noms de personnes. Pour cela, paramétrez les 4 propriétés *Contenu*, *Colonne liée*, *Nbre Colonnes*, *Largeurs de colonnes*.

Quelle colonne faut-il sélectionner en tant que colonne liée ?

Donnez un nom à la liste modifiable plus explicite que le nom par défaut donné par Access. Nommez-la par exemple, *ListeClients*.

Programmez une macro-commande que vous appellerez *OuvrirClient* qui ouvre le formulaire *FichePersonne* sur la personne sélectionnée dans la liste modifiable. Sélectionnez pour cela l'action ***OuvrirFormulaire*** et paramétrez les arguments ***Nom du formulaire*** et ***Condition Where*** de la manière suivante :

- ***Nom du formulaire***: FichePersonne
- ***Condition Where***:
[Personne]![IdPersonne]= Formulaires![FicheAccueil]![ListeClients]

4.10.2 Exercice

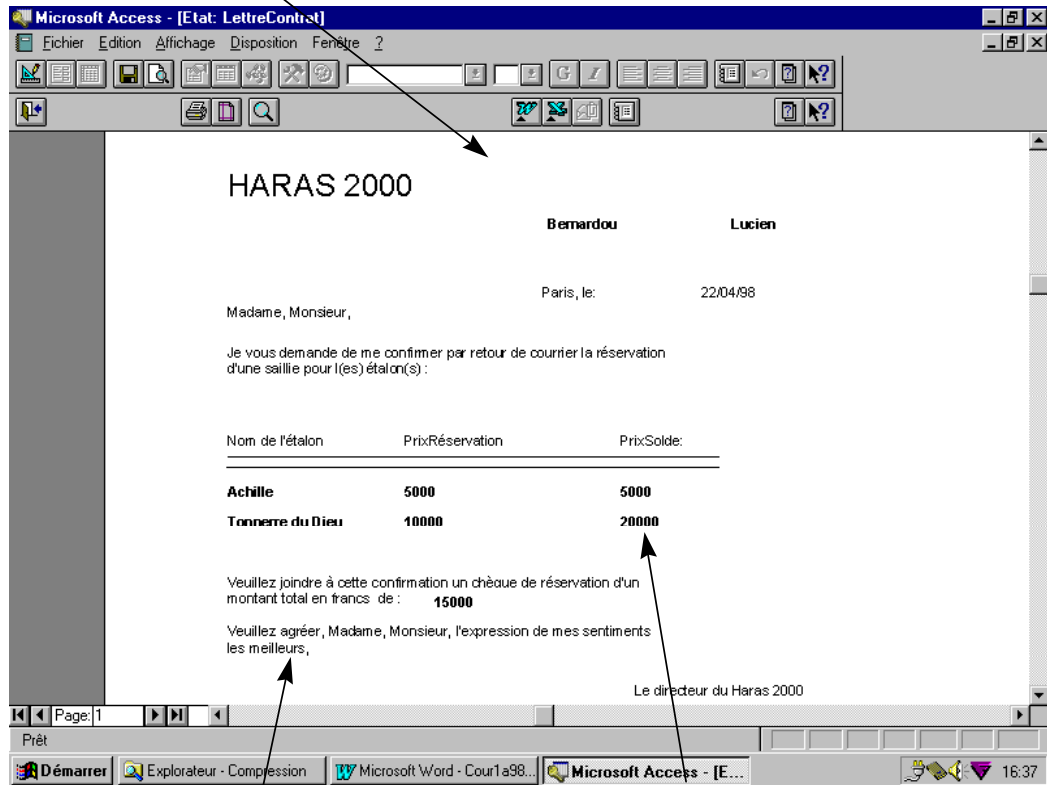
Programmez deux autres listes modifiables pour accéder depuis le formulaire *Accueil* aux fiches *FicheJument* et *FicheEtalon*

4.11 Questions de révision sur les formulaires

Répondez par oui ou non aux affirmations suivantes :

1. La source de données d'un formulaire est toujours une table ?
2. L'assistant Formulaire Instantané associe à chaque colonne de la source de données du formulaire un contrôle graphique (zone de texte, ...) ?
3. La liste des champs contient l'ensemble des colonnes de la source de données ?
4. On peut toujours modifier la valeur affichée dans un contrôle zone de texte ?
5. Dans la liste déroulante d'un contrôle liste modifiable, on peut afficher plusieurs colonnes ?
6. On peut afficher une liste de valeurs triées dans la liste déroulante d'un contrôle liste modifiable ?
7. La colonne liée d'un contrôle liste modifiable peut être n'importe laquelle des colonnes affichées dans la liste déroulante ?
8. Dans un contrôle liste modifiable, on peut affecter la colonne liée à l'une des colonnes de la source de données du formulaire ?
9. Dans une macro, on peut exécuter au maximum deux actions consécutives ?
10. On peut associer une macro à n'importe quel événement d'un contrôle graphique du formulaire ?
11. L'action DéfinirValeur permet de recopier une valeur dans un contrôle zone de texte ?
12. On peut visualiser un formulaire sous la forme d'un tableau de valeurs ?
13. Les arguments champs père et champs fils d'un contrôle sous-formulaire contiennent le nom des colonnes permettant de synchroniser les sources de données du formulaire et du sous-formulaire ?

L'état en mode aperçu avant impression



Texte fixe

Texte provenant de la source de données

5. Les états

5.1 Définition

Un composant Etat permet de mettre en forme des informations stockées dans la base de données en vue d'une édition.

Par exemple, on désire éditer une lettre de confirmation de réservation d'une ou plusieurs saillie(s) pour l'envoyer au client.

Comme pour les formulaires, on associe une source de données (une table ou une requête) à l'état.

Par exemple, une requête regroupant des informations provenant des tables Etalon, Personne et Contrat .

Des contrôles **zone de texte** sont associés aux colonnes de la source de données.

L'état est structuré au minimum en 5 parties :

1. l'entête d'état, imprimé une seule fois au début de l'état,
2. l'entête de page, imprimé sur chaque début de page,
3. le détail, imprimé pour chaque ligne de la source de données,
4. le pied de page, imprimé sur chaque fin de page,
5. le pied d'état, imprimé une seule fois à la fin de l'état.

On peut également opérer des regroupements de lignes de la source de données.

Par exemple, on désire regrouper dans une lettre à expédier à un client donné toutes les réservations de saillie d'étalon qu'il a effectuées.

Dans ce cas, Access crée deux nouvelles zones :

1. l'entête de groupe (après l'entête de page),
2. le pied de groupe (avant le pied de page).

On peut utiliser des fonctions de calcul sur les regroupements dans les contrôles *zone de texte*.

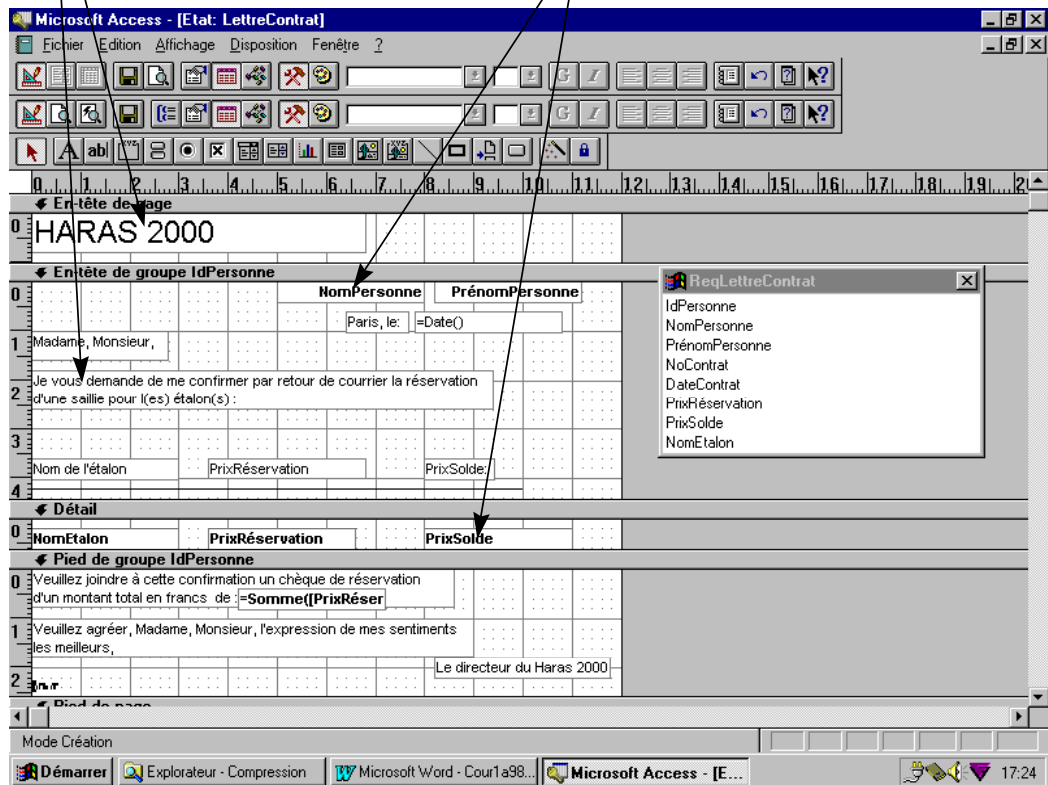
*Par exemple, on affiche dans le pied de groupe **IdPersonne**, le montant total à payer pour l'ensemble des saillies réservées :*

=Somme([PrixRéservation])

Dans cet exemple, la fonction **Somme** cumule les prix de réservation des lignes de la source de données correspondant au même client.

Contrôles étiquette

Contrôles zone de texte



L'état en mode création

5.2 Manipulation : la lettre de réservation

Créez une requête que vous nommerez *ReqLettreRéservation* qui regroupe :

- IdPersonne, NomPersonne, PrénomPersonne de la table Personne,
- IdContrat, DateContrat, PrixRéservation, PrixSolde de la table Contrat,
- NomEtalon de la table Etalon.

Depuis la fenêtre **Base de données**, cliquez sur l'onglet **Etats**. Cliquez sur le bouton **Nouveau**, sélectionnez la source de données *ReqLettreRéservation*. Sélectionnez le **mode création**. Cliquez sur OK.

Pour obtenir une zone de regroupement sur IdPersonne, sélectionnez dans le menu **Affichage** l'option **Trier/Regrouper**. Dans la fenêtre Trier/regrouper, sélectionnez la colonne IdPersonne. Paramétrez à oui les **options entête de groupe** et **pied de groupe**.

A partir de **la liste des champs**, placez les zones de texte contenant les colonnes de la source de données sur l'état comme sur le modèle de la page de gauche.

Tapez les textes dans des contrôles étiquette.

Dans un contrôle zone de texte que vous placerez dans le pied de groupe IdPersonne, tapez l'expression :

=Somme([PrixRéservation])

Testez votre état en mode aperçu avant impression.

5.3 Questions de révision sur les états

1. Chaque ligne de la source de données est affichée dans la zone Détail de l'état ?
2. Les colonnes de la source de données sont affichées dans des contrôles zone de texte ?
3. Le résultat de l'exécution d'un état peut être repris sous Word ?
4. La liste des champs contient la liste des colonnes de la source de données associée à l'état ?
5. On ne peut pas effectuer de cumul dans une zone de regroupement ?

6. Définition d'un menu pour l'application

Depuis la fenêtre *Base de données*, sélectionnez dans le menu *Affichage*, l'option *Barres d'outils, Personnaliser*.

Cliquez sur le bouton *Nouvelle*. Donnez un nom à la barre, par exemple *BarreHaras*. Cliquez ensuite sur le bouton *Propriétés*. Sélectionnez la barre d'outils *BarreHaras*. Sélectionnez **Barre de menu** dans l'attribut **Type**. Fermez la boîte **Propriétés**.

Pour saisir des options de menus dans cette barre, cliquez sur l'onglet **Commandes**. Sélectionnez la catégorie *Nouveau Menu*. Dans la fenêtre **Commande**, cliquez et glissez le texte *Nouveau Menu* sur votre barre de menu. Pour changer le texte de l'option de menu que vous venez de créer, placez-vous sur cette option dans la barre de menu et cliquez sur le bouton droit. Entrez le nom de l'option dans la propriété *Nom*. Pour créer un sous-menu, cliquez et glissez le texte *Nouveau Menu* sur l'option de menu à laquelle vous voulez l'associer.

Exercez-vous en créant une barre de menus que l'on associera au formulaire d'accueil de l'application incluant les menus suivant :

Fichier

Quitter

Edition

Couper

Copier

Coller

Juments

Liste des juments

Fiches jument

Fiches de suivi

Juments à voir

Etalons

Liste des étalons

Fiches étalons

Clients

Liste des clients

Fiches clients

Nouveau contrat

?

A propos de EquiManager

Pour associer une action à une option de menu, il faut d'abord créer une macro qui réalise l'action (par exemple, une action **OuvrirFormulaire**). Puis depuis la boîte **Personnaliser**, cliquez sur l'option de menu, cliquez sur le bouton droit et sélectionnez l'option **propriétés**. Sélectionnez la macro qui réalise l'action dans la propriété **SurAction**.

Pour associer cette barre de menu au formulaire principal *Accueil*, ouvrez le formulaire en mode création. Sélectionnez les propriétés du formulaire. Dans les **autres propriétés**, sélectionnez *BarreHaras* dans la propriété **Barre de menu**.

7. Démarrage automatique de l'application

7.1 La macro autoexec

Définition : La macro au nom prédéfini *autoexec* est exécutée automatiquement au démarrage d'Access.

Nous allons nous en servir pour ouvrir au lancement de l'application les formulaires d'accès aux entités de la base.

7.2 Manipulations

Sous l'onglet *Macros* de la fenêtre base de données, sélectionnez le bouton Nouveau. Sous l'éditeur de macro, sélectionnez l'action *OuvrirFormulaire*. Passez en argument le nom du formulaire d'accueil, à savoir *Accueil*. Puis, sélectionnez l'action *Agrandir*.

Sauvegardez la macro sous le nom *autoexec*. Quittez Access et relancez depuis l'explorateur. Les trois formulaires d'accès s'affichent automatiquement.

Table des matières

| | |
|---|-----------|
| 1. INTRODUCTION | 3 |
| 1.1 Introduction générale | 3 |
| 1.2 Système d'information | 6 |
| 1.3 L'exemple de la gestion d'un haras | 8 |
| 1.4 Utilité d'un système d'information automatisé | 10 |
| 1.5 Un système d'information enrichi par toute une équipe | 12 |
| 1.6 Prise en main de l'application Gestion de Haras | 14 |
| 1.7 Programmer une base de données Access | 16 |
| 1.8 Création d'une table | 18 |
| 1.8.1 Définitions | 18 |
| 1.8.2 Manipulations : Création d'une version simplifiée de la table Personne | 19 |
| 1.9 Création d'un formulaire | 22 |
| 1.9.1 Un formulaire, pour quoi faire ? | 22 |
| 1.9.2 Un formulaire, définitions | 22 |
| 1.9.3 Manipulations : création du formulaire Personne avec l'aide d'un assistant | 24 |
| 1.9.4 Manipulations : Personnalisation du formulaire FichePersonne | 25 |
| 1.9.5 Paramétrage des contrôles du formulaire | 26 |
| 1.9.6 Récapitulation sur la création d'un premier formulaire | 27 |
| 1.10 Pour en finir avec cette introduction | 30 |
| 1.11 Questions de révision sur l'introduction | 30 |
| 2. STRUCTURATION DES DONNEES | 32 |
| 2.1 Rappels sur les tables | 32 |
| 2.2 Objectif du chapitre | 32 |
| 2.3 Représentation des relations de type un à plusieurs entre entités | 33 |
| 2.3.1 Introduction | 33 |
| 2.3.2 Notion de relation un à plusieurs | 33 |
| 2.3.3 Représentation de la relation un à plusieurs | 33 |
| 2.3.4 Notion de clé étrangère | 34 |
| 2.3.5 Manipulations : l'exemple de la table Etalon | 34 |
| 2.4 Maintien de la cohérence des relations entre les tables | 35 |
| 2.4.1 Le problème de la cohérence | 35 |
| 2.4.2 Notion de contrôle d'intégrité référentielle | 35 |
| 2.4.3 Manipulations | 35 |
| 2.4.4 Maintien de la cohérence en cas de modification ou de suppression dans les tables référencées | 37 |

| | | |
|-------------|---|-----------|
| 2.5 | Représentation des relations de type plusieurs à plusieurs entre entités | 40 |
| 2.5.1 | Introduction | 40 |
| 2.5.2 | Notion de relation plusieurs à plusieurs | 40 |
| 2.5.3 | Représentation d'une relation plusieurs à plusieurs | 40 |
| 2.5.4 | Manipulations : création de la table EstLaPropriétéDe | 41 |
| 2.5.5 | Nouvelle définition d'une clé primaire | 42 |
| 2.6 | Récapitulation | 44 |
| 2.7 | Exercice : la gestion d'un ensemble d'exploitations agricoles | 45 |
| 2.8 | Questions de révision sur la structuration des données | 46 |
| 3. | LES REQUETES | 48 |
| 3.1 | Vues sur la base de données | 48 |
| 3.2 | Définition d'une requête de sélection | 50 |
| 3.3 | Manipulations : Création de la requête ReqFicheEtalon | 51 |
| 3.4 | Plusieurs types de jointures possibles entre les tables | 54 |
| 3.5 | Manipulations : paramétrer le type de jointure | 55 |
| 3.6 | Possibilités de sélection et d'interrogation multi-critères | 55 |
| 3.7 | Possibilités de tri | 56 |
| 3.8 | Possibilités de calcul | 56 |
| 3.9 | Possibilités de calcul sur regroupement de lignes | 56 |
| 3.10 | Manipulation | 57 |
| 3.11 | Exercices | 58 |
| 3.12 | Questions de révision sur les requêtes | 58 |
| 4. | FORMULAIRE | 60 |
| 4.1 | Rappels sur les formulaires | 60 |
| 4.2 | Source de données | 60 |
| 4.2.1 | Manipulations : création du formulaire FicheEtalon | 62 |
| 4.2.2 | Manipulations : accès à la source de données avec l'éditeur de propriétés | 63 |
| 4.3 | Contrôles listes modifiables | 66 |
| 4.3.1 | Définition | 66 |
| 4.3.2 | Un premier exemple d'utilisation du contrôle liste modifiable | 66 |
| 4.3.3 | Manipulations : création d'une liste modifiable | 67 |

| | | |
|-------------|--|------------|
| 4.4 | Impact de la modification de la structure de données sur l'application | 70 |
| 4.4.1 | Manipulation : modification de la structure de la table Etalon | 70 |
| 4.4.2 | Manipulation : un deuxième exemple d'utilisation du contrôle liste modifiable dans la définition d'une table | 71 |
| 4.4.3 | Manipulation : modification de la requête ReqFicheEtalon | 72 |
| 4.4.4 | Manipulation : modification du formulaire FicheEtalon | 73 |
| 4.5 | Verrouillage des contrôles zone de texte | 74 |
| 4.6 | Exercices | 76 |
| 4.6.1 | Création du formulaire FicheContrat | 76 |
| 4.6.2 | Création du formulaire Parcelle | 78 |
| 4.7 | Evénements et macro-commandes | 80 |
| 4.7.1 | Exemple d'introduction | 80 |
| 4.7.2 | Définitions | 80 |
| 4.7.3 | Manipulation : insertion automatique de la date du contrat | 81 |
| 4.7.4 | Manipulation : initialisation des prix négociés | 83 |
| 4.7.5 | Exercice | 83 |
| 4.8 | Contrôle sous-formulaire | 86 |
| 4.8.1 | Définition | 86 |
| 4.8.2 | Relation entre les sources de données associées au formulaire et au sous-formulaire | 86 |
| 4.8.3 | Exemple | 86 |
| 4.8.4 | Manipulations : création du formulaire FicheJument | 91 |
| 4.8.5 | Conclusion sur la structuration d'un formulaire | 94 |
| 4.8.6 | Exercice : création de la fiche de suivi d'une jument | 94 |
| 4.9 | Navigation dans la base de données | 96 |
| 4.9.1 | Paramétrer l'action OuvrirFormulaire | 96 |
| 4.9.2 | Manipulation : Ouvrir le formulaire FicheContrat sur le contrat sélectionné | 97 |
| 4.9.3 | Exercice : Ouverture de la fiche Etalon depuis la fiche Contrat | 100 |
| 4.10 | Un formulaire libre comme formulaire d'accueil de l'application | 102 |
| 4.10.1 | Manipulation | 102 |
| 4.10.2 | Exercice | 103 |
| 4.11 | Questions de révision sur les formulaires | 104 |
| 5. | LES ETATS | 106 |
| 5.1 | Définition | 106 |
| 5.2 | Manipulation : la lettre de réservation | 110 |
| 5.3 | Questions de révision sur les états | 111 |
| 6. | DEFINITION D'UN MENU POUR L'APPLICATION | 112 |
| 7. | DEMARRAGE AUTOMATIQUE DE L'APPLICATION | 114 |
| 7.1 | La macro autoexec | 114 |
| 7.2 | Manipulations | 114 |