

**I.U.P.**  
**Ingénierie Informatique**  
**AVIGNON**



**Institut National de la Recherche Agronomique**

UNITE DE SCIENCE DU SOL D'AVIGNON

UNITE DE BIOCLIMATOLOGIE D'AVIGNON

Base de données PORTOS, mise en forme de la base et  
création d'un interface utilisateur;

Sébastien MERCIER

Rapport de stage de deuxième année  
Juin-Septembre 1998

encadrants :  
A. CHANZY  
J.-P WIGNERON

INRA AVIGNON Science du Sol  
INRA AVIGNON Bioclimatologie

# PLAN GENERAL

INRA  
STATION DE SCIENCE DU SOL

- 7 SEP. 1999

Domaine Saint Paul - Site Agroparc  
84914 AVIGNON CEDEX 9

## Introduction :

### Chapitre I : Description de la manipulation Portos 93

- 1.1 L'unité de science du sol à l'INRA.
- 1.2 La manipulation Portos 93.
  - 1.2.1 Généralités sur la télédétection et le radar Portos .
  - 1.2.2 Portos 93 : cadre de la manipulation et mesures effectuées.

### Chapitre II : choix du système de gestion de base de données en fonction du cahier des charges.

- 2.1 cahier des charges et objectifs demandés pour la création de la base de données.
- 2.2 Le choix du système de gestion de base de données.
- 2.3 Access 97 : esprit du logiciel et méthodes globales choisies pour la construction de la base.
  - 2.3.1 Présentation de l'environnement.
  - 2.3.2 La programmation VBA.

### Chapitre III : Mise en œuvre de la base Portos 93.

- 3.1 Insertion des données et création des tables.
  - 3.1.1 Les différents types de données à gérer .
  - 3.1.2 Les tables : choix et mode de création.
  - 3.1.3 Méthodes d'insertion des données.
- 3.2 Requêtes utilisées
  - 3.2.1 Généralités sur les requêtes mises en œuvre dans la base.
  - 3.2.2 mode assistant de requête : un moyen facile et rapide mais...
  - 3.2.3 Le langage VBA pour l'implémentation des requêtes : un support incontournable.
- 3.3 L'interface utilisateur
  - 3.3.1 L'esprit « *formulaire* » Microsoft Access.
  - 3.3.2 La mise en forme que nous avons adoptée : présentation des menus.
- 3.4 Dernière étape : création d'un exécutable avec programme d'installation.
  - 3.4.1 Office Developer.
  - 3.4.2 Constatations
  - 3.4.3 Problèmes communs aux trois catégories de machines :

## Conclusion.

# INTRODUCTION:

Dans le cadre de ce stage de deuxième année, il m'est proposé la réalisation complète d'une base de données sur une expérimentation de l'Unité de Science Du Sol de L'INRA d'Avignon. Cette manipulation dénommée **PORTOS 93** s'est déroulée durant les mois de Mars à Juin de l'année 1993. Elle a pour objectif de mieux connaître le radiomètre Portos, appareil qui sera installé prochainement sur des satellites. L'encadrement est assuré par les chercheurs André Chanzy et Jean-Pierre Wigneron.

Après présentation de l'Unité de Science du Sol, nous cadrerons plus précisément les objectifs de cette expérimentation. Ensuite, nous développerons les différentes étapes qui ont conduit à la création du CD-ROM que vous trouverez avec le rapport. Je vous prie de bien vouloir excuser le retard dans la remise de mon compte rendu. Les améliorations successives apportées à la Base ont en effet retardé la production du CD-ROM final.

## CHAPITRE I:DESCRIPTION DE LA MANIPULATION.

### 1.1 L'unité science du sol à L'INRA.

Avant de parler de l'unité de Science du Sol à l'INRA d'Avignon, cadre de mon stage de deuxième année, nous allons donner quelques renseignements généraux sur l'INRA.

L'institut national de la recherche agronomique a été créé en 1946, alors dirigé par le ministère chargé de l'agriculture. En 1984, l'institut devient un établissement à caractère scientifique et technique (E.P.S.T) et se trouve placé sous la double tutelle du :

- Ministère chargé de la recherche et de la technologie
- Ministère chargé de l'agriculture et des forêts

Sa mission (décret du 14-12-84) consiste à :

- Organiser et exécuter toutes les recherches intéressant l'agriculture et les industries qui lui sont liées ;
- Contribuer à l'élaboration de la politique nationale de la recherche ;
- Publier et diffuser les résultats de ses travaux ;
- Apporter son concours à la formation.

L'institut dispose d'un budget d'environ 2.5 milliards de francs, dont 80% proviennent de l'état et 20% de ses propres ressources. L'effectif du personnel permanent est de 8500 agents, ce qui le place en troisième position des organismes de recherche français derrière le C.N.R.S et le Commissariat à l'Energie Atomique.

Le centre de Recherche Agronomique en Avignon est l'un des plus importants sites avec 500 personnes réparties sur plusieurs domaines dont celui de Saint Paul situé à côté de l'I.U.P. L'unité de Science du Sol qui m'accueille aujourd'hui appartient au domaine Saint Paul.

### Unité de Science du Sol

L'objectif général de cette unité de recherche est la compréhension des processus et des phénomènes physiques qui ont lieu dans le sol ou en rapport avec le sol.

Les principaux axes de recherches sont :

- l'influence de l'état et de la constitution des sols sur la circulation de l'eau, des solutés, des gaz et de la chaleur.
- Les mécanismes de contrôle de la qualité des eaux de surface et de profondeur, à l'échelle de la parcelle cultivée et du bassin versant.
- Le rôle des propriétés mécaniques du sol sur la croissance des racines

Du point de vue finalisé, les applications de ces travaux concernent les problèmes d'environnement, mais aussi l'optimisation des facteurs du milieu pour la croissance et le développement des couverts végétaux.

Les travaux sont effectués en équipe et sont répartis entre les analyses au laboratoire, les missions sur le terrain, et l'exploitation des résultats.

## 1.2 La manipulation Portos 93.

L'objectif de ce stage est la réalisation d'une base de données centrée sur une manipulation dénommée **Portos93**. Cette expérimentation s'est déroulée sur le site même du domaine Saint Paul d'avril à juillet 1993. Son objectif est de mieux connaître l'influence de certains paramètres de surface sur la mesure effectuée par le **radiomètre Portos**. Des données de différentes natures, essentiellement de type numériques ou photographiques ont été collectées par les chercheurs André Chanzy et Jean Pierre Wigneron qui ont suscité le stage et assuré l'encadrement. Un travail de regroupement et de mise en forme s'est alors avéré nécessaire devant l'existence de données en grand nombre (environ 1700000 mesures) se trouvant dispersées entre ces deux chercheurs et sous des formats plutôt variés. Cependant, avant de nous expliquer plus clairement sur les objectifs de ce stage et sur la création de cette base, intéressons nous brièvement à la manipulation elle-même.

### 1.2.1 :Généralités sur la télédétection.

L'expérimentation Portos93 a pour objectif de mieux cerner le lien entre les mesures de télédétection dans le domaine des micro-ondes et les paramètres géophysiques de surface. La télédétection est une technique scientifique qui est utilisée entre autre pour l'observation, l'analyse, l'interprétation et la gestion de l'environnement à partir de mesures et d'images obtenues à l'aide de plates-formes aéroportées, spatiales, terrestres ou maritimes. Comme son nom l'indique, elle suppose l'acquisition d'information à distance, sans contact direct avec l'objet détecté. Sa définition officielle est « l'ensemble des connaissances et techniques utilisées pour déterminer des caractéristiques physiques et biologiques d'objets par des mesures effectuées à distance, sans contact matériel avec ceux-ci » (COMITAS, 1988).

La télédétection moderne est née de la photographie aérienne, dont la vue générale et verticale a considérablement développé nos possibilités d'inventaire, de cartographie et de surveillance de l'environnement et des ressources depuis plus d'un siècle. Elle intègre les développements plus récents de la recherche spatiale, de la physique et de l'informatique pour constituer aujourd'hui un outil des plus puissants et des plus flexibles pour la gestion du milieu, la planification et le développement économique. Comme elle fait appel à des techniques en évolution rapide dans un monde en changement, ses possibilités d'applications sont en croissance constante et l'on découvre de plus en plus de domaines où elle peut avantageusement remplacer des méthodes traditionnelles d'investigation, longues, coûteuses, fastidieuses. Elle apporte aussi une information nouvelle, différente et qui présente l'avantage d'être localisable dans l'espace. En particulier, la télédétection est la principale source

d'information pour les études de la transformation du globe dans le cadre des programmes internationaux de recherche sur les changements planétaires (climat, évolution du taux de CO<sub>2</sub>, météorologie), et à ce titre, on peut considérer qu'elle intervient directement dans la conscience écologique de nos sociétés.

La télédétection dans le domaine des micro-ondes, ou hyperfréquences (1 à 100 GHz), peut être envisagée selon deux principes différents. On distingue habituellement les techniques « passives », appelées ainsi parce que l'on n'enregistre que l'émission naturelle des surfaces visées, et les techniques « actives » où l'on « illumine » artificiellement la surface avec un radar pour interpréter le signal rétrodiffusé (voir plus loin).

L'émission micro-onde des surfaces naturelles est aussi plus complexe que celle qui se produit dans le domaine thermique. L'émission micro-onde est fonction de la température et de l'émissivité. L'émissivité d'une surface dépend de la constante diélectrique du milieu, de la géométrie de la surface ainsi que de la fréquence de l'onde et de l'angle de visée du radiomètre. Des contraintes matérielles d'instrumentation empêchent également de balayer le spectre d'une manière continue car il faut une antenne spécifique pour chaque bande de fréquence. Toutefois, le potentiel d'application de la radiométrie en hyperfréquences est très important pour **l'humidité des sols**, l'étude de la neige et des glaces, l'océanographie, ainsi que pour la végétation.

#### 1.2.2 Portos 93 : cadre de la manipulation et mesures effectuées.

La compréhension du cadre de la manipulation reste une étape importante dans la perspective d'une modélisation à effectuer pour développer une base de donnée.

La campagne de mesure Portos93 s'est déroulée sur le site de Poirson du domaine Saint Paul de l'I.N.R.A. de Montfavet sur une période de 3 mois, du 19 avril 1993 (jour julien 109) au 15 juillet 1993 (jour julien 205). Plus précisément, la finalité de l'expérience est l'estimation et le suivi des paramètres influençant simultanément ou séparément le signal micro-onde à travers l'analyse des possibilités offertes par la multiplicité des configurations du radiomètre PORTOS (fréquence, polarisation et angle de visée : nous retrouverons l'importance de ces paramètres dans l'implémentation de la base de données). Les **mesures radiométriques** réalisées ont été effectuées sur des parcelles de sol nu et des parcelles de végétation (cultures). Ces mesures se traduisent sous forme de températures de brillance en Kelvin. Plus précisément, cette température représente une énergie émise par le sol sous forme radiative dans le domaine de longueur d'onde considérée. C'est en fait l'énergie dégagée par un corps noir chauffé à cette température. Une mesure radiométrique est définie par sa **fréquence**, sa **polarisation** et son **incidence**. L'ensemble de ces paramètres constituent la configuration qu'on a fait varier dans le cadre de l'expérimentation Portos pour isoler les paramètres de surface qui influencent la mesure radiométrique.

#### **Description du site de mesures**

Le site est constitué d'un ensemble de 7 parcelles, couvrant une surface totale de 1 hectare. Deux rails traversent longitudinalement le site permettant le déplacement d'une grue de chantier (hauteur 20m), sur laquelle sont disposés les radiomètres. En jouant sur les possibilités de déplacement de la grue (déplacement de la tour, rotation de la flèche,

déplacement du chariot), on assure une couverture du site et la possibilité de « voir » les parcelles selon différents angles d'incidence. Une « thématique » bien précise a été attribuée à chaque parcelle. Cette « thématique » reste importante pour définir l'objet parcelle (" field" dans la base). Elles seront tout d'abord référencées par rapport à leur couverture végétale : sol nu, blé, sorgho homogène ou hétérogène. Ensuite, pour les parcelles nues, les états de surface des parcelles évoluant au cours de l'expérience (le sol est travaillé avec des engins agricoles pour obtenir des états de rugosité différents), les parcelles seront référencées en fonction du type de rugosité. En résumé, deux paramètres distinguent les parcelles :

- **Couverture végétale** : blé, sorgho ou nue.
- **Rugosité** pour les parcelles nues.

## Chapitre II : choix du système de gestion de base de données en fonction du cahier des charges.

### 2.1 cahier des charges et objectifs demandés pour la création de la base de données.

Le cahier des charges demandé pour la réalisation de la base de données Portos93 a été fixé par les chercheurs Jean-Pierre Wigner et André Chanzy (maître du stage). Ils ont plus particulièrement insisté sur les points suivants :

- La base devra **regrouper** et **uniformiser** toutes les données réparties entre les deux chercheurs. L'un et l'autre ont traité les données de manières différentes, que ce soit sous tableur ou sous le logiciel de mathématique « Splus ». Un stagiaire travaillant à l'époque sur la manipulation a également participé au traitement de ces données.
- Tous les paramètres de la manipulation devront être **extraits** de la base suivant des critères minimums tels que le numéro de la parcelle et une période entrés par l'utilisateur. L'extraction doit se présenter de deux manières :
  - Soit pouvoir être **imprimer**.
  - Soit pouvoir être **exporter** vers un fichier texte ne contenant que des données en colonne.
- Le logiciel devra permettre la création d'une recherche « à la carte » centrée sur la mesure de température de brillance (mesure centrale de la base de données). Pour chaque mesure de température de brillance, il sera bon de retrouver d'autres types de paramètres ayant été mesurés le même jour et se trouvant dans des tables différentes (ces paramètres étant recherchés suivant demande de l'utilisateur). C'est en somme une **analyse croisée**. Certaines mesures pourront être **interpolées**.
- L'**interface** utilisateur devra être simple, **compréhensible** et en **anglais**. Elle ne devra aborder aucune connaissance informatique majeure. La base s'accompagnera d'une notice explicative en anglais.

- Dans sa version finale, la base de données devra être distribuable sur support **CD-ROM**. L'exécutable présent sur le CD devra fonctionner sur **PC**. L'utilisateur ne possédera pas forcément le système de gestion de base de données. Il faudra tenir rigoureusement compte des problèmes de **copyright** qui pourraient survenir.

Sans être essentiel, il serait bon que la base de données contiennent également des publications et des photos concernant la manipulation. Elle s'accompagnera éventuellement de graphiques.

## 2.2 Le choix du système de gestion de base de données.

En premier lieu, notre choix dépendait de toute évidence des systèmes de gestion de base de données présents à l'I.N.R.A. . Quels sont-ils ?

- **Oracle version 7** utilisable sur n'importe quelle station.
- **Microsoft Access97** avec une licence PC pour l'unité Science du Sol.

Plusieurs solutions s'offraient à nous. Rappelons tout de même que cette base a pour objectif d'être distribuée facilement et que les données doivent en être exportable.

- 1) Une première solution consisterait à développer la base sous Oracle. La consultation se ferait via Internet sur un site géré par l'I.N.R.A. C'est une solution propre mais difficile à mettre en place (gestion de l'accession multi-utilisateur, gestion du site après mon départ,...).
- 2) Une deuxième solution serait de construire les tables sous Oracle, puis de gérer l'interface avec Access97. Un CD-ROM serait envoyé aux utilisateurs et Access se chargerait de la gestion des liens avec les tables présentes sur un site de l'I.N.R.A. Cette solution présente les mêmes problèmes que précédemment.
- 3) Une dernière solution serait de construire totalement la base sous Access . Ensuite, l'utilisation du logiciel « Office Developer » permettrait de créer un programme totalement autonome (installation et création d'un exécutable). Un CD-ROM contenant interface et données serait envoyé aux utilisateurs le demandant. Les problèmes de copyright seraient évité par l'envoi d'une version Access distribuable ne permettant que la lecture des fichiers "\*.mdb."

Nous avons opté pour la dernière solution. Plusieurs raisons ont motivées notre choix. Tout d'abord, un **exécutable PC sur CD-ROM** paraît être le vecteur de distribution le plus facilement intégrable par les futurs utilisateurs. Il suffira d'un PC environnement Windows95, NT ou 98 et d'un lecteur CD, outils minimum utilisés par les chercheurs à l'heure actuelle.

Ensuite, cette solution évite les problèmes de maintenance qui découleraient de la gestion d'un site sur station après mon départ. L'unité de Science Du Sol ne peut s'en occuper pour le moment. Tout doit être en place pour une distribution avant que je sois parti.

Enfin, la base Portos93 ne présente pas une proportion de données hors norme nécessitant impérativement l'utilisation d'un S.G.B.D puissant. Après évaluation du nombre de donnée, Access semble tout à fait suffisant. Nous avons estimé à environ 1000000 le nombre de chiffres à insérer dans la base. Nous verrons que nous nous sommes légèrement trompé : 1700000 données auront été rentrées à la fin du stage.

## 2.3 Access 97 : esprit du logiciel et méthodes globales choisies pour la construction de la base.

### 2.3.1 Présentation de l'environnement.

Microsoft Access est un système de gestion de base de données relationnel combinant les atouts de Windows95 , à savoir la clarté et l'aisance de certaines manipulations. Toute nouvelle base amorcée sous Access97 se présente sous la forme de plusieurs onglets dénommés : Table, Requête, Formulaire, Macro, Module.

- **Tables** : Les tables se présentent sous forme de tableaux deux dimension à la manière d'Excel. Leur création est relativement aisée. Une attention plus particulière doit être apportée sur les propriétés des champs (type, valeur par défaut, clé primaire, validité ...). Un assistant « création de table » est également disponible pour une création sans effort (nous verrons les immenses limites de ce système).
- **Requêtes** : Une méthode « visuelle » est disponible pour la création de requêtes. Elle se présente sous la forme d'un tableau où l'on peut sélectionner les différents champs piochés dans différentes tables. Des regroupements et des analyses croisées sont possibles. Un assistant est également disponible.
- **Formulaires** : Les formulaires font office d'interface entre la base et les utilisateurs. Ils permettent de créer des présentations personnalisées et d'afficher des résultats divers. Ils sont simples d'emploi et offrent des possibilités de présentation vraiment importantes. Les formulaires sont reliés au moteur de la base par le biais d'événements que provoquent l'utilisateur (double-cliquer par exemple ) et qui appellent des programmes associés.
- **Macros** : Les Macros Access offrent une programmation très « visuelle ». En effet, de petits programme peuvent être créés sans produire une ligne de code. Des menus déroulant contenant des actions diverses, en français, permettent une programmation aisée. Cependant, ces Macros se limitent à des routines vraiment simples (par exemple ouvrir un formulaire après click sur un bouton poussoir). Elles ne peuvent en aucun cas répondre à des besoins plus approfondis.
- **Modules** : Les modules sont des feuilles contenant du code VBA (Visual Basic Access : voir plus loin). C'est ici que va réellement se créer le cerveau de la base de donnée. L'environnement de programmation est agréable. (le compilateur reconnaît par exemple les différents types d'objet lors de la saisie et propose « en direct » les propriétés ou fonctions associées à cet objet. Les options de « débogage » sont très classiques et pratiques. Soulignons notamment la clarté de la fenêtre débogage qui affiche toutes les variables du programme sous forme de menus déroulants. Quelle différence avec la « watch » des compilateurs C.



Sur ce logiciel, Microsoft base sa publicité sur le fait qu'il est possible de créer une base de données sans utiliser une ligne de programmation. Des assistants, outils de fabrication automatique sans connaissances informatiques particulières, sont disponibles pour la création de tables, de requêtes et de formulaires. Les requêtes sont également réalisables par le biais de tableaux plus ou moins clairs et la programmation des Macros ne nécessite pas la connaissance d'un langage particulier. Cependant, ce mode de création de la base limite considérablement les possibilités globales de développement. Par exemple, dans le cahier des charges, il m'est demandé de mettre en place une extraction de paramètres divers choisis à la carte et centrés sur les mesures de températures de brillance. Cette opération complexe demandant plusieurs requêtes ne peut se faire à l'aide des fonctionnalités proposées par le logiciel.

Inévitablement, la réalisation d'une application plus soutenue et personnalisée nécessitent inévitablement l'utilisation d'une programmation « en dure » utilisant le langage fourni (ici Visual Basic Access).

### 2.3.2 La programmation VBA.

Ce nouveau langage sous Access97 remplace Access Basic qui était utilisé dans les versions 2 et antérieures. Ce langage est également utilisé pour la programmation de Microsoft Word, Microsoft Excel et Microsoft Project, et se retrouve dans Microsoft Visual Basic. Il a une structure orientée **objet**. Tous les objets de Visual Basic sont présents et de nouvelles classes particulières à la réalisation d'une base de données ont été rajoutées. Nous allons décrire les classes les plus importantes en nous appuyant sur des exemples de programmes utilisés pour la réalisation de Portos93 (voir Annexe ).

#### ➤ **Class Database** : objet base de données

Cette classe définit un pointeur sur une base de donnée Access quelconque. Le plus souvent, c'est la base courante sur laquelle on travaille.

Ex : Dim mabase as Database

*(définition du pointeur "mabase" de type base de donnée)*

Set mabase = currentDb

*(on le fait pointer sur la base en cour)*

*("mabase" peut également pointer sur autre base. )*

Set mabase = "c:\nom\_base.mdb"

*("mabase" pointe sur une base se trouvant sous la racine du lecteur c.)*

#### ➤ **Class Querydef** : objet requête

Cette classe permet de créer une requête dans la base choisie ou bien de placer un pointeur sur une requête temporaire en mémoire. C'est un des pointeurs que j'ai le plus utilisé. Il m'a permis de paramétrer des requêtes temporaires définies en langage SQL avec comme paramètres des valeurs entrées par l'utilisateur.

Ex : Dim req as Querydef

*(définition du pointeur req de type requête)*

Set req= mabase.createQuerydef()

*("req" est une requête temporaire de la base pointée par "mabase".)*

➤ **Class TableDef** : objet table

Un pointeur de type TableDef se référera à une table. Des opérations diverses sont réalisables tel que ajout ou suppression de champs, modification de propriétés,...

Ex : Dim table as TableDef  
Set table=mabase.createTableDef("toto")

➤ **Class Form**: objet formulaire

Un pointeur de type Form se référera à un formulaire. On pourra par la suite modifier les propriétés de celui-ci et lui ajouter des contrôles (zone de saisie ou d'affichage d'informations).

➤ **Class recordset**: objet enregistrement

Une variable de type "recordset" pointera sur le premier enregistrement provenant d'une table ou d'une requête. Il sera ensuite facile de passer en revue tous les enregistrements par l'intermédiaire d'une boucle pour modifications de certains champs ou insertions de nouveaux enregistrements.

Ex : Dim rec as Recordset *(déclaration du pointeur.)*

➤ **Class DoCmd**: objet commande système

C'est une classe regroupant des fonctions de gestion du système. C'est par elle que nous avons géré les ouvertures de formulaires lors d'événements particuliers. Par exemple, si une requête ne trouve aucun enregistrement, un formulaire donnant un message négatif sera affiché. Dans le cas contraire, un formulaire de visualisation des données sera affiché.

Ex : If res.EOF Then DoCmd.OpenForm ("pas\_dinfos")  
*(s'il n'y a pas d'enregistrements, ouvrir le formulaire "pas\_dinfos")*

Un exemple plus complet de programme illustrant l'utilisation de ces objets est fourni en **Annexe 1**.

Les procédures et les variables peuvent être déclarées de type **Public** dans les modules de formulaire et sont donc disponibles dans d'autres modules. On peut également créer de nouvelles classes avec le mot réservé **New**. L'utilisation d'un autre mot réservé **Optional** vous permet de déclarer une procédure à l'aide d'arguments facultatifs. De même, vous pouvez employer le mot **ParamArray** pour utiliser un nombre variable d'arguments.

Un explorateur objet bien conçu permet de naviguer entre toutes les classes pour en connaître leurs propriétés. Il constitue un outil de référence complet et facile à utiliser pour tous les objets, fonctions, constantes, méthodes et propriétés disponibles (du code est donné

en exemple). Sa consultation est indispensable, la plupart des documentation ne pouvant traiter de tous les objets.

Le langage V.B.A amène donc une grande souplesse de programmation sous Access97. L'environnement de programmation apporte un confort total, que se soit dans la correction automatique de la syntaxe ou encore dans la facilité de débogage. Les cours de programmation objet vus en deuxième année ont grandement contribué à une compréhension rapide de celui-ci.

## Chapitre III : Mise en œuvre de la base Portos 93.

### 3.1 Insertion des données et création des tables.

#### 3.1.1 Les différents types de données à gérer .

Les données de la manipulation Portos93 sont exclusivement de type numérique, à l'exception des photos et des publications. Elles se composent de données physiques et climatiques. Citons les principales :

- Températures de brillance (en Kelvin) : C'est la **donnée centrale** de la base. Elle est acquise certains jours par le radiomètre en fonction de protocoles mis en place par les chercheurs. Chaque donnée est fonction de certains paramètres de réglage du radiomètre : Fréquence, polarisation, azimuth et angle de visé. Chaque enregistrement contiendra ces paramètres, auxquels s'ajouteront des valeurs correctives . **Les jours et l'heures** correspondant à chaque mesure seront pris comme référence pour les recherches d'autres paramètres venant de tables différentes.
- Humidités : Des prélèvements de terre sont effectués sur le terrain puis pesés. Ils sont de nouveau pesés après séchage sous étuve. La différence de masse nous donne la quantité d'eau présente. Ces prélèvement sont effectués une fois par jour sur chaque parcelle à des profondeurs variées (0-0.5cm, 0-1cm, 1-2cm, 2-3cm, 3-4cm, 4-5cm, 5-7cm, 7-10cm).
- Températures : Des mesures de températures ont été faites en grand nombre (environ 880000 mesures). Elles sont effectuées automatiquement toutes les 15 minutes à des profondeurs de sol variées.
- Végétation : Cette catégorie de paramètres se présente sous la forme de mesures de biomasse des végétaux présents sur la parcelle. Il peut s'agir de mesures sur les tiges, les feuilles, les épis... effectuées sur une journée.
- Données climatiques : Ce sont des paramètres météorologiques classiques enregistrés sur le site de mesures. Il peut s'agir de données horaires ou journalières (températures, force du vent , évaporation, rayonnement solaire, couverture nuageuse...).

- Densités, rugosités : Mesures de densités et de rugosité (estimation du caractère plat ou vallonné d'une parcelle) effectuées une seule fois sur toutes les parcelles. Ces paramètres sont considérés stable au cours du temps.
- Textures : Elle se compose de 4 paramètres représentant la composition du sol :
  - Argile
  - Limon fin
  - Limon grossier
  - Sable
- Photos : Des photos prises du sommet de la grue où se situait le radiomètre doivent donner une idée sur la couverture végétale de la parcelle à une date donnée.

Toutes ces données seront archivées suivant deux notions importantes :

- Le **numéro** de la parcelle où la mesure a été prise.
- La **date** : jour de l'année (au 1<sup>er</sup> janvier = jour 1 , ...) et éventuellement l'heure.

Les données de type numérique ne sont pas les plus difficiles à gérer. Cependant, elles ont été traitées par des personnes différentes (5 exactement). Il en résulte une diversité de présentation dans leur mise en forme, ce qui a compliqué quelque peu leur insertion. L'emploi de routines de traitements et d'insertion a été nécessaire (Un système d'importation de données est fourni avec Access mais son utilisation est limitée : voir 3.4.3).

### 3.1.2 Les tables : choix et mode de création.

Le choix de la modélisation n'a pas été trop difficile. Les différentes relations entre les objets sont simples et sans ambiguïtés. En règle général, une table correspond à un type de données (température, humidité, température de brillance...). La table centrale de la base représente les parcelles avec comme clé primaire le numéro de chacune d'elles (numérotation choisie par les chercheurs lors de la manipulation). La plupart des autres tables rapatrient ce numéro comme clé étrangère. En ce qui concerne les champs, une valeur par défaut est systématiquement définie. Les chercheurs de l'I.N.R.A optent habituellement pour le code 9999 pour les valeurs manquantes et c'est donc celui que j'ai choisi.

Décrivons brièvement quelques tables et intéressons nous plus particulièrement à la table "final\_table".

- **Table Fields** : C'est la table regroupant les parcelles et leurs propriétés par numéro. Il est possible que 2 parcelles de la base (donc deux numéros différents) puissent se situer sur le même carré de terrain à des époques différentes. Une parcelle de terrain X peut par exemple contenir la parcelle numéro 13, puis être labouré entraînant la création d'un nouveau numéro de parcelle (une propriété fondamentale, la rugosité, ayant changé). **Le numéro de la parcelle** sera pris comme clé primaire.

Les tables suivantes prendront un **champs numérotation automatique** comme clé primaire, aucun autre champ ou groupe de champs ne pouvant faire office de clé.

- **Table Brightness** : Toutes les mesures de température de brillance (mesure centrale de cette manipulation) sont regroupées dans cette table. Chaque mesure est faite sur une parcelle précise à une date précise. Plusieurs paramètres lui sont associés (angle, polarisation, facteurs correctifs...) . Des facteurs correctifs sont également introduit dans cette table. Des facteurs correctifs correspondant à chaque mesure sont récupérés dans les tables "atmos" et "atmos\_key". A chaque paramètre et facteurs correctifs est associé un champs.
- **table Moisture** : Humidités gravimétriques relevées lors de la manipulation. Cette table contient les champs numéro de parcelle (clé étrangère), jour et heure ainsi que des champs associés aux profondeurs des relevés.
- **Table Temperature** : Table contenant toutes les températures prises sur le site de mesure. C'est de loin la table la plus volumineuse du point de vue espace disque requis (environ 40000 enregistrements sur une trentaine de colonnes). Elle possède les champs numéro de parcelle, jour, heure ainsi qu'un champ pour chaque profondeur de mesure.
- **Table Vegetation** : Elle contient les données sur la végétation. (biomasse, longueur de tige, taille des épis...). Pas de surprise au niveau des champs : numéro de la parcelle, jour, champs correspondants à la mesure faites).
- **Table Pictures** : Les images sont stockés dans cette table. La technologie OLE (liaison et incorporation d'objets) nous permet d'insérer des objets provenant d'autres composants dans un formulaire ou un état Microsoft Access. Ces objets sont généralement des documents créés par un autre composant générant la technologie OLE. Le composant qui fournit ses documents en vue de leur liaison ou incorporation dans d'autres composants est appelé serveur OLE. Lorsqu' on incorpore un objet OLE, Access stocke en fait une copie du document dans l'application finale. Dans le cas présent, les images ont été intégrées en tant que document Microsoft Photo Editor, logiciel de retouche d'images, dans un champs de type OLE.
- **Tables atmos et atmos\_key** : Les tables atmos et atmos\_key contiennent des coefficients de couvertures nuageuses relevés quasiment tous les jours. Des facteurs correctifs sur la mesure de températures de brillance sont également présents dans la table atmos (ils sont fonction des couvertures nuageuses). Ces deux tables sont essentielles pour la recherche des facteurs correctifs nécessaires au calcul d'une mesure corrigée de température de brillance. En effet, suivant le jour et l'heure de la mesure, nous effectuons une première recherche dans la table "atmos\_key". "Atmos\_key" contient des enregistrements sur plusieurs jour a plusieurs heures. Chacun de ces enregistrements contient une deuxième heure correspondant à une observation de nuage enregistrée dans la table "atmos".
- **Table final table** : Avant de parler de cette table, revenons sur un point du cahier des charges : Pour chaque mesure de température de brillance, il serait bon de faire correspondre **d'autres paramètres** relevés à la même heure (humidité, température du sol, rugosité et végétation, données climatiques...). Ces paramètres seront **choisis « à la carte »** par l'utilisateur. Ils devront être ensuite visualisés et

exportés d'un seul bloc. Pour cela, la table "final\_table" est **recrée** en permanence. Dans le formulaire de construction, l'utilisateur choisie une période de mesure, une parcelle et certain paramètre qui lui seront utile . Les mesures de température de brillance effectuées pendant cette périodes seront sélectionnées et insérées dans la nouvelle table "final\_table". Ensuite,

Voici un **exemple** typique de "final\_table" : Les premiers champs sont tous le temps présents. Rappelons que cette table est centrée sur les mesures de températures de brillance. Celles-ci sont donc automatiquement insérées avec les paramètres correspondants : "field number", "dayofyear"(jour de la mesure), "Hour", "Frequency", "Polarisation", "angle", "TB"(la mesure elle même).

L'utilisateur a ensuite demandé la recherche d'autre champs en cochant les cases appropriées :

- Humidité à 0-0,5cm : HM 0-0,5cm (heure mesure) et M0-0.5cm(mesure elle même) .
- Température à 0 cm : HT 0cm et T 0cm.
- La vitesse du vent : "H wind" et "wind velocity".

Numauto	Field number	Dayofyear	Hour	Frequency	polarisation	angle
1	15	162	15.52	1.41 V		0
2	15	162	15.58	1.41 V		10
3	15	162	15.63	1.41 V		20
4	15	162	15.68	1.41 V		30
5	15	162	15.72	1.41 V		40
6	15	162	16.08	1.41 V		30
7	15	162	16.1	1.41 V		30
8	15	162	16.13	1.41 V		50
9	15	162	16.13	1.41 V		50

TB	HM 0-0.5cm	M 0-0.5cm	HT 0cm	T 0cm	H wind	Wind velocity
236.57	15	0.1172	15.52	31.52588	9999	9999
236.59	15	0.1172	15.58	31.41647	9999	9999
238.85	15	0.1172	15.63	31.41882	9999	9999
242.12	15	0.1172	15.68	31.48353	9999	9999
248.28	15	0.1172	15.72	31.53529	9999	9999
241.41	15	0.1172	15.93	32.37	9999	9999
241.74	15	0.1172	15.93	32.37	9999	9999
227.03	15	0.1172	15.93	32.37	9999	9999
227.73	15	0.1172	15.93	32.37	9999	9999

Le programme de création de la table se trouve en **Annexe 2** :

Un **problème théorique** se pose tout de même : la base de donnée possède une table qui varie au cours du temps. Normalement, les tables sont construites au début de la modélisation de manière à répondre aux besoins futurs. Je suis conscient de ce problème mais je n'est pas trouvé d'autre manière de répondre à cette partie du cahier des charges. A cela plusieurs raisons. Il étaient possible de construire la table "Brightness" (température de brillance) avait les 83 champs consultables par l'utilisateur. Seulement, un problème d'affichage et d'exportation se posait. Il ne faut sélectionner que les champs cochés par l'utilisateur, pas les autres. Le formulaire chargé de l'affichage serait donc différent à chaque fois. De plus, il ne pourrait afficher tous les enregistrements en même temps. Un problème de **redondance** des

données présentes dans la base se pose également. Les 83 champs supplémentaires à rajouter dans la table "brightness" contiendront des données déjà présentes ailleurs (sans compter la place qu'elle vont occuper sur le disque).  
 Ne trouvant donc aucune autre solution, j'ai choisie de reconstruire la table à chaque fois. La table nouvellement créée s'exporte facilement (les champs en sont déjà sélectionnés) et la manœuvre est complètement **transparente** du point de vue de l'utilisateur.

### 3.1.3 Méthodes d'insertion des données.

#### Données numériques :

Les divers mesures effectuées sur la base ont été stockées par les chercheurs sous forme de **fichiers texte**.

Exemple de fichier :

```
110 17.20 9 1 23.80 2 10 251.48 0.95 99.00 0.00 2 245.4046 9999.000000
110 17.20 9 1 36.50 2 10 260.57 0.89 99.00 0.00 2 262.1482 9999.000000
112 15.08 9 1 5.05 1 10 214.49 1.06 24.50 0.00 1 250.2663 -12.766060
112 15.08 9 1 5.05 2 10 224.37 1.48 24.50 0.00 1 238.2156 -12.766060
112 15.08 9 1 10.65 1 10 247.41 0.77 24.50 0.00 1 251.0155 2.636570
112 15.08 9 1 10.65 2 10 280.33 0.39 24.50 0.00 1 257.1634 2.636570
112 15.08 9 1 23.80 1 10 261.76 0.94 24.50 0.00 1 257.4527 9999.000000
112 15.08 9 1 36.50 1 10 270.78 0.87 24.50 0.00 1 273.0423 9999.000000
112 15.08 9 1 90.00 1 10 276.81 0.54 24.50 0.00 1 274.0940 9999.000000
112 15.08 9 1 90.00 2 10 277.45 0.80 24.50 0.00 1 274.0365 9999.000000
...
```

Dans un premier temps, j'importe les données dans une table temporaire sans détruire la mise en forme de celles-ci. Le système d'importation de Microsoft Access ne permet pas de trier instantanément les données lorsque celles-ci sont organisées de manière complexe. Ensuite, une routine programmée en V.B.A. se charge de placer toutes les mesures dans les bonnes tables. De manière générale, un pointeur de type "recordset" est placé sur la table temporaire, un autre pointant sur la table à remplir. Une boucle sur la table temporaire permet de passer tous les enregistrements en revue. Chaque champ est pointé à tour de rôle et sa valeur transmise est vers le pointeur sur la table à remplir.

Exemple: Cette routine effectue le rangement de données directement importées dans une table temporaire ("TB931"). Elles sont ensuite rangées dans la table ("Brightness").

```
Public Function tri_ble()
```

*(déclaration des variables : Database= base de données, Querydef=requête, Recordset=enregistrements.)*

```
Dim requete As QueryDef, query1 As QueryDef, query2 As QueryDef, req_delta As QueryDef
Dim i As Integer, j As Integer, k As Integer
Dim p93 As Recordset, rec_delta As Recordset
```

```
Set mabase = CurrentDb
```

(pointeur « table » sur la table brightness, table à remplir)  
 Set table = mabase.OpenRecordset("brightness")  
 (définition d'un pointeur sur une requête temporaire.)  
 Set requete = mabase.CreateQueryDef("")

requete.SQL = "select \* from TB931;"  
 (pointeur « res » sur la table temporaire TB931)  
 Set res = requete.OpenRecordset()  
 DoCmd.Hourglass True

(définition de la boucle sur les enregistrements de la table temporaire. Boucle tant que l'on est pas sur le dernier enregistrement : Not res.EOF. )

```

Do While Not res.EOF
  With table (remplissage de la table brightness par le biais du pointeur « table ».)
    .AddNew (demande d'insertion d'un nouvel enregistrement)
    !pnum = 8
    !dayofyear = res.Fields(1)
    !Hour = res.Fields(2)
    !frequency = res.Fields(4)
    If res.Fields(5) = 1 Then
      !polarisation = "H"
    Else: !polarisation = "V"
    End If
    !angle = res.Fields(6)
    !azymut = res.Fields(3)
    !Tsurf = res.Fields(8)
    !BTdown = res.Fields(9)
    .update (insérer l'enregistrement)
  End With

```

remplissage des champs

...  
 De nombreuses routines du même type ont été nécessaires devant la variété des types de fichiers fournis (5 personnes ont travaillé sur les données Portos93 se qui implique la variété des formats utilisés).

### Photographies :

Toutes les photos de la base sont issues de négatifs. Le scanner utilisé est un Agfa Arcus II au format A4 relié en mode SCSI à une station Sun SparcStation 20 de 64 MO de mémoire vive.

Ce travail de numérisation fut particulièrement difficile. Dans un premier temps, j'ai scanné les négatifs sans trop me préoccuper des paramètres du scanner. Un réglage luminosité/contraste approximatif suffisait. Les photos ainsi obtenues se retouchaient sans problèmes sous Corel Photo Paint version 8. Jusque là, rien de bien exceptionnel. Pour une pellicule, ce fut tout autre chose. Une première retouche sous Corel fut désastreuse. Il ne me restait plus qu'à les scanner de nouveau, en jouant cette fois ci sur le paramétrage du scanner (6 paramètres en tout), méthode très longue puisqu'il faut attendre le passage de celui-ci avant de visualiser les résultats. Des améliorations sensibles ont été obtenue mais le rendu reste



encore à mon sens bien mauvais. Cette pellicule ne sera probablement pas introduite dans la base.

Les numérisations de bonne qualité sont quant à elles introduites dans la table "Pictures" par l'intermédiaire d'un champ de type OLE (voir définition 3.4.2 : table **Pictures**).

## 3.2 Requêtes.

### 3.2.1 Généralités sur les requêtes mises en œuvre dans la base.

La plupart des requêtes générées dans la base portos93 ne sont pas trop compliquées à mettre en place. Elles se contentent en règle générale de sélectionner certains champs suivant des critères tels que le numéro de parcelle, le jour, l'heure, la fréquence... . Cependant, elles sont en grand nombre. De méthodes s'offrent à nous pour la création de requêtes sous Access.

### 3.2.2 mode assistant de requête : un moyen facile et rapide mais...

L'utilisation du mode « visuel » de création de requête fourni par Microsoft Access peut s'avérer un moyen rapide de création. Des requêtes relativement complexes peuvent être mise en route (analyse croisée, regroupement selon critères...). La sélection des tables et des champs se fait par simple clic et la plupart des demandes se font en français, Access se chargeant d'écrire du SQL en tâche de fond. Portos93 en utilise plusieurs, mais celles-ci sont complétées de routines V.B.A. . Pour plus de clarté, examinons cette exemple utilisé dans portos93 :

Ce mode visuel se présente sous forme de tableaux dans lesquels on rentre les champs à sélectionner.

Champs :	Day of year	pnum	hour	0-05	0-1	1-2
Tables :	Moisture	moisture	moisture	moisture	moisture	Moisture
Tri :						
Afficher :	OK	OK	OK	OK	OK	OK
Critères :	Entre doydebut() et doyfin()	pnum()				
Où :						

Cette requête se fait sur la table "moisture" contenant des humidités . Ici, on aimerait sélectionner des enregistrements pour un numéro de parcelle donné dans une fourchette de dates données (numéro parcelle et jours entrés par l'utilisateur dans un formulaire). On utilise pour cela les fonctions "doydebut()", "doyfin()" et "pnum()" programmées en V.B.A. . Ces routines vont chercher dans le formulaire adéquat les paramètres demandés par l'utilisateur. Elles sont appelées par la ligne « Critères » et font référence au champ correspondant. Par exemple, la requête ne sélectionnera que les enregistrements compris entre les jours renvoyés par "doydebut()" et "doyfin()". Voici le listing de ces fonctions très simples :

Public Function doydebut() as integer

doydebut = Forms!infos\_date!doydebut  
(recherche du jour dans le formulaire « infos\_date »)

End Function

Public Function doyfin() as integer

```
doyfin = Forms!infos_date!doyfin  
(recherche du jour dans le formulaire « infos_date »)
```

End Function

Il est ensuite très simple de lier un état à ce genre de requête pour en présenter les résultats. C'est à mon sens leur premier avantage. Elles ne suffisent cependant pas à résoudre tous les problèmes, et ceci malgré l'insertion de routines V.B.A. pour en étendre la portée.

### 3.2.3 Le langage VBA pour l'implémentation des requêtes : un support incontournable.

Nous avons dans la plupart des cas utilisé des requêtes programmées directement dans des modules V.B.A. Une recherche spécifique est souvent plus facile à obtenir via une requête programmée (en comparaison avec les requêtes créées par l'assistant). En effet, celles-ci se composent directement de langage **SQL**, et le paramétrage, moyennant une syntaxe particulière, reste simple. A cet effet, c'est l'objet "**QueryDef**" qui permet de définir une requête.

Exemple de définition d'une variable appelée "req" de type requête :

```
Dim req as QueryDef.
```

Voici une liste des propriétés les plus importantes associées à l'objet QueryDef :

- .OpenRecordset : affecte un pointeur sur le premier enregistrement trouvé par la requête.
- .SQL : Variable de type chaîne (string) : Elle contient une ligne en langage SQL définissant la requête. C'est la propriété la plus importante.
- .Parameters (« nom paramètre ») : Un paramètre utilisé dans la ligne SQL est défini ici.
- .Fields.count : nombre de champs sélectionnés par la requête.
- .Recordsaffected : nombre d'enregistrements trouvés.

### Exemple de programme : Annexe 3.

La plupart des requêtes utilisées dans la base Portos93 ont été écrites de cette manière. Elles présentent l'énorme avantage de pouvoir s'exécuter en mémoire ce qui évite de les insérer à chaque fois dans la base. le gain de temps est appréciable puisqu'un appel disque est supprimé.

## 3.3 L'interface utilisateur .

### 3.3.1 L'esprit « formulaire » Microsoft Access.

La partie visible et interactive du logiciel se présente sous la forme de fenêtres appelées formulaire sous Access. Ces formulaires sont au cœur de toutes les présentations réalisables sous ce logiciel. Une application Access ne démarre d'ailleurs qu'après création

d'un formulaire d'entrée(menu général en fait). Les formulaires permettent d'insérer où de présenter toutes les données de la base. Ils sont reliés entre eux par des événements que déclenche l'utilisateur (click sur un bouton par exemple).

Access fournit de nombreux outils de construction très simple à utiliser. Il suffit de sélectionner un objet dans la boîte à outils et de le déplacer dans le formulaire à l'endroit voulu. Chacun de ses objets (formulaire dans sa globalité compris) peuvent réagir à des événements en lançant des Macros où des programmes VBA :

Exemples d'événements :

- Sur click pour un bouton de commande.
- Sur réception focus pour une zone de texte (réception du curseur ou click utilisateur).
- Sur ouverture pour un formulaire.
- Sur changement pour une liste déroulante.
- ...

Tous les boutons de commandes, les images, les cases à cocher,... de la base Portos 93 sont associés à des programmes VBA (sauf les boutons de navigation qui font appel à des macros simples).

3.3.2 La mise en forme que nous avons adoptée : présentation des menus.

Nous ne nous attarderons pas sur la description des différents formulaires utilisés dans la base Portos93 (environ une trentaine). Une consultation du CD-ROM et de la documentation (Annexe ) fournis avec ce rapport seront beaucoup plus parlant. Notons cependant les 3 axes principaux qui ont guidés la répartition des tâches dans la base :

➤ **Inventaire général sur les données présentes:**

Celui-ci est établi par parcelle avec propriétés de celles-ci : premier choix du menu général (DATA BASE CONTENT). L'utilisateur doit pouvoir connaître le type de données disponibles sur chaque parcelle avant d'aller éventuellement les rechercher.

➤ **Consultation des données :**

Deuxième bouton de commande du menu général (DATA EXTRACTION). Deux subdivisions importantes apparaissent ici.

- Consultation de chaque paramètre séparément en fonction de la parcelle choisie et de la fourchette de date (Par défaut, une requête recherche puis affiche cette fourchette). Ce sont les sous-menus DATA EXTRACTION, CLIMATIC PARAMETERS et ENERGY BALANCE. L'utilisateur peut consulter ou exporter les données les unes après les autres (températures de brillance, températures, humidités, végétation). Ne sont ici fournis que des données brutes. L'utilisateur pourra comme cela revenir aux données brutes lorsque les interpolations effectuées dans l'extraction multi-source ne le satisferont pas.
- Demande de requête multi-source (EXTRACTION OF MULTISOURCE DATA). C'est en fait tout le processus de construction de la table

"final\_table" avec sélection de paramètres à la carte (voir 3.1.2, table "final\_table"). Certaines des données retournées sont interpolées.

➤ **Photographies:**

Mise à disposition des photographies prises pendant la manipulation. Celles-ci sont classées par parcelle.

➤ **Notes des auteurs :** (bouton INFORMATION). Seront présents ici les commentaires, publications et statistiques des chercheurs.

### **3.4 Dernière étape : création d'un exécutable avec programme d'installation.**

#### **3.4.1 Office Developer.**

La base sous Access terminée, il restait à la rendre distribuable. Rappelons que cette base sera envoyée à des utilisateurs ne disposant pas forcément du SGBD de Microsoft. Pour ce faire, nous avons utilisé le logiciel « Office Developer ». Il permet, à partir d'un fichier Access (\*.mdb ou \*.mde) de réaliser une configuration exportable de la base de données. Cette configuration comprend une interface d'installation ainsi qu'une version Access exportable, le tout facilement exécutable à partir de menus déroulants.

Office Developer se présente sous forme de menus invitant le programmeur à spécifier un certain nombre de paramètres en vue de la création du logiciel :

- ❑ **Fichiers à insérer :** ici se sera le fichier Access de Portos93, des fichiers de type « readme.txt » ainsi que le fichier correspondant à la documentation (fichier Word).
- ❑ **Création de raccourcis** dans le menu démarrer. Chaque fichier sélectionné au dessus pourra obtenir un raccourci installé dans le menu démarrer de Windows.
- ❑ **Copie de certains utilitaires à l'installation :** En ce qui nous concerne, nous aurons besoin de la **version Access distribuable**. Elle est fournie avec « Office Developer » et permet de lire les fichiers base de données Microsoft Access. Nous installons également une version allégée de Microsoft Graph (affichage des graphiques utilisés dans Portos) et un driver pour l'exportation des données.
- ❑ **Choix du support :** Les supports CD\_ROM ou disquettes sont disponibles. Nous avons choisi le CD-ROM comme prévu. Dans le cas des disquettes, Office Dev. construit des sous répertoires contenant chacun les fichiers à copier sur une disquette.

Après calculs, Office Developer crée un répertoire avec la liste des fichiers à graver. Ils occupent un espace disque de **130 Mo** environ (125Mo pour la base elle même, les 6Mo restant pour les fichiers d'installation, la version Access exportable et les documentations). Il ne reste plus qu'à lancer le fichier *install.exe*.

#### **3.4.2 Constatations:**

Cette base de données a été réalisée sur une machine puissante, en l'occurrence un PentiumII 266MHz composé de 64 Mo de mémoire vive. Cette configuration musclée apporte un grand confort de travail au niveau des temps de réponse du système gérant la base. Mais comment le logiciel allait-il fonctionner sur des machines plus modestes ? Nous lançons donc une batterie de tests sur plusieurs machines présentes au laboratoire de Science Du Sol. Cela consistait en une installation normale puis lancement de quelques menus et requêtes pour estimer les temps de réponse. Nous avons identifié 3 catégories d'ordinateurs :

- 1<sup>er</sup> catégorie : Série 486 DX2-66 au DX4-100 jusqu'au Pentium 75.
- 2<sup>e</sup> catégorie : Pentium 100 au Pentium 166.
- 3<sup>e</sup> catégorie : Pentium 166 et plus.

### Les résultats :

#### 1<sup>er</sup> catégorie :

Le premier test a été effectué sur un 486 DX2-66MHz. Premières constatations, les programmes V.B.A s'exécutent assez rapidement et la recherche d'enregistrements est assez rapide. Par contre, l'affichage des menus (ou formulaire) est vraiment trop lent. Quand à la bibliothèque d'images, c'est une catastrophe. Ces ralentissements ne sont cependant pas étonnant. Me laissant entraîner par la puissance de la machine qui m'a servi à la base, je n'ai pas hésité à embellir les formulaires de photos et dessins en tout genre. De plus, en ce qui concerne les photos de la manipulation, celles-ci ont été scannées avec une grande définition en un million de couleurs. Les ralentissements ne sont donc pas surprenants. Nous avons décidé, au vue de ces constatations, de fournir sur le CD-ROM une **version allégée de Portos 93**. C'est la seule solution pour une utilisation confortable de ce logiciel sur des machines âgées. Les formulaires dans leur présentation sont dépouillés au maximum et les photos de la manipulation disparaissent complètement. Le gain de place obtenu au niveau du logiciel est également important. Les machines de génération plus ancienne ne dispose pas de disque dur de grande capacité et trouver 130 Mo devient vraiment difficile. La version allégée n'occupent plus que **40 Mo**, une taille beaucoup plus raisonnable.

#### 2<sup>e</sup> catégorie :

En ce qui concerne les machines moyenne gamme (Pentium 100 au Pentium 166),l'utilisateur pourra au choix installer l'une ou l'autre des deux versions. La version normale tourne correctement avec quelques ralentissements pouvant justifier l'installation de la version allégée (sur un Pentium 100 notamment).

#### 3<sup>e</sup> catégorie :

Pas de problèmes particuliers relevés.

#### 3.4.3 Problèmes communs aux trois catégories de machines :

La première fois que nous avons utilisé Portos93 sur une autre machine, des erreurs fréquentes se sont produites sur des requêtes particulières liées à la table des températures de brillance. En fait, on s'est vite aperçu que les erreurs dépendaient de la fréquence sélectionnée

par l'utilisateur (paramètre lié à la mesure de température de brillance). Ce paramètre est sélectionné dans une liste déroulante dont les valeurs disponibles ont été saisies manuellement.

**Or, les valeurs saisies manuellement sont sauvegardées tel quel et ne sont plus prises en compte par les paramètres régionaux de Windows.**

En l'occurrence, j'avais saisi des valeurs de type anglo-saxon (1.41 en comparaison avec 1,41 par exemple) alors que la machine essai était configurée en Français (1.41 n'est pas reconnu).

**Pour y remédier, toutes valeurs affichées dans un formulaire doit provenir d'une table, ou d'une requête elle-même basée sur une table.**

En effet, les valeurs numériques de chaque table sont automatiquement convertis suivant les paramètres régionaux de la machine. Nous avons donc créer une table simple contenant les différentes fréquences utilisées par le radiomètre. Ensuite, les listes déroulantes affichant le choix de la fréquence sont directement liées à cette table par une requête SQL.

## CONCLUSION.

Voilà, les deux mois de stage passés à l'unité de Science Du Sol arrivent à leur fin. Les dernières améliorations sont en cours et la première version de Portos93 naîtra d'ici quelques heures. Il me reste donc quelques instants pour remercier chaleureusement les deux chercheurs à l'origine de ce stage : André Chanzy et Jean-Pierre Wigneron. Ils m'ont fourni un support idéal pour la création et la réalisation d'une première base de données. Cette mission alliait en effet simplicité dans le type de données à gérer (numérique essentiellement), mais également complexité dans les recherches demandées et dans la mise en forme. Le travail demandé fut en somme vraiment bien équilibré. Cet équilibre, on le retrouve dans leur façon de travailler et de raisonner, ce qui ne pouvait être que source d'enrichissement personnel.

J'ai également beaucoup appris dans le domaine de la base de donnée. Certes, il me reste encore beaucoup de travail, notamment dans le domaine de la gestion multi-utilisateurs, thème que je n'avait pas à aborder lors de ce stage. Mais, je me suis familiarisé avec le langage Visual Basic très utilisé sous l'environnement Windows et bien sûr avec le SGBD Access. J'ai pu renouer avec "plaisir" aux joies de la programmation objet et de l'utilisation des pointeurs.

J'ai également compris qu'il était souvent difficile de mettre un terme à un projet et qu'il était plus facile de commencer que terminer. En effet, nous trouvions sans cesse des améliorations à apporter sur ce logiciel, que se soit au vue des essais sur les diverses machines du centre, ou encore en se fondant sur les commentaires des diverses personnes l'ayant fait fonctionner. Cela a entraîné de multiples reports dans la finalisation du produit (production du CD-ROM). Je vous pris de bien vouloir accepter mes excuses pour le retard dans la remise de ce rapport.

Vous trouverez dans ce rapport un CD-ROM contenant le logiciel Portos93. Il ne sera pas tout à fait complet, les fichiers texte de description de la manipulation étant en cour de rédaction ainsi que la version allégée en cour de préparation. Encore tous mes remerciements à l'I.U.P., et aux personnes de l'unité de Science Du Sol de l'I.N.R.A. d'Avignon qui m'ont fait confiance.

# **BIBLIOGRAPHIE**

Thèses et ouvrages utilisés pour la réalisation de Portos93 :

**Thèse présentée par Laurent Laguerre sur Portos93** : Novembre 1995

**Thèse présentée par André Chanzy sur la télédétection.**

**Grand Livre Microsoft Access97** : J .BAR et I.BAUDER chez Micro Application.

**Comment créer des applications Microsoft Access 97** : Microsoft Edition

**Microsoft Office 97, guide du programmeur Visual Basic.**

**Précis de télédétection volume2** : Ferdinand Bonn



# ANNEXES.

ANNEXE1: Exemple de programmation Visual Basic Access.

ANNEXE2: Programme de création de la table "final\_table".

ANNEXE3: Exemple d'utilisation d'une requête programmée sous Visual Basic.

## ANNEXE 1 :

### Exemple simple d'utilisation des objets Database, QueryDef, Recordset et DoCmd dans une routine de Portos93.

Ce programme lance une requête sur la table "moisture", table contenant des humidités. Cette requête est basée sur un numéro de parcelle et sur une fourchette de dates entrés par l'utilisateur. La routine va ensuite regarder si l'utilisateur demande un rapport, un graphique ou une exportation puis va afficher les résultats de la requête.

Public Function hum()

*(déclaration des variables.)*

*---pointeur de type base de donnée---*

Dim mabase As Database

*--- pointeur de type requête---*

Dim requete As QueryDef

*---pointeur de type enregistrement---*

Dim res As Recordset

Dim choix As Integer

Dim chaine As String

Set mabase = CurrentDb *(mabase pointe sur la base courante ouverte)*

*Cette commande crée une requête temporaire sur la base de donnée pointée par mabase*

Set requete = mabase.CreateQueryDef("")

*(utilisation de la propriété .SQL associée à la classe QueryDef. Access exécute du code SQL en tâche de fond. Il suffit simplement lui donner une chaîne de caractère contenant l'instruction SQL à exécuter. C'est ce qui est fait ici avec .SQL . )*

requete.SQL = "Parameters dtedebut integer,dtefin integer,n integer; SELECT moisture.[day of year], moisture.hour, moisture.[0-05], moisture.[0-1], moisture.[1-2], moisture.[2-3], moisture.[3-4], moisture.[4-5], moisture.[5-7], moisture.[7-10], moisture.pnum FROM moisture where (moisture.[day of year] between [dtedebut] And [dtefin]) AND (moisture.pnum=n);"

*(Des paramètres utiles pour la construction de la ligne SQL ont été déclarés au début de l'instruction. Il faut maintenant leur affecter une valeur. Celles-ci ont été rentrées par l'utilisateur dans le formulaire « infos\_date ».)*

requete.Parameters("dtedebut") = Forms!infos\_date!doydebut

requete.Parameters("dtefin") = Forms!infos\_date!doyfin

requete.Parameters("n") = Forms!infos\_date!liste\_parcelle

*(« res » va pointer sur le premier enregistrement découvert par la requête)*

Set res = requete.OpenRecordset()

*(on regarde quel choix a fait l'utilisateur : affichage des enregistrements, graphique associé ou exportation.)*

choix = Forms!infos\_date!choix

*(if la requête n'a rien trouvé, ouverture du formulaire annonçant qu'il n'y a pas d'enregistrements disponible : utilisation de la classe DoCmd gérant entre autre l'ouverture/fermeture des formulaires.)*

If res.EOF Then

DoCmd.OpenForm ("pas\_dinfos") *(pas d'enregistrements)*

*(Sinon affichage de l'option demandée en fonction de « choix ».)*

Else:

*(demande d'impression ?)*

If Forms!infos\_date!print = False Then

Select Case choix

Case 1

DoCmd.OpenReport "Einfos\_date\_hum", acViewPreview

Case 2

DoCmd.OpenReport "G\_infos\_date\_hum", acViewPreview

Case 3

chaine = Forms!infos\_date!chemin

DoCmd.TransferText acExportDelim, , "infos\_date\_hum", chaine

End Select

Else:

Select Case choix

Case 1

DoCmd.OpenReport "Einfos\_date\_hum", acViewPreview

DoCmd.OpenReport "Einfos\_date\_hum", acViewNormal

Case 2

DoCmd.OpenReport "G\_infos\_date\_hum", acViewPreview

DoCmd.OpenReport "G\_infos\_date\_hum", acViewNormal

Case 3

chaine = Forms!infos\_date!chemin

DoCmd.TransferText acExportDelim, , "infos\_date\_hum", chaine

End Select

End If

End If

*(destruction des pointeurs)*

res.Close

requete.Close

mabase.close

End Function

## ANNEXE 2 :

### Programme de création de la table final table.

*(Programmé est lancé lorsque l'utilisateur clique sur le bouton « Launch » du formulaire de paramétrage multi-sources après avoir choisi les données dont il avait besoin. Une partie du code a été supprimé.)*

---

*(définition d'une structure comportant deux chaînes : une chaîne comprendra le nom du champ correspondant à l'heure où le paramètre X a été mesuré, l'autre comprendra le nom du champ correspondant au paramètre X proprement dit.)*

*(nouveau type struct\_chaine)*

```
Type struct_chaine
  nom_champ As String
  heure_champ As String
End Type
```

*( effacement de la table en vue d'une recréation : utilisation de l'objet DoCmd gérant les interventions système.)*

```
DoCmd.DeleteObject acTable, "final_table"
```

*(création de la nouvelle table : affectation du pointeur new\_table par l'intermédiaire de la propriété « CreateTable » de l'objet Database.)*

```
Set new_table = mabase.CreateTableDef("final_table")
```

*(création des champs dans la table, utilisation de « .CreateField », propriété de la classe TableDef. )*

*(clé primaire = nombre automatique)*

```
Set champ(0) = new_table.CreateField("numauto", dbLong)
```

*(champ numero auto=>clé primaire)*

```
champ(0).Attributes = champ(0).Attributes + dbAutoIncrField
new_table.Fields.Append champ(0)
```

*(création des champs correspondants à la mesure de température de brillance : champs présents par défaut dans la table final\_table comme vu précédemment .*

*Nouvelle utilisation de CreateField(« nom\_champ »,type de champ)*

```
With new_table
```

*(numero parcelle sur laquelle a été faite la mesure)*

```
.Fields.Append .CreateField("field number", dbInteger)
```

*(champ correspondant au jour de la mesure)*

```
.Fields.Append .CreateField("dayofyear", dbInteger)
```

```
.Fields.Append .CreateField("hour", dbSingle) ( champ heure)
```

```
.Fields.Append .CreateField("frequency", dbSingle) (fréquence)
```

```
.Fields.Append .CreateField("polarisation", dbText, 1) (polarisation)
```

```
.Fields.Append .CreateField("angle", dbInteger) (angle)
```

```

.Fields.Append .CreateField("TB", dbSingle) (Temperature de brillance)
.Fields.Refresh
End With

```

*(création d'un index)*

```

Set idx = new_table.CreateIndex("clé primaire")
Set champ_index = idx.CreateField("numauto", dbLong)
idx.Fields.Append champ_index
idx.Primary = True
new_table.Indexes.Append idx

```

nbr\_champ = 6

remplir\_tableau (*check up des cases à cocher. L'utilisateur vient de sélectionner les champs. Remplissage du tableau de booléens « bool » avec la fonction remplir\_tableau*)

i = 0

*(création des champs associés à la table brightness)*

```

Do While i < 6
    If bool(i) = True Then
        renvoie_nom_champ i, chaine (utilisation de la fonction
renvoie_nom_champ : si la case i est sélectionnée, cette fonction
renvoie le nom du champ associé à cette case.)

        Set champ(nbr_champ) = new_table.CreateField(chaine.nom_champ,
dbSingle)
        (création du champ dénomé « nom_champ »)
        champ(nbr_champ).DefaultValue = 9999
        (valeur par défaut du champ : 9999, valeur demandée par le maitre de
stage)
        new_table.Fields.Append champ(nbr_champ)
        nbr_champ = nbr_champ + 1
    End If
    i = i + 1
Loop

```

*(création des champs associés aux tables moisture, temperature, climH et Energy. un champ correspondant à l'heure de la mesure est rajouté. La fonction renvoie\_nom\_champ initialise une variable « chaine » qui contient les noms des champs en fonction de la case cochée i (tableau bool) . « Chaine » est implémentée suivant une structure créée au début : struct\_chaine.)*

Do While i < 54

```

    If bool(i) = True Then
        renvoie_nom_champ i, chaine
        Set champ(nbr_champ) = new_table.CreateField(chaine.heure_champ,
dbSingle)
        champ(nbr_champ).DefaultValue = 9999
        new_table.Fields.Append champ(nbr_champ)
    End If
    i = i + 1
Loop

```

```

        nbr_champ = nbr_champ + 1

        Set champ(nbr_champ) = new_table.CreateField(chaine.nom_champ,
dbSingle)
        champ(nbr_champ).DefaultValue = 9999
        new_table.Fields.Append champ(nbr_champ)
        nbr_champ = nbr_champ + 1
    End If
    i = i + 1
Loop

```

*(création des champs associés à la table vegetation : pas de champ heure à rajouter)*

```

Do While i < NBR_CASE
    If bool(i) = True Then
        renvoie_nom_champ i, chaine
        Set champ(nbr_champ) = new_table.CreateField(chaine.nom_champ, dbSingle)
        champ(nbr_champ).DefaultValue = 9999
        new_table.Fields.Append champ(nbr_champ)
        nbr_champ = nbr_champ + 1
    End If
    i = i + 1
Loop

```

```

(----- création des champs terminée -----)
        mabase.TableDefs.Append new_table
(----- la nouvelle table est maintenant dans la base -----)

```

*Cette routine enchaîne sur le remplissage de la table "final\_table" nouvellement créée. Des requêtes sont lancées sur les divers tables contenant les champs demandés par l'utilisateur.*

## ANNEXE 3 :

Exemple simple d'utilisation d'une requête programmée sous Visual Basic.

*(Cette requête se fait sur la table des humidités ("moisture"))  
(On recherche les enregistrements disponibles pour une parcelle donnée dans une fourchette de dates donnée.)*

Public Function req\_hum()

*(déclaration des variables.)*

Dim mabase As Database *(pointeur sur la base)*

Dim requete As QueryDef *(pointeur « requete » de type QueryDef)*

Dim res As Recordset *(pointeur sur les enregistrements obtenus par la requête)*

Dim choix As Integer

Dim chaine As String

*(" mabase" pointe sur la base courante ouverte)*

Set mabase = CurrentDb

*(Définition du pointeur "requete" : ce sera une sélection sur la base pointée par « mabase ». Ne pas définir de nom pour cette requête entraîne son exécution automatique en mémoire. Elle n'est pas copiée sur le disque.)*

Set requete = mabase.CreateQueryDef("")

*(Chaîne de caractères représentant la ligne SQL : Parameters... SELECT ... FROM...Where.... définition des paramètres au début.)*

requete.SQL = "Parameters dtedebut integer,dtefin integer,n integer; SELECT moisture.[day of year], moisture.hour, moisture.[0-05], moisture.[0-1], moisture.[1-2], moisture.[2-3], moisture.[3-4], moisture.[4-5], moisture.[5-7], moisture.[7-10], moisture.pnum FROM moisture where (moisture.[day of year] between [dtedebut] And [dtefin]) AND (moisture.pnum=n);"

*(affectation des paramètres choisis par l'utilisateur (le formulaire « infos\_date ») dans les variables initialisées dans la propriété .SQL .)*

requete.Parameters("dtedebut") = Forms!infos\_date!doydebut *(premier jour)*

requete.Parameters("dtefin") = Forms!infos\_date!doyfin *(dernier jour)*

requete.Parameters("n") = Forms!infos\_date!liste\_parcelle *(numéro de la parcelle)*

*(affectation d'un pointeur sur les enregistrements sélectionnés par la requête)*

set rec = requete.OpenRecordset()

...

End Function