

Epidata : contrôle à la saisie

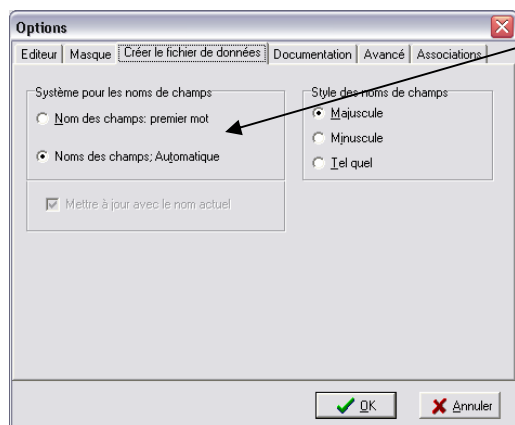
Niveau 1 bis avec EPIDATA Entry

Bernard BRANGER – Réseau « Sécurité Naissance – Naître ensemble » des Pays de la Loire
1, allée Baco – 44000 NANTES. Tél 02 40 48 55 81 –
Courriel : bernard.branger@naître-ensemble-ploire.org
Janvier 2010

Préalables : le nom des variables

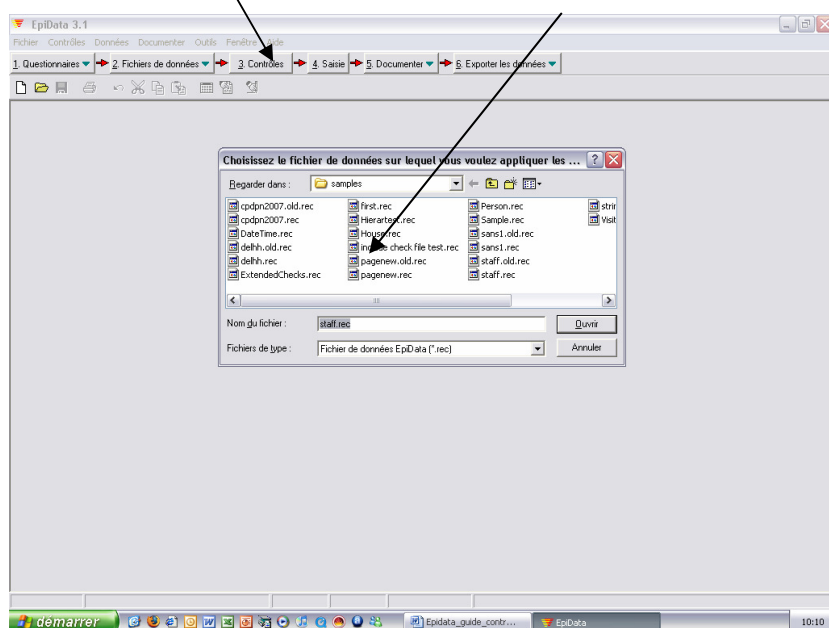
Noter que les noms des variables doivent être bien définis. Par défaut (automatique), ce sont les 10 premières lettres qui sont retenues (sans les espaces et sans les caractères accentués).

Pour modifier et ne garder que le premier mot, aller dans *Fichier, Options, Créer le fichier de données* et cocher « Nom des champs : premier mot ». Attention, si vous avez le premier mot de plusieurs champs qui commencent pas le mot « Date » par exemple, Epidata prend comme nom de la première variable « Date » et la deuxième « Date1 ». Sinon, ne pas mettre d'espace et le mot sera les 10 premières lettres.

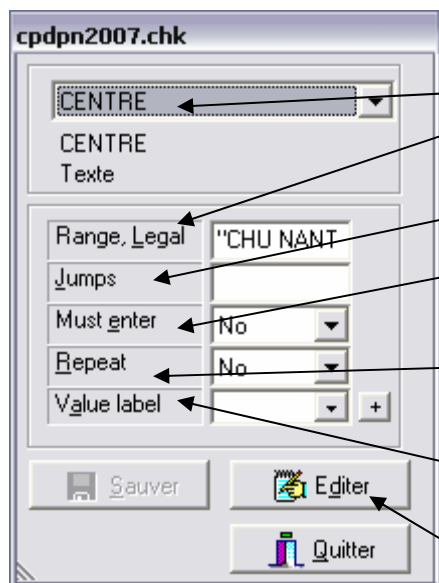


Les premiers pas avec le menu *Contrôle*

Cliquer sur **Contrôle** et choisir le fichier .rec désiré



Pour chaque champ, un menu s'affiche comprenant



- Le nom de la variable tel qu'il a été retenu par le logiciel
- *Range, Legal* : Case pour établir des limites au codage et déterminer un codage simple. Exemple : 25-90 OU 1,2,3,9 (voir infra)
- *Jumps* : Case pour sauter à un prochain champ d'une variable (pas celui qui suit). Exemple : 1>V1,2>V2. OU AUTOJUMP V1 (voir infra)
- *Must enter* : Case pour que le champ soit obligatoirement rempli (sinon la saisie se bloquera) ; délicat à utiliser
- *Repeat* : Case pour « répéter » le codage de l'enregistrement précédent ; vaut pour le nom de son établissement par exemple
- *Value label* : Case pour utiliser un codage fréquent appelé label (comme 1 pour « oui » et 2 pour « non »)
- **Editer** : Case pour écrire un programme de contrôle

Tout n'est pas utile pour chaque champ. Si vous êtes le seul à faire la saisie, très peu de contrôles sont nécessaires. Tout au plus, quelques « range » pour ne pas se tromper. Si plusieurs personnes saisissent, il est préférable, pour harmoniser les saisies, de fixer des codages à chaque variable.

Exemple pour un fichier type (fichier .ges) :

```

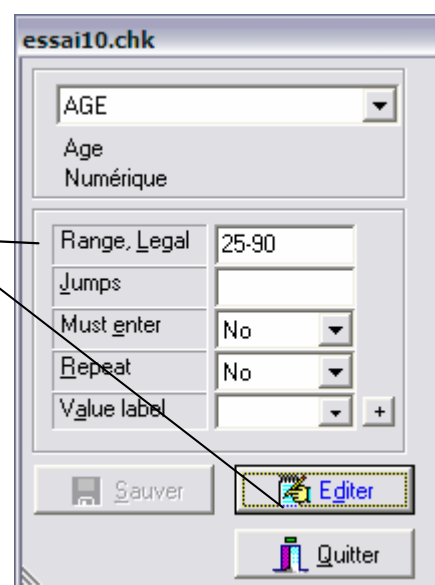
CENTRE _____
DATE ENQUETE <dd/mm/yyyy>          DATE NAISSANCE <dd/mm/yyyy>
AGE ###          SEXE #          SAUT #          GROUPE #          VILLE _____
    
```

Le calcul de l'âge se fait dans **Editer** : il faut écrire la différence entre la date de naissance et la date de l'enquête pour obtenir l'âge en jours, puis diviser par 365.25 (le nombre de jours moyens dans une année en tenant compte des années bissextiles pour une période longue). A signaler que le mot `After entry` se met automatiquement. Le résultat à la saisie sera le suivant : faire ↵ sur le champ qui se remplit automatiquement.

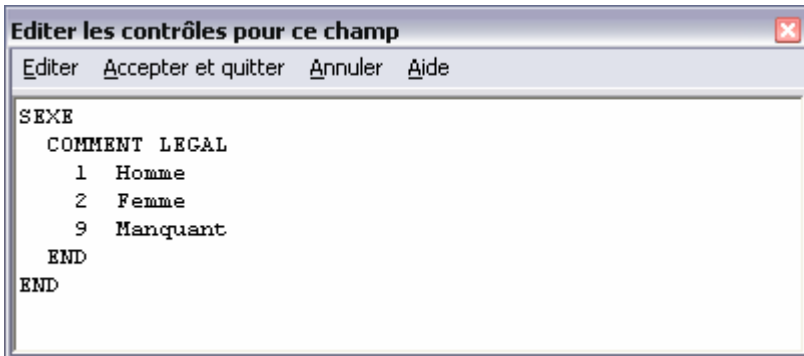
On peut aussi rajouter des étendues (ou range en anglais) de type 25 - 90 (ans). Le tout se traduit automatiquement par le programme suivant dans *Editer* avec `range` comme commande.

```

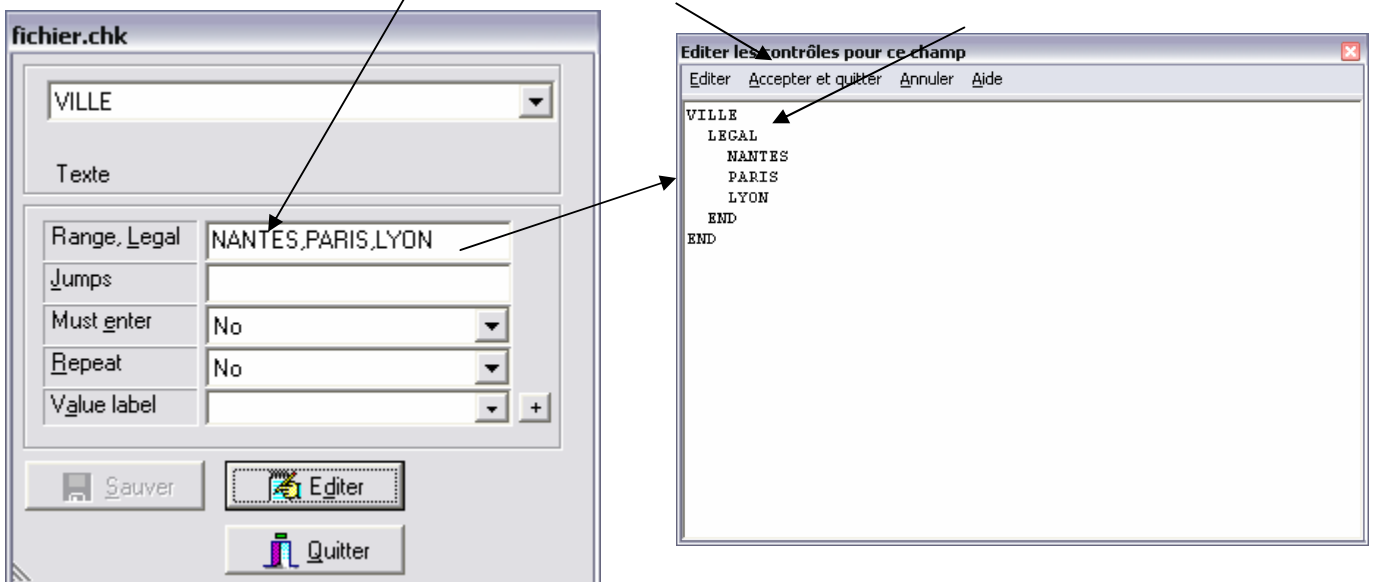
Editer les contrôles pour ce champ
Editer  Accepter et quitter  Annuler  Aide
AGE
RANGE 25 90
AFTER ENTRY
AGE=(DATEENQUET - DATENAISSA) / 365.25
END
END
    
```



Pour le codage du Sexe : passer par le menu **E**diter avec `comment legal` pour déterminer un codage à 3 modalités. Noter les deux « end ». A signaler que les décalages de lignes se mettent automatiquement. Le résultat à la saisie sera le suivant : faire F9 pour afficher le menu et se déplacer avec les $\uparrow\downarrow$ pour saisir.



Pour « Ville », on peut utiliser *Range,Legal* en mettant les codages séparés par des virgules. Noter que le programme implicite « derrière » cette manœuvre (à voir avec le menu **E**diter) est le suivant, avec `legal` comme commande.



Ecrire les programmes directement "à la main"

En fait, la page **C**ontrôle crée automatiquement un fichier de type fichier.chk. On peut le lire directement dans *F*ichier, *O*uvrir (il est en principe dans le même répertoire que les autres fichier.*). Il est libellé ainsi (une seule page à la suite) :

```
AGE
RANGE 15 45
AFTER ENTRY
AGE=(datenquet - datenaissa) /365.25
END
END

SEXE
COMMENT LEGAL
1 Homme
2 Femme
9 Manquant
END
END
```

```
VILLE
LEGAL
NANTES
PARIS
LYON
"LE MANS"
END
END
```

Noter les " et " qui entoure des codages comprenant des espaces (pas nécessaire pour un mot entier). Noter qu'un `end` unique ou le deuxième `end` est suivi d'une ligne vierge. Distinguer bien `legal` qui commande du texte (comme avec la variable *ville*) et `comment legal` qui commande un code avec sa traduction (comme avec la variable *sexe*).

Quelques autres commandes supplémentaires

→ LABEL : pour définir des codages pour plusieurs variables.

Dans chaque variable concernée, écrire `command legal use binaire`

```
LABELBLOCK
  LABEL binaire
    1  Oui
    2  Non
    9  NSP
  END
END
```

→ IF ... THEN ... : permet de proposer un codage conditionnel à un remplissage préalable. Terminer par `ENDIF`

```
IF age > 20 THEN
  old = "JEUNE"
ENDIF
```

OU

```
IF age > 65 THEN
  old = "SENIOR"
ELSE
  old = "JUNIOR"
ENDIF
```

OU : remplissage conditionnel à une réponse préalable à une autre variable

```
V2
IF V1=1 THEN
  COMMENT LEGAL
  1  Alabama
  2  "New York"
  3  Nevada
  END
ENDIF
IF V1=2 THEN
  COMMENT LEGAL
  1  "Nova Scotia"
  2  Quebec
  END
ENDIF
END
```

→ AUTOJUMP VARIABLE : à partir d'un remplissage d'un champ, saut vers un autre champ déterminé (VARIABLE). Avec `AUTOJUMP END`, saut vers la fin de la fiche

→ JUMPS : saut conditionnel à une modalité de réponse. Si la réponse est 1, saut vers V5 et, si la réponse est 2, saut vers V10, et si réponse est 3, saut vers la fin de la fiche) :

```
VARIABLE
JUMPS
  1  V5
  2  V10
  3  END
END
```

→ NOENTER : pour ne pas remplir un champ (champ de résultat de calcul par exemple)

→ RANGE 1/3/2001 31/3/2001 : étendue pour des dates

→ TOPOFSCREEN 2 : permet de mettre le champ en haut de page, pour avoir une vision plus large d'un questionnaire

→ USE : pour se référer à un codage déjà fait

```
V1
  LEGAL
  2
  4
  END
END

V2
  LEGAL USE V1
END
```

→ IGNOREMISSING : pour, en cas d'addition, ignorer les champs vides (les met à 0)

```
V4
  BEFORE ENTRY
    IGNOREMISSING
    V4=V1+V2+V3
  END
```

→ VARIABLE <IDNUM> : variable qui s'incrémente à chaque fiche (en commençant à 1).

Voir exemple dans Annexe.

Fichiers relationnels

Un fichier relationnel est un fichier relié à un autre, avec généralement des variables communes.

Exemples pour 2 fichiers reliés :

- fichier "mere.rec" → concerne par exemple des femmes enceintes ou accouchées ou des mères. On parle de données ou de fichier MERE.

- fichier "fille.rec" → correspond aux enfants de cette femme qui eut concerner plusieurs enfants de cette femme soit des jumeaux, soit la fratrie) → on parle de données ou de fichier FILLE. Il faut une variable commune qui, dans l'exemple suivant, est **fiche**.

Type de fichier	Fichier "Mère" : mere.rec	Fichier "Fille" : fille.rec
Masque de saisie (exemple) *.ges	Agemere ## Fiche ### Enfants #	Fiche ### Ageenfant ## Sexe # Groupe #
Contrôles à la saisie *.chk Le nom de la variable de key unique doit être le même après relate La variable qui ouvre la fiche du fichier fille (ici enfants) doit être la dernière de la fiche du fichier père.	Fiche key unique end enfants relate fiche fille.rec end OU enfants if enfants>1 then relate fiche fille.rec endif end	fiche key end
Fichier de données *.rec	A la saisie : - soit enfants est rempli et le fichier fille s'ouvre - soit, en cas de enfants supérieur à 1, le fichier fille s'ouvre. Si enfants est saisi avec la valeur 0, le fichier fille ne s'ouvre pas.	- La variable <i>fiche</i> a un champ déjà rempli égal à celui du fichier <i>mere</i> - A la saisie, remplir tous les autres champs et passer à une fiche suivante éventuellement - A la fin, faire F10 pour revenir au fichier <i>mere</i> et passer à la fiche suivante
Analyse dans Analysis 2.1 *.rec	Analyse de x fiches "mère" remplies Voir dans Analysis, les commandes avec MERGE pour analyser un fichier avec les variables de l'autre : read mere.rec merge fiche fille.rec	Analyse de y fiches "filles" différentes des x fiches "mères"

Annexe : Exemple de programme pour un fichier non relationnel**Questionnaire `essai10.qes`**

NUMERO <IDNUM>	CENTRE _____	
DATE NAISSANCE <dd/mm/yyyy>	DATE ENQUETE <dd/mm/yyyy>	
Age ##	OLD _____	
SEXE #	GROUPE #	VILLE _____

Programme `essai10.chk`

```

LABELBLOCK
  LABEL BINAIRE
    1 OUI
    2 NON
    9 NSP
  END
END

CENTRE
  LEGAL
    "CHU NANTES"
    "CHU ANGERS"
    "CH LE MANS"
  END
REPEAT
END

AGE
  RANGE 25 90
  AFTER ENTRY
    AGE=(DATEENQUET - DATENAISSA) / 365.25
  END
END

OLD
  AFTER ENTRY
    IF age>65 THEN
      old="SENIOR"
    ELSE
      old="JUNIOR"
    ENDIF
  END
END

SEXE
  COMMENT LEGAL
    1 HOMME
    2 FEMME
    9 NSP
  END
  JUMPS
    1 GROUPE
    2 VILLE
  END
END

GROUPE
  COMMENT LEGAL USE BINAIRE
END

VILLE
  COMMENT LEGAL
    1 NANTES
    2 "LE MANS"
    3 ANGERS
    4 "HORS REGION"
  END
END

```