Clinical Epidemiology and Biostatistics Unit
Murdoch Childrens Research Institute &
Department of Paediatrics, University of Melbourne

Database setup and EpiData

Contents

Database setup and data entry: basic principles

A database is used to input the information in your questionnaire in a format that can be used for analysis.

Process

Data collection, entry and analysis should follow this order:

Create a coding sheet.

Design the questionnaire. This step involves designing both the paper form and the computer database.

Enter data.

Clean and correct data. Questionnaire responses may be illogical, and mistakes occur during data entry. You need to identify and correct as many errors and inconsistencies as possible before starting the process of analysing data.

Export clean data for use in analysis.

Analysis using Stata or similar software (not covered in this document).

Questionnaire design

In order for your database to be effective, your questionnaire needs to be well designed.

Variable names

A ***field*** or ***variable*** contains responses to particular questions. Each variable in the database has a unique name, such as "idnum", "sex" or "age". In EpiData, a variable name can be up to 10 characters long, can contain letters or numbers, and must begin with a letter. You should be consistent in making variable names uppercase or lowercase. Lowercase is recommended if you will be analysing your data in Stata.

Variable names should be meaningful and consistent. There are two common methods for creating variable names:

1. The variable name is a common word based on the question text. Examples include "age", "sex", "bloodtype". These are easy to remember and understand, but in questionnaires with many variables it can become difficult to generate unique variable names within the database system's constraints.
2. Most questionnaires will have numbered questions, so the variable names can be based on the question number, for example `q1, q2`… In EpiData, variable names must begin with a letter, so you can use a system like `q1`, or `a1, a2, b1, b2` if the questionnaire has sections identified by letter.

Whichever naming system you use, it should be consistent within the database.

Coding sheets

You should always create a ***coding sheet*** (also called a ***codebook***) for your questionnaire and database, and it is a good idea to begin this process when you are writing the questionnaire, rather than afterwards. A coding sheet is used to keep track of the all the variables in your questionnaire. The following information should be provided for each variable:

- Variable name used in the database
- Corresponding text/question in the questionnaire
- Type of variable (numeric, categorical or string)
- Values allowed
- Other special requirements (for example: "must be unique") and branching options (for example: "if response is 1, go to question 8")

The coding sheet can be created in Excel, Word (using tables) or any other format that can use tables.

Sample coding sheet

Variable name	*Description*	*Type*	*Values allowed*	*Other requirements*
idnum	ID number	Numeric	1-500	Must enter. Must be unique.
sex	Sex	Categorical	1=male 2=female	
age	Age in years	Numeric	6-18	

Unique identifiers

Unique identifiers are used to refer to specific records in your database without using identifying information such as names. The unique identifier is usually a number assigned by the researcher. The identifying variable should be the first or one of the first variables to appear in your database. It is difficult to manage your data without unique identifiers.

Usually only one variable is used as the identifier, but it may be useful to use a composite variable, for example *School ID + Student ID*. In EpiData, this can be done using check files. See the section on "Composite unique identifiers" later in this manual.

Confidentiality/anonymity

The database used for data entry and analysis should not contain any identifying information such as names and addresses. Names and contact information for research participants should be kept in a separate document (electronic or paper) along with each individual's ID number in the database, so that participants can be contacted about their survey information if necessary.

Categorical variables

Categorical variables are variables which have a fixed set of possible responses, such as sex, country of birth or blood type. It should only be possible to choose one correct response. Categorical variables are commonly represented in a database by a number, for example:

Sex: 1=male, 2=female

ABO group: 1=O, 2=A, 3=B, 4=AB

Researchers are sometimes tempted to represent these values with "meaningful" text strings, for example:

Sex: m=male, f=female

ABO group: o=O, a=A, b=B, ab=AB

One problem with this approach is that statistical packages cannot analyse string variables and the strings will need to be recoded to numeric values prior to analysing your data. Both EpiData and statistical packages like Stata allow you to assign value labels to categorical variables and display these labels during data entry or analysis.

One advantage of representing categorical variables with numbers is that skilled data entry operators can use the numeric keypad on the keyboard to enter numbers, which is much faster than typing letters.

As numeric values may not be intuitive or easy to remember, it is essential to use a coding sheet. It's also helpful to print these numeric values, or codes, on your questionnaire, for example:

```
Sex:
1. Male
2. Female
```

or

```
Sex:
1[ ] Male
2[ ] Female
```

This can greatly reduce the number of data entry errors.

BE CONSISTENT across the database. Don't use "1=male, 2=female" in one section and the opposite in another section.

A special kind of categorical variable is often called ***binary*** or ***Boolean***. Binary means that only two responses are possible, and often refers to *yes*/*no* responses (similar ones are *on*/*off* or *have*/*don't have*).

In computing, mathematics and statistics, binary outcome variables are conventionally represented as 1 or 0. Statistical programs such as Stata recognise this convention when analysing statistical variables. An outcome variable is the main variable of interest, for example presence or absence of a disease. The category of interest would be coded 1, so in this example we would code 1=disease present, 0=disease absent.

Binary exposure variables should also be coded this way: 1=exposed, 0=unexposed.

Single vs multiple options

When offering respondents a list of possible options, you must state clearly on the questionnaire whether exactly one option must be selected, or whether more than one can be selected. These two scenarios are treated in different ways in the database. A single-option question is treated as a single variable in the database.

Example:

> Do you use long-acting insulin? (tick one only)
>
> No
>
> Yes
>
> Unknown

Appears on database as:

```
INSULINL  Do you use long-acting insulin?     #
```

If multiple selections are allowed, then each option is treated as a separate binary variable.

Example:

> Which health practitioners are involved in managing your diabetes? (tick as many as apply)
>
> Endocrinologist

GP

Nurse

Dietician

Other

(Please specify) _____

Appears on database as:

```
Which health practitioners are involved in managing your diabetes? (tick
as many as apply)
ENDO       Endocrinologist        #
GP         GP                     #
NURSE      Nurse                  #
DIET       Dietician              #
OTHER      Other                  #
OSPEC     Specify other           _____________________________
```

Continuous and discrete variables

Continuous and discrete variables contain numbers only, for example height or weight (continuous) or number of people in a household (discrete). You will often have reasonable limits or ranges for continuous variables, for example an Apgar score must be an integer between 0 and 10. When setting up a database you can specify these limits to reduce the chance of data entry error.

Many measurements come in units such as days, grams or millilitres. Decide on the most appropriate unit (grams or kilograms? minutes or hours?) and specify this unit on the coding sheet, questionnaire and data entry form. All entries for one variable must use the same unit. Don't mix up (for example) grams and kilograms or months and years. If this happens, meaningful statistical analysis will be impossible.

Date variables

Date formats vary around the world. Both day-month-year and year-month-day are logical date formats that are easily understood. Use a date format that will be understood by everybody working with your data, and keep it consistent throughout the database.

EpiData allows you to specify the date format for your database. Some other programs commonly used for data entry, including Excel, use the date format set by the operating system, which can be changed using the operating system's control panel. It might use American-style dates (month-day-year) by default. If data is being entered in Excel, check this setting in the control panel before entering dates. You should also check this setting if

the computer is used by other people or is taken away for repairs. If data is entered on multiple computers, check that the setting is the same on all of them.

Wide and long data

Some of your data may include repeated measurements. For example, a child's height and weight may be measured at regular time intervals. There are two different ways repeated data can be represented: as “wide” data or as “long” data. This difference is illustrated with a simple example.

Wide example:

ID

Systolic 1

Diastolic 1

Systolic 2

Diastolic 2

Systolic 3

Diastolic 3

Other variables that only need to be collected once, like date of birth or sex

This system is known as “wide” because in a table or spreadsheet it will appear like this:

ID	***Sys1***	***Dia1***	***Sys2***	***Dia2***	***Sys3***	***Dia3***	***DOB***	***Sex***
1	110	60	105	65	120	80	2/4/1994	2

Table 1: Example of wide data format

Long example:

ID

Observation number

Systolic

Diastolic

This system is known as “long” because in a table or spreadsheet it will appear like this:

ID	*Observation*	*Systolic*	*Diastolic*
1	1	110	60
1	2	105	65
1	3	120	80

Table 2: Example of long data format

Wide tables result in fewer records with more variables; long tables result in more records with fewer variables.

Wide tables work well if each subject has data collected at the same time point, eg everyone has blood pressure taken at first survey date and two subsequent followups. In example 1 above, each person is expected to have three blood pressure readings taken.

Long tables work well if exactly the same variables are collected at each time point. In example 2 above, people may have two, three, four or more blood pressure readings taken over the course of the study. However, it would be pointless to record their sex or date of birth every time, as these variables should not change. Such variables might be stored in a separate file or table.

String variables

String variables are variables that can contain text and numbers. As it is difficult to perform mathematical or statistical operations on strings, they should not be used if a numeric variable can be used instead. You might, for instance, have a question that asks for country of birth. The paper questionnaire can have a space for the country to be filled in, but before the questionnaires are entered in the database, you can give each country a numeric code which is entered in the database. String variables usually contain data that is too complicated to categorise, such as long comments.

Many programs have limitations on the length of string variables. In EpiData, a string variable can have a maximum length of up to 80 characters (though you can specify a

lower maximum). This limitation can be overcome by using more than one variable to represent the string. For example, if you allow space for an 800-character comment, you could use 10 string variables, comment1, comment2, ... comment10.

Your coding sheet should have guidelines for string variables as well as categorical and continuous ones. You should decide whether text should use lowercase or sentence case, and whether spelling errors on the form should be corrected during data entry or entered as they are on the form.

Missing data

Missing data is represented by special codes to indicate that something is missing. We do this so that we can tell between a response that was really missing from the questionnaire and one that has been accidentally skipped during data entry. All fields should have data in them unless they are skipped because of a conditional jump. For example, if a person is male, then he shouldn't answer a question on how many times he has been pregnant.

You can set up your database so that each field must have a value. It will only be possible for these fields to be blank if they are skipped. Some variables, such as ID numbers and eligibility criteria, should never be missing.

For numeric variables, missing data is conventionally represented by "9" for one-digit fields, "99" for two-digit fields, and so on. The missing code must never be a valid response for that variable, so if someone's age could be between 1 and 9, "99", and not "9", should be used as a missing code. Alternatively you may prefer to nominate a large number, say 9999, as the missing code for all your numeric variables.

Depending on what kind of analysis you plan to do, you might find it useful to distinguish between different kinds of missing value, such as: missing because not applicable; missing because the response does not make sense (but this should be queried first, see below); or missing because nothing was written down. You can use different numeric codes for these different circumstances.

Queries

Sometimes a response written on a form will be illegible, illogical, impossible or inconsistent with other responses. Some examples:

1. Sex: Male (if Male, go to question 3)

2. Have you been pregnant? Yes

On a questionnaire for children:

Date of birth: (more than 20 years ago)

Your database should allow “query codes” to indicate that a response has been queried and is being followed up. We often use “8” (or “88”, “888”, etc.) to indicate a query. When the data entry operator finds a strange response, a query code should be entered and the offending questionnaire set aside with a Post-It note or similar item to indicate the query. The researchers can contact the respondent for clarification, or otherwise decide what response should be entered. This official response should be entered in the database and also written on the questionnaire.

Minimising errors

Your database should be set up so that errors in data entry are minimised and can be easily identified, but errors are inevitable. Errors can be minimised in a few different ways. You can set limits for each continuous variable so that, for example, a date of birth must fall within reasonable limits. For categorical variables you can specify the valid values and restrict the database to accept these only for a particular variable. For example, valid codes for sex are 1, 2, 8 and 9.

EpiData and other data entry programs can be set up with these limits. You can also check that data falls within these limits using analysis software like Stata. Stata is also capable of identifying inconsistencies involving more than one variable, like the pregnant men in the previous section.

Cleaning data

Cleaning data involves identifying queries or errors and fixing them. This step requires a data analysis program such as Stata. After problems have been identified by the analysis program, you should go back and compare them to the written questionnaires. Some typing errors will be obvious and just need to be corrected in the database. Other errors will require some interpretation. A man who has been pregnant is clearly an error, but where is the error? Should the respondent have answered the “Sex” question as “Female” instead of “Male”, or skipped the “Have you been pregnant” question? Responses like these can be clarified by contacting the respondent or looking for clues elsewhere in the questionnaire. If it is not possible to decide on a meaningful response, these variables should be recorded as missing.

Data can be corrected either in the original database and re-exported for use in Stata, or the original database can be kept and corrections made in Stata.

EpiData

EpiData is a Windows program for creating data entry forms and entering data. It can be downloaded free of charge from http://www.epidata.dk/ .

Why use EpiData?

EpiData is a small and portable program. The setup file is less than 1 MB in size. The program and all of its configuration files are contained in a single directory which can be located anywhere on your computer. EpiData does not interfere with your computer's system settings.

EpiData is entirely free of charge.

EpiData uses the same file format as Epi Info. It can also import data from plain text, Stata and dBase, and export to plain text, dBase, Excel, Stata, SPSS and SAS formats.

Limitations of EpiData

Because EpiData aims to be small, portable and simple, it has limitations related to its simplicity.

Questionnaire files cannot exceed 999 lines of text. If you have a large database that exceeds this length, you can break it up into different files.

Unlike more complex database systems, EpiData is not designed to operate over a multi-user network. Only one person can modify a file at any one time.

A string variable cannot have more than 80 characters: if you anticipate long text comments, you can break them up into multiple comment variables.

EpiData has some commands to enable relationships between different database files, but it is not truly relational in the way that SQL and Access are relational.

Where to get EpiData

Installing EpiData

The download page for EpiData is http://www.epidata.dk/download.php . Note that the first section of this page relates to the EpiData analysis program. To download the data entry program, scroll down the page or click "EpiData Entry".

Click the "Complete Setup" link. If you are using Internet Explorer, you will be asked whether you want to open or save the file; select "Save" and specify a convenient location on your hard drive, for example "Desktop". When the download is complete, you should

have a file called "`setup_epidata.exe`". Double-click on this file to install EpiData. The setup program will install EpiData itself as well as documentation. The default directory used by the setup program is "`C:\Program Files\EpiData`".

Updating EpiData

EpiData is regularly updated with bug fixes and new features. Announcements about updates are sent to the EpiData announcement list.[1] If you have previously installed EpiData and want to update the program, go to the download web page (as above) and select "Zip – exe only". You will download a file called "`epidata.zip`" which is a ZIP archive containing the program but no help files. To update EpiData, unzip this archive and move the new "`epidata.exe`" file to EpiData's program directory, which by default is "`C:\Program Files\EpiData`".

Starting EpiData

If EpiData was installed using "`setup_epidata.exe`", there should be a link to EpiData in the "Programs" section of the Windows Start menu, and an EpiData icon on the Windows desktop.

You can also start EpiData by selecting "Run…" in the Start menu and typing the location of the EpiData program, which by default is "`C:\Program Files\EpiData\epidata.exe`".

When EpiData is started, a window pops up with a brief introduction to the program. Click the **Close** button to start working.

The EpiData interface

When you start EpiData, at the top of the screen you will see the:

- Menu bar: all EpiData commands are available from this menu.
- Workflow toolbar with the most essential commands in logical order.

- Toolbar with common commands such as create/open/save file, print, etc. (the "editor toolbar")

[1] Visit http://www.epidata.dk/lists.htm for instructions on joining the announcement list.

Creating the data form

A questionnaire file contains field names, field definitions and other text. It is used to generate a data entry form, which has the extension *.qes*. There are two ways you can create the questionnaire file: you can type it up from scratch, or if you already have the questionnaire in electronic format, you can use the text from that file as a basis for the EpiData questionnaire.

Starting from scratch

To create a new questionnaire from scratch, do one of the following:

- From the workflow toolbar, select *1. Define Data → New .QES file*.
- From the menu bar, select *File → New*.

These will open EpiData's editor with an empty questionnaire file. Start typing the questionnaire. Save the file by using the menu *File → Save* or clicking the floppy disk icon on the editor toolbar.

Using an existing file

You need to make sure that the file you will be importing is in plain text format, and that the file name ends with *.qes*. If the questionnaire is in Microsoft Word format, follow these steps to save it as a *.qes* text file:

1. In Word, select *File → Save As…* from the menu bar.
2. In the "Save as type" box, select "Text Only". The file extension is now *.txt*.
3. You can change the name of the file if you like.
4. Click the **Save** button.
5. Close the file in Word.
6. Open the file in EpiData by selecting *File → Open* from the menu bar. Make sure that "All" is selected in the "Files of type" box.
7. Select *File → Save As…* from the menu bar.
8. In the "File name" box, change the file extension from *txt* to *qes*.

You can now edit the questionnaire file as needed in EpiData's editor.

Format of the questionnaire file

The questionnaire file contains lines of text that either define a data entry field or are ignored by the database. A line is recognised as a field line if it contains special characters used to represent a data entry field. You can have more than one field on one line.

Field type	*Examples*
Numeric	#### (integer up to four digits) ##.# (up to three digits including one decimal place)
Automatically generated ID number (you do not enter this number yourself)	<IDNUM>
Text/string	_____ (underscores; you can specify a maximum of 80 characters in a text field)
Date	<dd/mm/yyyy>, <yyyy/mm/dd> or <mm/dd/yyyy> – all dates must use the same order
Automatically enter today's date (you do not enter this date yourself)	<today-dmy>

These are the most common field types used. Others are available: search for "Field types" in the EpiData help system for more information.

EpiData has a Boolean (binary) variable type which is restricted to 0 and 1. It may seem appropriate to use this for binary variables. However, we recommend allowing missing and query codes in our variables, including binary ones, so it is better to use a single-digit numeric variable rather than a Boolean variable.

Variable names can be up to 10 characters long, can contain only letters and numbers, and must start with a letter.

There are two ways to specify the variable name. You can change this behaviour in the menu *File → Options → Create data file → How to generate field names*.

- **First word in question is field name:** The first word on the line (up to 10 characters before the first space) becomes the variable name.
- **Automatic field names:** EpiData tries to guess the variable name based on the words in the question. (See the help system for more information. This method requires more work.)

Each line containing a variable should follow this format:

```
(Field name)    (Field label)     (Field definition)
```

Variable labels describe individual variables. EpiData automatically assigns variable labels based on the text that appears before the field definition. If you are using the "First word in question is field name" option, then all the text to the left of the field definition and to the right of the field name is used as the variable label. If you are using the

"Automatic field naming" option, then all the text to the of the field definition is used as the variable label.

Lines that do not contain a field definition are ignored by EpiData; they can be used for comments, headings or text to assist the data entry process.

Save the questionnaire file by pressing Ctrl+S, using the menu *File* → *Save* or clicking the Save button  . If the file has not previously been saved, you will be asked to specify the file's location and name.

Helping you create field definitions

As you continue using EpiData you will find it easier to type in field definitions yourself, but when you are starting out you may need help remembering them. The ***field pick list*** and the ***code writer*** can help.

Field pick list

To use the pick list, open the *.qes* file that you want to edit, position the cursor at the place where you want to enter a field definition, and do one of the following:

- Press Ctrl+Q on the keyboard
- Use the menu *Edit* → *Field Pick List*
- Click the "Field pick list" button  on the editor toolbar

Now you can choose the type of field (number, text, date or other) and field parameters such as the length and format of the field.

To close the pick list, click on the Close button  on the top right corner of the pick list.

Code writer

The code writer automatically completes field definitions when you start typing field characters. For example, typing # will create a numeric field, typing _ will create a text field, and so on.

To turn the code writer on or off, do one of the following:

- Press Ctrl+W on the keyboard
- Use the menu *Edit* → *Code Writer*
- Click the "Code writer" button  on the editor toolbar

When you begin typing one of the recognised characters, you will be prompted for the length of the field (for numeric, text and IDNUM fields) or the field will automatically be inserted (for date and Boolean fields).

NOTE: The code writer and the field pick list cannot be used at the same time. Turning on one of these items will turn off the other if it is on.

Formatting issues

You will have to take care that the text of your questions and comments does not include any characters that have special meaning to EpiData, such as field definition characters or "@" (used to force a tab character). A common problem is the "<" (less than) character which indicates the start of a date, Boolean or uppercase text field. If the text in your questionnaire contains this character, replace it with the text "less than" or "LT".

Underscores ("_") will be seen as text fields.

EpiData supports a maximum of 80 characters preceding a field on each line. If you type a field name, question text and field definition and the line is longer than 80 characters, the text before the field definition will be truncated so that the data entry field starts at the 81st character. To overcome this, you can split the question over multiple lines. Here are some examples of how to do this:

Question on a line separate to the field name and field definition. This means that there will be no variable label.

```
8. What is the total number of units of short-acting insulin per day?
shortunt   ###
```

Question on separate line. Brief description added to line with field name and field definition.

```
8. What is the total number of units of short-acting insulin per day?
Shortunt  units short-acting insulin per day        ###
```

Split questions over multiple lines.

```
          8. What is the total number of
shortunt  units of short-acting insulin per day?    ###
```

The most appropriate solution will depend on how long the question is and whether it can be condensed into its essential parts. Note that in the first example above, there will be no automatic variable label; in the two later examples, a short variable label will be automatically assigned.

It is helpful, but not compulsory, to keep the questionnaire file tidy. One useful command is "Align Fields". This will attempt to line up the field definitions in the file so that each field definition starts from the same position.

1. Place the cursor on a line which has the field definition positioned where you would like it. This works best if the field definition is near the end of the line but does *not* go past the 80th character of the line.
2. Use the menu *Edit* → *Align Fields*.

For example:

If the cursor is positioned on the first line, the following form

```
v1 A small text     ####
v2 Other text <A   >  v3 ###.#
v4 Text ###
```

becomes

```
v1     A small text ####
v2       Other text <A   >  v3 ###.#
v4             Text ###
```

Previewing the data form

When editing the questionnaire file, you can see what the resulting data entry form will look like with the *Preview data form* command. You can run this command in the following ways:

- Press Ctrl+T on the keyboard
- Use the menu *Data File* → *Preview Data Form*
- Click the "Preview data form" button  on the editor toolbar

The data form preview will appear in a new screen. You can switch between the file editor and the preview by clicking the buttons at the bottom left of the screen. If you change the questionnaire file, you must run the preview command again to see the changes in the preview window.

IMPORTANT: You do *not* enter data in the preview window.

Creating the data entry file

Now we create the file where data is actually entered. First you must close the questionnaire (*.qes*) file if it is open by typing Ctrl+F4 or F10 on the keyboard, selecting *File* → *Close* from the menu or clicking the Close button  .

WARNING: Do not follow these steps if the data entry file contains data that has already been entered. If you need to change the questionnaire file, follow the instructions under "Modifying the database".

To create the data entry file, do one of the following:

- On the workflow toolbar, click *2. Make Data File*

- From the menu bar, select *Data in/out* → *New Data File*

A dialogue box will appear. On the first line, next to "Enter name of .QES file", enter the name of your questionnaire file. This line should already contain the name of the file that you have just been working on. If you need to use a different file, click the folder icon next to this line and you will see another dialogue box where you can select your questionnaire file.

The second line on the dialogue, next to "Enter name of data file", will automatically contain the name of the data entry file (with a *.rec* extension) that corresponds to your questionnaire file. **You should use the file name that EpiData automatically selects.** This makes it easier to remember which questionnaire file belongs to which data entry file.

Press **OK**. You will be asked to enter a brief ***file label*** for the database. It is a good idea to enter a meaningful description here because if you later import the data into Stata the data file label will also be imported. You can also add or change the label later using the menu *Tools* → *Edit File Label*.

Press **OK** again. EpiData will confirm if the database is successfully created. If there is a problem, an error message will tell you what went wrong.

Check files – checking validity of data

A *check file* contains commands that EpiData uses during data entry to check the validity of the data being entered. A check file is not essential, but highly recommended. Mistakes are always possible during data entry, and check files can help to minimise these.

Check files are plain text files with special commands. They can be created and edited in one of two ways: using an interactive interface, or editing the file manually. Names of check files end with the extension *.chk* and the first part of the name must be the same as the corresponding data file. For example, a check file `questions.chk` can only be used with the data file `questions.rec` in the same directory.

Types of check available in EpiData

The following types of check are available in EpiData:

Unique: Each record must have a unique value for this field. This check is often used for identification fields.

Must enter: This field must have a value; it cannot be missing. (Use a *missing code* if the value really is missing. See the discussion under "Missing data".)

Range/Legal values: Limit the values that can be entered in this field. For example:

Sex: 1=male, 2=female, 8=query, 9=missing, so only 1, 2, 8 or 9 can be entered.

Weight: 20-40, 88=query, 99=missing.

Date of birth: 01/01/1990 to 31/12/1992, 8/8/8888=query, 9/9/9999=missing.

These can be specified in EpiData using a combination of the "Range" and "Legal" commands.

Jumps: By default, data entry progresses from the first field to the last field in the form, one field at a time. Jumps are used to move to different fields. They are most useful for conditional jumps, for example "if response to question 8 is No, go to question 10". It is possible to jump to an earlier field or even to the current field, but these should be avoided.

A word of caution when using Boolean fields: EpiData's special Boolean field type is designed so that entering "1" is the same as entering "Y", and entering "0" is the same as entering "N". When using check commands on Boolean fields, you must specify the values as "Y" or "N", not "1" or "0".

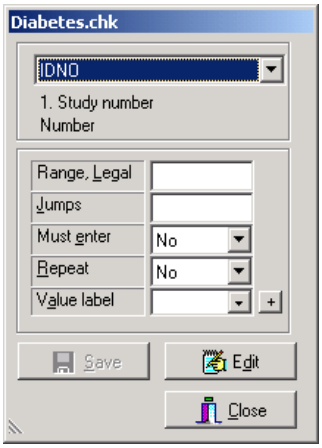
The **Repeat** command means that when you start a new record, any field which is set to repeat will already contain the value that was entered in that field in the previous record. This is useful for example when you are entering a number of questionnaires from the same school and you don't want to have to type the same School ID at the top of each questionnaire. This command helps prevent one kind of data entry error, where entering the same value in the same field for a number of questionnaires can be repetitive and you might occasionally enter the wrong value. However, the repeat command also has a danger: when moving on to a different school, the data entry operator must remember to type in the new School ID for the first record in that school.

The interactive check interface

The easy way to start using checks is to use the interactive check interface. Do one of the following:

- On the workflow toolbar, click *3. Checks*
- From the menu bar, select *Checks* → *Add / Revise*

You will see a preview of the data form along with a floating dialogue box where you can create or modify checks for each field in the form. The top part of the dialogue box contains a drop-down list with the names of all the fields in the current database, followed by two lines with the current field's label and type. The next part of the dialogue box contains boxes where you can enter the most frequently used checks: value ranges and legal values, jumps, must-enter, repeat and value labels. The buttons at the



bottom of the dialogue box are: **Save** the check file (and keep working); **Edit** the checks for this field (open a check file editor for the current field) and **Close** the check file.

To work with checks interactively, you must first select the field whose checks you are editing. You can do this by clicking on the field in the preview form, or by selecting the field name from the drop-down list at the top of the dialogue box.

Common check commands

The following commands can easily be created using the dialogue box.

Command	*Description*	*Example*
Range, Legal	This is a combination of the "Range" and "Legal" commands. In the dialogue box, enter the range separated by a hyphen, then add other legal values separated by commas.	1-10,88,99
Jumps	For example, "if response is 0, go to question 7 ; if response is 1, go to question 8".	0>q7,1>q8 where q7 and q8 are the field names
Must enter	If yes, then you will not be able to move to the next field if this field is blank.	Yes or No
Repeat	Automatically enter the same value entered in this field in the previous record.	Yes or No

Value labels

Value labels are used to indicate the meaning of codes for categorical variables. They appear in label blocks at the top of the check file. The label block takes the following form:

```
LABELBLOCK
    LABEL labelname1
          {label specifications}
    END
    LABEL labelname2
          {label specifications}
    END
    {...}
END
```

For example, a label block for sex, where 1=male and 2=female, would look like this:

```
LABEL sex_label
    1     male
    2     female
END
```

If a label contains spaces, it must be surrounded by double quotes (" and ").

```
LABEL numcigs
```

```
    1     "1-10 cigarettes/day"
    2     "11-30 cigarettes/day"
    3     "31-50 cigarettes/day"
    4     ">50 cigarettes/day"
END
```

Using value labels has the added effect of defining the field's legal values: the values listed in the label block become the legal values for that field.

Value labels can be added and edited using the interactive check dialogue. Next to the text "**Value label**" is a drop-down list containing the file's current value labels. To add a new label for that field, click on the **+** sign next to the drop-down list. An editor window will appear with the start and end of the label block filled in. Type the label descriptions in the format above, one on each line. Click "**Accept and Close**" when you are done to save your changes and close the editor.

A label can be used for more than one field. To use a label that has already been created, just select it from the drop-down list.

To modify an existing label, first select a field containing the label, then click on the + sign next to the list. An editor window will appear with the label block. Make your changes and click "**Accept and Close**".

Key unique

The "KEY UNIQUE" command is used to specify that a field is a unique identifier. This command does not appear in the interactive check dialogue. To add this command, select the unique ID field and click on the "Edit" button in the dialogue box (or edit the check file manually, see below). Type "KEY UNIQUE" on the line following the field name. For example

```
IDNUM
    KEY UNIQUE
END
```

See "List of check commands" in EpiData's help system for a list of all check commands, and for instructions on how the above commands should be entered using the check file editor.

Editing check files manually (*advanced*)

The interactive check dialogue can be used to edit checks and labels for specific fields. Check commands can also be applied to the entire data file. You must edit the check file directly to use these commands.

EpiData provides a text editor for editing check files, similar to the questionnaire file editor. To open a file in this editor, use the menu *File* → *Open*, select "EpiData check file" from the "Files of type" drop-down list, and select the check file. You can now edit the whole check file all at once rather than having to select fields in the interactive dialogue.

Format of check files

A small check file might look something like this:

```
BEFORE FILE
    CONFIRM
    COLOR BACKGROUND WHITE
END

* a comment begins with an asterisk and is ignored

IDNUM
    KEY UNIQUE
    MUSTENTER
    RANGE 1 1000
END

BRANCH
    MUSTENTER
    RANGE 1 3
    LEGAL
          8
          9
    END
END
```

As you can see, commands are divided into blocks: one block for commands that apply to the whole file, and one block for each field. Each block begins with either "BEFORE FILE" for the general commands, or the field name for field commands, and ends with "END". If you enter check commands using the interactive check dialogue, commands will be saved in upper case and commands within blocks will be indented. Commands can be either upper or lower case and do not need to be indented, but these conventions make check files easier to read.

The first block of commands sets the data entry method to "Confirm" (see below) and makes the background of the form appear white. The next block applies to the IDNUM field and says that IDNUM is a unique identifier, must be entered and can take a value between 1-1000. The next block applies to the BRANCH field and says that BRANCH must be entered and can take a value between 1-3 or 8 or 9.

Some useful tricks using checks

Changing the appearance of the data entry form

You can set the default colour and font for your personal EpiData environment (see "Options/customisation" below).

You can also set the colour for individual data files using the COLOR check commands and common colour names. (Note the American spelling.) These should appear in the BEFORE FILE block at the top of the check file.

```
COLOR DATA textcolour [backgroundcolour [highlightcolour]]
```

This changes the colour of the data entry fields. Colours must be entered in this order; background and highlight colour are optional. (Do not enter the square brackets [and].) The first colour is for the text in the field, the second is for the background of the field and the third is the background colour of the field where the cursor is positioned.

```
COLOR QUESTION textcolour [backgroundcolour]
```

This changes the colour of the question text. The optional second colour is for the background underneath the question text.

```
COLOR BACKGROUND backgroundcolour
```

This changes the colour of the form background.

Automatic values

Sometimes you will want a field to contain a value calculated from the values entered in previous fields. An example is where you might enter a date of birth, then have a separate field for age which is calculated by the formula:

$$\frac{today - dateofbirth}{365.25}$$

You want the age calculated automatically by EpiData, not typed in by the data entry operator. You can use check commands to do this, but you should add the NOENTER command so that data cannot be entered in the automatic field.

The check command for the age example above would look like this:

```
DOB
    MUSTENTER
    AFTER ENTRY
          LET AGE = (Today - DOB)/365.25
    END
END

AGE
    NOENTER
END
```

Note that "Today" is a function that returns today's date.

Composite unique identifiers

Usually a single field will be used as the unique identifier, but sometimes you may need to use a combination of more than one field as the unique ID. In EpiData, you cannot specify more than one unique ID field, but you can get around this limitation by creating a new field that is automatically filled in with a value calculated from the values of other fields.

In this example, the combination of "idno" and "week" must be unique. Firstly in the QES file create a new field called "recid":

```
idno            Study Number      ##
week            Week number       #
recid                             ##.#
dov       Date of Visit     <dd/mm/yyyy>
....
```

Then include the following code in the check file:

```
week
    RANGE 1 5
    MUSTENTER
    AFTER ENTRY
          LET recid=idno+week/10
    END
END

recid
    KEY UNIQUE 1
    NOENTER
END
```

The resulting data will look something like this:

idno	*week*	*recid*
1	1	1.1
1	2	1.2
2	1	2.1
2	2	2.2

Entering data

Once you have your data file and check file, you can start entering data. Open the data entry file:

- On the workflow toolbar, click *4. Enter Data*
- From the menu bar, select *Data in/out* → *Enter Data*

The cursor is positioned in the first field. To move to the next field after entering a value, use the "Tab" or "Enter" keys on the keyboard. If the field has a "must enter" check and you try to move to a different field when the field is empty, an error message will remind you to type something.

Don't use the mouse to navigate around the form, as this bypasses the "must enter" checks.

By default, if you type the maximum number of characters in a field, the cursor will automatically move to the next field. This can be confusing because you might lose your

place on the form and continue typing characters that belong in the previous field. We recommend that you use the "CONFIRM" setting so that ENTER must be hit after every field. Add the following commands to the top of your check file:

```
BEFORE FILE
  CONFIRM
END
```

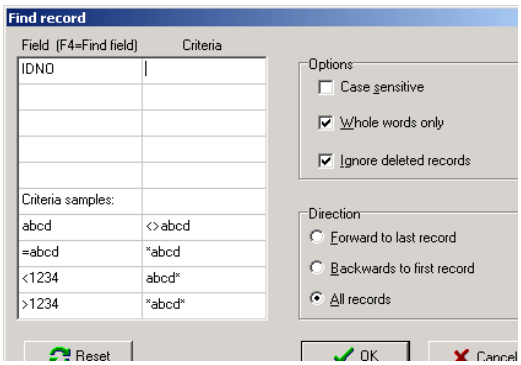
When CONFIRM is used, the cursor will not automatically move to the next field when the current field's maximum length is reached; you will need to press "Tab" or "Enter" at the end of every field.

Data entry progresses along each field in sequence, unless there are jumps or automatic fields.

When you have entered data in the last field and press Tab or Enter, a dialogue box will ask "Save record to disk?" Click Yes to save the record and start a new empty record, or No to cancel and stay in the current record.

Editing data

The data entry interface can be used for searching, modifying and removing existing data as well as entering new data. This can be used for correcting data entry errors or queries. You need to be able to identify the record you are looking for: this is why unique identifiers are essential.



Open the data entry form as described in the previous section; you should start with a blank record. Open the "Find" dialogue by pressing Ctrl+F or using the menu *Goto → Find Record*. The left-hand side of the dialogue box contains one column for field names and one column for search criteria. The right-hand side contains various options. Make sure that the field name that you are searching for is listed; if it isn't you can replace it with the correct field name. In the criteria box next to the field name, type the value you are searching for. If you list more than one field name, then the search will only identify records that match all the criteria.

Press OK to start the search. If EpiData finds a record that matches the criteria, it will jump to that record. You can use the "Find again" command (press F3 or use the menu *Goto → Find Again*) to search for the next record that matches. If there are no matches, an error message will appear.

When you have found the record you need to change, you can jump to the field you need to change by pressing F4 and selecting from a list of all fields. Make the corrections to the data, then press Ctrl+N to move to a new blank record. You will be asked whether you want to save the record to disk—click "Yes" to save.

Entire records can be deleted. As this is a drastic measure, EpiData has some safeguards to make it difficult to delete data by accident. You do not delete records immediately but mark them for later deletion. To mark a record for deletion, search for the record, then click on the red cross at the bottom of the data entry window or use the menu *Goto → Delete record*. "**DEL**" will appear at the bottom of the window. Press Ctrl+N to save this record and move to a blank form.

The record is now marked for deletion. If you marked it by mistake, you can remove the mark by clicking on the red cross again or using the menu *Goto → Undelete record*. To actually delete it permanently, close the data entry form, then use the menu *Tools → Pack Data File* and select the file with the deleted records. You will be warned, *twice*, that this action will permanently remove marked records. If you are sure that this is what you want to do, click the **OK** button twice and the marked records will be permanently deleted.

Modifying the database

Occasionally you may need to add or change one or more fields in the data file. For instance, you may need to add a new variable to the data file. You can modify the data entry file after you have started entering data, but if you are not careful you can accidentally lose a lot of data.

Make the changes to the questionnaire file. When you are done, ***do not press the "Make Data File" button***. Doing this will cause EpiData to erase the existing data file and replace it with the new empty data file.

Instead, use *Tools → Revise Data File*. This command will modify the existing data entry file based on the new questionnaire file.

WARNING: If you remove an existing variable, all the data for this variable will be lost. Changing the name of an existing variable will have the same effect. You will be warned if you try to "Revise Data File" and any information will be lost in the process.

Documentation tools

EpiData provides some commands to view your data and summarise database characteristics such as the number of fields and records, whether checks are applied, and field definitions. These commands can be found under the following menus:

- On the workflow toolbar, click 5. *Document*

- From the menu bar, select *Document*

Exporting data

You can export your EpiData database to the following formats:

- plain text
- dBase
- Excel
- Stata
- SPSS
- SAS

Begin the export process by clicking one of the following:

- On the workflow toolbar, click *6. Export Data*
- From the menu bar, select *Data in/out* → *Export*

Choose the export format, then the data file that you want to export. A dialogue box will appear where you can choose which records and fields to export. Depending on the export format, this dialogue box may also have a second "Options" tab with additional options.

Plain text is the most portable format as these files can be imported into all database, spreadsheet and analysis programs. However, if you know that you are going to use one of the more specific programs, then use that format instead.

Plain text files will contain each record on one line, with variables separated by a special character. You can specify this character in the "Options" tab of the text export dialogue. The default separator is the semicolon (";") but you can choose to use a comma (","), tab character or any other character you like. Semicolons, commas and tabs are the characters that are conventionally used for text-delimited files. Make a note of which separator you use so that when you import the file into another program you can specify this character.

Excel files created by EpiData are compatible with Excel version 2.1, and can be read by version 2.1 or later of Excel.

When exporting Stata datasets (*.dta* files), you can select a Stata version between 4 and 8 in the "Options" tab of the Stata export dialogue. Version 8 is compatible with Stata version 8 or 9. You can also specify whether the field names in the exported file are in uppercase, lowercase or as they appear in EpiData.

Options/customisation

You can change various options using the *File* → *Options* menu. This command will bring up a dialogue box with different options grouped in different tabs. The first two tabs allow you to change the default font and background colour for the editor (for editing questionnaires and check files) and data form respectively.

It is a good idea to use a ***monospace*** (also known as ***fixed-width***) font for database setup. In monospace fonts, each character takes up exactly the same amount of horizontal space. This means that it is easy to see how long a word or line is. Monospace fonts include "Courier New" and "Lucida Console".

The third option tab allows you to change the method for identifying field names when creating a data file. See the section on "Format of the questionnaire file" for an explanation of these options.

Getting help

EpiData help system

Help is available from within EpiData, under the *Help* menu.

EpiData home page

Enter http://www.epidata.dk/ in your web browser or, in EpiData, use the menu *Help* → *EpiData Homepage*.

http://www.epidata.dk/support.htm explains some other ways of getting help with EpiData.

Mailing lists

There are two mailing lists available. An overview of the lists, with instructions on joining them, is at http://www.epidata.dk/lists.htm.

- EpiData news: official announcements from the EpiData Association
- Discussions: questions and solutions

Other resources

EpiData documentation: http://www.epidata.dk/documentation.php

International Union Against TB and Lung Disease:
http://www.tbrieder.org/epidata/epidata_course/epidata_course.pdf

Svend Juul, "Take good care of your data". Includes guidelines on questionnaire layout.
http://www.epidata.dk/php/downloadc.php?file=takecare.pdf

Frequently used commands

Command	*Menu*	*Workflow toolbar*	*Keyboard*	*Button*
Create new questionnaire	File → New	1. Define Data → New .QES file	Ctrl+N	
Save file	File → Save		Ctrl+S	
Open file	File → Open		Ctrl+O	
Options	File → Options			
Field pick list	Edit → Field Pick List		Ctrl+Q	
Code writer	Edit → Code Writer		Ctrl+W	
Close pick list or code writer				
Close current file	File → Close		Ctrl+F4 or F10	
Preview data form	Data File → Preview Data Form		Ctrl+T	
Generate data file from questionnaire file	Data in/out → New Data File	2. Make Data File		
Edit checks interactively	Checks → Add / Revise	3. Checks		
Data entry form	Data in/out → Enter Data	4. Enter Data		
Documentation	Document	5. Document		
Export	Data in/out → Export	6. Export Data		

File types

<filename>.qes	Questionnaire file
<filename>.rec	Data file
<filename>.chk	Check file