

Exercise 4: Create a composite identifier

At the end of this exercise you should be able to:

- a. Recognize the efficiency of using date components instead of dates
- b. Creating a unique identifier from several variables
- c. Automatically capturing data entry time

Sometimes we have time information but it is incomplete, such as when a patient may remember the month but not the day of disease onset. In such a case a date field cannot be used sensibly because a date needs all three date components. It becomes thus useful to utilize instead three variables for day, month, and year respectively and then reserve it for the analysis to set rules what approximation might be used if one or more date components are missing. For data entry, this can also have the advantage that some components (like the year, or even the month) are set to Repeat if records are entered serially along a time axis and several sequential records have the identical year and perhaps also month.

A date component might also be used to construct a more complex identifier if the simple one does not suffice. In these exercises using the tuberculosis microscopy laboratory we defined the unique identifier as the laboratory serial number. However, the serial number starts with 1 every calendar year and is thus unique only for a single year. By combining the year plus the serial number, it becomes unique for the laboratory at any time, and adding the laboratory identifier, we can make a unique identifier for any examinee for any year, in any laboratory.

To learn about the workload of data entry, there is a possibility to instruct EpiData to record automatically the computer time at the time of opening a new record and to add another time stamp at the completion of the record.

We will introduce here these three concepts and integrate the computer-calculated fields into a special section.

Capturing time with date components instead of with dates

Export the structure of the `a_ex03.epx` file to a new `a_ex04.epx` file and open the new file. We then start with deleting the field `regdate`, dragging down the block of variables below it, and inserting the three new integer variables `regdd`, `regmm`, and `regyy` of lengths 2, 2, and 4 respectively, for `Registration day`, `Registration month`, and `Registration year`. We make each field `Must Enter` and provide a legal range and put the respective three values 99, 99, and 9999 for not recorded into two different `ValueLabels`. Remember from Exercise 2 that we need to untick the `Show valuelabel text after ...` in the `Extended` tab. We can make the fields for day and month set to `Repeat` as these two remain identical in most laboratories for quite a few sequential records.

Making an identifier composed of 3 existing variables

On the top we section off a part that will accommodate three calculated fields: the new identifier, the start time and the ending time of data entry for the record. Drag thus all fields down on the

canvass and insert a section above them that we call `Calculated fields`. The identifier `id`, labeled `Unique identifier` is a string field of length 14 as we will give it the form `YYYY-labcode-serno`, i.e. the 4-digit year, a hyphen, the 4-digit laboratory code, a hyphen, and the 4-digit laboratory serial number. It is placed inside the section and it has to be a `No Enter` field as we want to prevent a data entry person being able to ever touch a computer-calculated field. The top of our revised data entry form may thus look now as follows:

How do we go about making the value for this identifier? It can only be composed once all three required components have been entered, i.e. after entering the `Registration year`. We thus go to that field and go to the tab `Calculate`, where we choose the bottom option and fill by selection from the drop-down fields and inserting a hyphen in between, that's all there is to it:

A little more sophisticated is the approach to make `id` a unique identifier. Intuitively, we might think that we just make this field `id` the key field, but that is not how it's done: it is each of the components that is set to be the key and EpiData will then correctly interpret the resulting combined field required to be unique. It may appear a bit counter-intuitive, but that's how it's done. Remember where to `Define Key`? Right-click the data form name or go to the menu point `Dataform`:

Tasks:

- o Add two fields for *Starting time* and *Ending time* for data entry in the section *Calculated fields*.
- o Test the data entry form in *EpiData EntryClient*