

1 Implemented data stream

Dies sind die Datenströme wie sie sich jetzt (Okt 2015) darstellen. Wir haben versucht so dicht wie möglich an den Datenerfassungsgepflogenheiten dran zu bleiben, wollen aber die zu erfassenden Informationen minimieren.

DS01 wir fangen mit der Vorbereitung der Reproduktion an. In diesem Datenstrom werden die Käfige ausgewählt, aus denen Eier zur Reproduktion genommen werden soll. Auf dem Betrieb wird in einem ersten nicht weiter dokumentiertem Schritt eine Liste der im Weiteren zu berücksichtigen Käfige erstellt. Somit gibt es diese Elemente im DS01:

1. cage number selected
2. ordinal number
3. date

What information does the form need to provide:

1. the header shows the date and location
2. one screen lists all cages in a grid with a check box each. If checked the cage will be considered active. All others will be closed plus the animal in `ANIMAL.exit_dt`
3. maybe the breeds need to be indicated through the underlying color of each cage
4. one button at the bottom executes the 'submit'. In this way updates can be handled.
5. the check boxes should reflect the selection already done through a previous 'submit'

The LO needs to do:

1. on the basis of the list of cages from a given location close the cage: for `ext_unit`, `ext_id`, `db_member` set the closing date.
2. set the exit date in `ANIMAL` to the recording date but leave the data channel open, in case we get more data after this cage selection stage.
3. Eintrag in `EVENT`?

DS02 number of eggs collected are recorded three weeks after DS01; these are recorded only for the cages selected in DS01. The data entry form should propose only these cages sorted by cage number (cages are located in the real world in ascending order). A preselect by breed should be possible. Elements for each cage are:

1. cage number
2. the list of active cages for this breed (online action, we assume something like max 30 cages/breed)

3. the cages are placed on the screen on a grid with an entry field for each cage
4. number of eggs collected in two days
5. collection date

The form needs to contain the following information:

1. chose the breed
2. the list of active cages for this breed (online action, we assume something like max 30 cages/breed)
3. the cages are placed on the screen on a grid with an entry field for each cage
4. the fields are preset with values already entered before

The LO needs to do:

1. for the remaining active cages record the number of eggs collected is inserted in table ???
2. the grid is preset with the data already entered. Fields can be overwritten. Treat the complete grid as one transaction triggered by one submit button.

DS03 number of eggs put in the incubator (2 weeks after DS03). Each egg is marked with the cage number it originated from. All eggs from one cage are put under one mesh wire cage for separation. Elements are:

1. cage number
2. ordinal number of breed's cages
3. number of eggs put in incubator

The form is basically the same as for DS02. The form needs to contain the following information:

1. chose the breed
2. the list of active cages for this breed (online action, we assume something like max 30 cages/breed)
3. the cages are placed on the screen on a grid with an entry field for each cage
4. the fields are preset with values already entered before

The LO needs to do:

1. for the remaining active cages record the number of eggs put in the incubator is inserted in table ???

2. the grid is preset with the data already entered. Fields can be overwritten. Treat the complete grid as one transaction triggered by one submit button.

DS04 number of eggs hatched for each set of eggs from a defined cage (some 15 days after loading the incubator). Elements are:

1. for each cage number
2. number of chicks hatched
3. date

The form is basically the same as for DS02 and DS03. The form needs to contain the following information:

1. chose the breed
2. the list of active cages for this breed (online action, we assume something like max 30 cages/breed)
3. the cages are placed on the screen on a grid with an entry field for each cage
4. the fields are preset with values already entered before

The LO needs to do:

1. for the remaining active cages record the number of eggs hatched is inserted in table ???
2. the grid is preset with the data already entered. Fields can be overwritten. Treat the complete grid as one transaction triggered by one submit button.

DS05 here we leave the cage number and move to individual animal identification. At sexing the chicks are tagged with wing bands. For each cage number we record:

1. for each cage number
 - (a) 1st male chick wing ID
 - (b) 2nd male chick wing ID
 - (c) ..
 - (d) 1st female chick wing ID
 - (e) 2nd female chick wing ID
 - (f) ..
2. date

The form needs to contain the following information:

1. choose the breed

2. for each (parental) cage we have one screen.
3. the cage number is the header
4. the fields in a row are: wingID, sex, wingID, sex,
5. maybe we have room for 5 ID/sex pairs per row with 5 rows
6. preset the wingID fields with the incremented last ID
7. the only data to be entered is the sex: m/f
8. if a an ID is broken, simply no sex is specified
9. the last used ID is the field with the last sex entered.

Breed abc

chicks from parental cage _

4711 _ 4712 _ 4713 _ 4714 _ 4715 _

4716 _ 4717 _ 4718 _ 4719 _ 4720 _

4721 _ 4722 _ 4723 _ 4724 _ 4725 _

4726 _ 4727 _ 4728 _ 4729 _ 4730 _

4731 _ 4732 _ 4733 _ 4734 _ 4735 _

submit

data entry with a broken wingID=4716:

chicks from parental cage 47

4711 m 4712 m 4713 f 4714 f 4715 f

4716 _ 4717 f 4718 f 4719 f 4720 f

4721 f 4722 f 4723 _ 4724 _ 4725 _

4726 _ 4727 _ 4728 _ 4729 _ 4730 _

4731 _ 4732 _ 4733 _ 4734 _ 4735 _

submit

the next form would be:

chicks from parental cage _

4723 _ 4724 _ 4725 _ 4726 _ 4727 _

4728 _ 4729 _ 4730 _ 4731 _ 4732 _

4733 _ 4734 _ 4735 _ 4736 _ 4737 _

4738 _ 4739 _ 4740 _ 4741 _ 4742 _

4743 _ 4744 _ 4745 _ 4746 _ 4747 _

submit

The load object will create new entries in ANIMAL for each chick, deriving pedigree information from the cage inhabitants.

1. for each ID a new record in ANIMAL needs to get created
2. skip those lines that have no sex specified
3. the last used ID needs to get determined on the basis of the last record with a sex specified.

DS06 records the information on the matings of the new generation and the allocation of the animals to the cages. Currently, a simple rotational scheme is used. This results in a list of animals allocated to a certain cage. The elements are:

1. cage number
2. cock ID
3. ID hen 1
4. ID hen 2
5. ..

This list will be produced for each breed either manually (as was done until now) or through some software. Thus, before DS06 can be executed, the procedure for doing the mating will have to be defined separately. The user should have the option of choosing animals different from the proposed list.

Operationally, an animal is picked up, the wingID is read. From list sorted by wingID the cage number is picked up and the chick placed in that cage. If this operation is done finishing one breed at a time, then the paper lists can be shorter: one for each breed.

The form needs to contain the following information:

1. choose the breed (if we operate by breed)
2. enter the wingID (i.e. look up on computer, do not use the paper option)
3. if found tick off the ID place the chick in the designated cage.
4. if a proposed chick is replaced (the first died) then the wingID in the form is overwritten.
5. with a submit button upload the complete set and download those that have not been ticked off (dangerous? not safe enough)

The load object will create new entries in CAGE for each cage, storing the wingIDs of the cage inhabitants.

1. for each cage store a new record in CAGE
2. insert the db_cage in ANIMAL for each of the chicks

DS07 two different weights are recorded. They can be treated as one data stream. The scale will produce an ASCII file for batch loading. The animal ID may be an issue and may require some translation from the scale ID.

DS08 the eggs weights are recorded in analogy to DS07

DS09 these are losses and other reasons for animals leaving the herd.

issues: where was the “overwrite” ?

2 Manual rotation scheme

The basis for DS06 is a list with the lines representing the mating plan i.e. the combination of hens and cocks and their allocation to a cage. These are the assumptions:

1. all female chicks with wingIDs are placed into cages
2. three females are put into one cache; they have to be parental half sibs.
3. two cages are populated with parental half sibs, i.e. we have two of item 2. This will result in a total of around n_lines cock families and $n_lines*2$ cages. It seems that they 23 sire lines. Thus they should have 46 cages. Thus, 23 males and $23*2*3=138$ hens.
4. for a rotation scheme we need to know the sequence in which the cocks are moved on. Let us assume we have n_lines sire lines. The distinct cocks in one generation gives the number of sire lines.
5. Thus, for each cock its sire line needs to be known.
6. what do we need to know about the hens? as we do not know which of the hens layed the egg which we selected after hatching we cannot say which of the three hens was the mother
7. hmmm. the way they do it:
 - (a) they have $23*2$ cages/breed. Their IDs are sorted ascendingly. From one year to the next the cage IDs may change (?), the position in the sort is the sID.
 - (b) the new cock from previous year's sID is put into this years sID+1
 - (c) three new hens from previous year's SID are put into this year's cage
 - (d) ok, what is this setup in principle: the $23*2$ cages per breed are placed in some order in the farms. From here on, we refer to each cage through its position in the farm. Then the new male from last years position 1 and 2 will get housed in position 3 and 4. (Only, not every cage produces offspring in the next.) This basic principle reflects the farm model, where next generation sires are moved to the next farm and so on.
 - (e) Thus, eggs are collected from one cage per line only.
 - (f) but