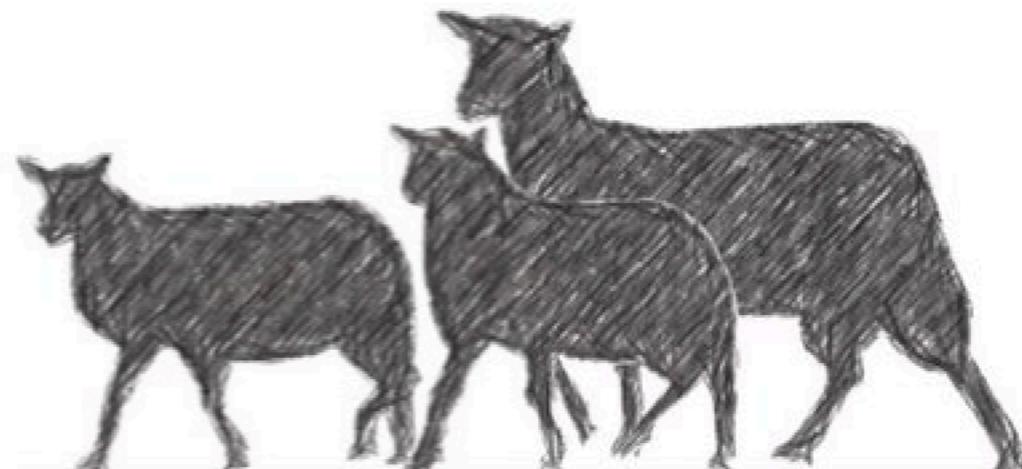


Counting sheep

Oviana - a Pharo based sheep administration system

Kasper Østerbye

ESUG 2022 - Novi Sad Serbia



Mårum lam

Background

- ◆ Mårumlam is our hobby sheep farm
- ◆ around 30 ewes
- ◆ yearly 60 lambs.
- ◆ lamb sold to friends and colleagues and a dealer.
- ◆ started in 2005, more than 700 lambs.
- ◆ Each year we say goodbye to some ewes,
- ◆ and we pick some of the lambs to join the ranks of ewe's.

Spring sheeps



Lemming



Lamb



Lambs



Purpose of system

The system helps us in performing the following tasks:

- ◆ Selecting which lambs to pick for further breeding
- ◆ Selecting which ewes to wave bye to
- ◆ Registering of location (we use some animals for grasing in natural reserves)
- ◆ Registration of weight of lambs 3-4 times a season
- ◆ Registration of medication and minerals - some vaccines are mandatory
- ◆ Registration of births

Statistics

- ◆ What is the average number of lambs born
- ◆ Are they all healthy - do we have to feed them for example
- ◆ How fast do her lambs grow
- ◆ How fast her litter grow in total
- ◆ What is the grow rate over the last three years
- ◆ How many lambs died in birth (a reasonable rare event fortunately)

System domain

- ◆ Statistics. No end to the questions
 - ◇ statistics should be easy to add and present.
- ◆ Registrations.
 - ◇ simple to add new kinds of registrations - and use these in statistics.
- ◆ Consistency
 - ◇ For example ensuring that a weight registration immediately updates the statistics

Disclaimer

I am new to Smalltalk and Pharo

- wrote some visual works in 1992-1996
- started on Pharo in 2020
- Ovina has been a way for me to learn Pharo
- Ovina existed in Java version before

Demo

- ◆ Sheep and Lamb view
 - ✧ Columns
 - ✧ Filters
 - ✧ Printing
- ◆ Events view
 - ✧ Sheep selection part
 - ✧ Event selection & creation
 - ✧ Data entry

Column handling

- ◆ Object model
- ◆ GUI level
- ◆ Registrations
 - ◆ Database
 - ◆ Object model
 - ◆ GUI level

Database schema

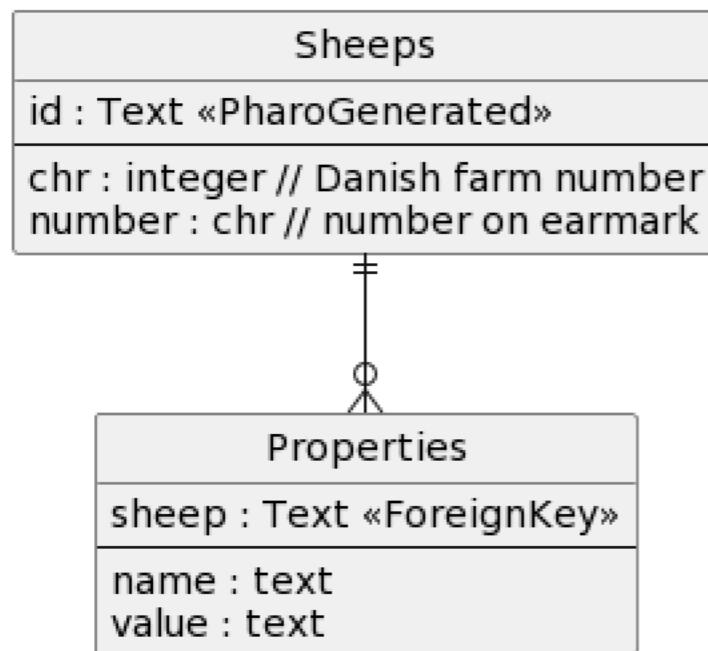
- ◆ Textbook mostly say "one table per domain class"
- ◆ Different ways to handle inheritance
 - ✧ Ovina: Sheep ↳ (Ewe, Ram, Lamb)

Ovina characteristics

- ◆ Fields change - breed was introduced, later removed (computed)
- ◆ It has been moving towards supporting history (the events part)
 - ✧ For example: Location is computed, not a property

Ovina design

- ❖ Sheeps are in a combo of two tables



The sheep id is '[L|E|R]number', Lamb, Ewe, Ram.

Maintaining date in Database

OvinaModel announcer

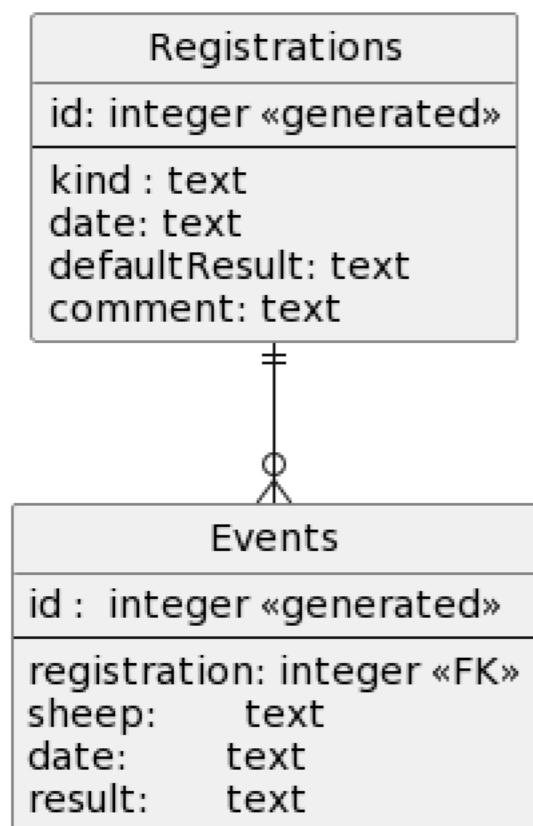
when: OvinaModelSheepChanged do: [:ann | self storeSheep: ann sheep];

when: OvinaModelSheepRemoved do: [:ann | self removeSheep: ann sheep];

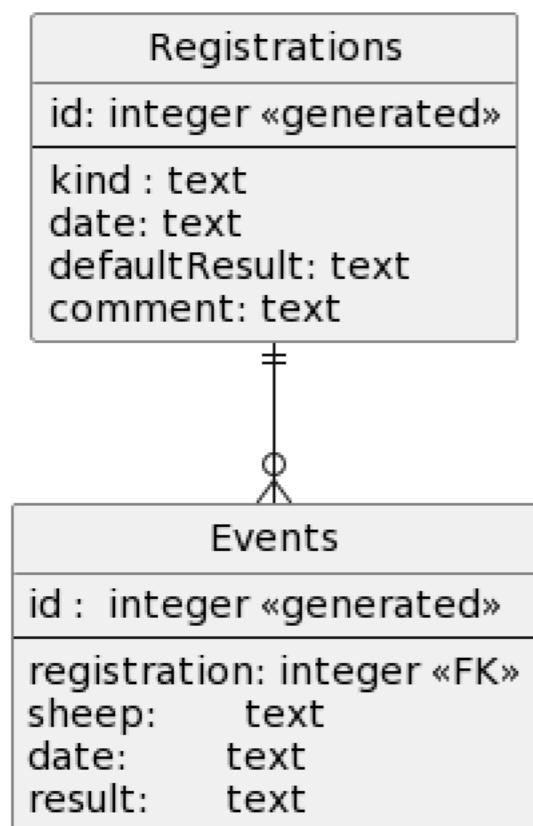
storeSheep: sheep

```
storeSheep: sheep
| stmt properties |
stmt := 'replace into Sheeps (id, chr, number) values ({1}, {2}, {3});'
format: {
    sheep id printString.
    sheep chr.
    sheep number }.
self execute: stmt.
properties := sheep properties associations collect: [ :a |
    '({1}, {2},{3})' format: {
        sheep id printString.
        a key asString printString.
        a value ovinaDBString } ].
properties ifEmpty: [ ^ self ]. "When no properties are yet set."
stmt := 'replace into Properties (sheep, name, value) values {1};'
format: { (properties joinUsing: ',') }.
self execute: stmt
```

Events and Registrations



Events and Registrations



Making it into a MacOS app

- ✓ Tools to clean image
- ✓ Tailor top level menu's

- ✗ Create a MacOS *app*
- ✗ Reset keybinding (no alt-. for example)