Introduction to information systems

Project: Farmer Bronson Alcott's dairy farm

Ana Roman (s2763753)
Jamil Karchoud (s2756560)

January 20, 2016

Abstract

For this assignment, the students were asked to create a database for a farmer who would like to improve his business. The students were given some initial information to begin with, and based on this information, we had to successfully design a database that implemented all the requirements.

Decisions argumentation

- First of all, we decided to have a field "Cattle", which has two sub-types, "Cow" and "Bull". Our group decided in favour of this splitting because there are a few fields which both Cows and Bulls have (for example EarTags or Gender), but at the same time, only Cows can be milked.
  In this way, since we have also added predicates like "Cow **X** is mother of Cattle **Y**" and "Bull **A** is father of Cattle **B**", we can also express the fact that cattle babies inherit the first name from their parents (cows from mothers and bulls from father).
  At the same time, we have added notes to explain what relation there is between the field Cattle and the two subtypes, Cow and Bull, which is obvious: A Cow is a female Cattle and a Bull is a male Cattle.

- By looking at the table, we know that each Cattle must have a unique EarTag. Also, some cows may have some comments, and it must be clearly specified what gender a Cattle has.

- We know that the farmer would like to keep track of the health of his cows. Therefore, our group decided to split, again, make two subtypes for Cow : Healthy Cow and Unhealthy Cow. An unhealthy cow is, strictly speaking, a cow that is sick, or for another reason cannot be milked. Therefore, only the healthy cows are the ones that can be milked.

- Our group decided to make a nested type out of the predicate HealthyCow was milked on MilkingDate, and call this nested type "TheMilking" (calling it only Milking resulted in a program crash so we had to rename it). This is useful, because now we can say that if a cow was milked date X, then this "session", so to say, of milking, has attributes such as Lipidity, Parlor where it took place, etc.

- We know that the farmer has a number of Employees. Out of these employees, only the ones who have a milking certificate can milk a cow. We expressed this using a subset constraint between the unary predicate "has milking certificate" and the predicate "performed" (TheMilking).

- The farmer wanted to be able to calculate the total amount of fat that one session of milking resulted in. This can be calculated if we know the lipidity of the milk (which is expressed in grams per deciliter), and also the volume. Knowing these two values, we can compute the total fat using the following formula : $TotalFat = Lipidity * 10 * Volume$.

1 Modeling assignments

1.1 CSDP 1

Here we are going to draw all fact type relations between entities. Here are all the fact types that are related with the entity type Cattle:

```
Each Cow is an instance of Cattle.
Each Bull is an instance of Cattle.
Cow is the mother of Cattle.
Cattle has Cattle_name.
Cattle has as father Bull.
Cattle has EarTag.
Cattle has Gender.
Cattle was born on Date.
Cattle was sold on Date.
Cattle has Comments.
```

For these fact types, we have the following examples derived from the given table:
Cattle Jo-52 is the mother of Cattle Jo-70
Cattle Jo-71 has as father Bull Fritz-15
Cattle Jo-52 has EarTag 699-140
Cattle Jo-52 has Gender C
Cattle Jo-52 was born on Date January 5 2008
Cattle Fritz-16 was sold on Date April 14 2008
Cattle Jo-52 has Comments "Died at July 16,2010"

Here are all the fact types that are related with the entity type Cow:

```
Fact Types:
Each Cow is an instance of Cattle.
Cow is the mother of Cattle.
Each HealthyCow is an instance of Cow.
Each UnhealthyCow is an instance of Cow.
```

Here are all the fact types that are related with the entity type Bull:

```
Fact Types:
Each Bull is an instance of Cattle.
Cattle has as father Bull.
```

Here are all the fact types that are related with the entity type HealthyCow:

```
Fact Types:
Each HealthyCow is an instance of Cow.
HealthyCow was milked on MilkingDate.
```

Example: *Jo-70 was milked at Date 7:20 , April 2, 2011*

Here are all the fact types that are related with the entity type UnHealthyCow:

```
Fact Types:
Each UnhealthyCow is an instance of Cow.
```

Here are all the fact types that are related with the entity type Employee:

```
Fact Types:
Employee has Employee_name.
Employee has milking parlor certificate.
Employee has Salary.
Employee has Job.
Employee has Skills.
Employee performed TheMilking.
```

Examples:
Employee Bronson Alcott has milking parlor certificate
Employee Bronson Alcott has Salary 2400
Employee Bronson Alcott has Job Manager
Employee Lloyd Garrison has Skills milking parlor certificate

Here are all the fact types that are related with the entity type MilkingDate:

```
MilkingDate has MilkingDate_mdy.
HealthyCow was milked on MilkingDate.
```

Here are all the fact types that are related with the entity type Parlor:

```
Parlor has Parlor_nr.
TheMilking took place on Parlor.
```

Here are all the fact types that are related with the entity type Lipidity:

```
Fact Types:
Lipidity has grams/dlValue.
TheMilking has Lipidity.
```

Here are all the fact types that are related with the entity type Fat:

```
Fact Types:
TheMilking has total Fat.
Fat has gramsValue.
```

Here are all the fact types that are related with the entity type Volume:

```
Fact Types:
TheMilking restulted in Volume.
Volume has literValue.
```

1.2 CSDP 2-5

Here we are going to show you all the (derived) fact types, uniqueness constraints, mandatory role constraints and derived entity types. We will evaluate every fact in our model with their constraints and all the derived entity types.

Here is the explanation of the fact: Cattle has EarTag.

```
 For each EarTag, at most one Cattle has that EarTag.
Each Cattle has exactly one EarTag.
EarTag is a mandatory role constraint of the entity Cattle.
```

Here is the explanation of the fact: Cattle has Gender.

```
Each Cattle has exactly one Gender.
It is possible that more than one Cattle has the same Gender.
Gender is a mandatory role constraint of the entity Cattle.
```

Here is the explanation of the fact: Cattle has Comments.

```
It is possible that some Cattle has more than one Comments
and that for some Comments, more than one Cattle has that Comments.
In each population of Cattle has Comments, each Cattle, Comments combination occurs at most once.
This association with Cattle, Comments provides the preferred identification scheme for CattleHasComments.
```

Here is the explanation of the fact: Cow is the mother of Cattle.

```
For each Cattle, at most one Cow is the mother of that Cattle.
It is possible that some Cow is the mother of more than one Cattle.
Cow is derived from the entity Cattle.
```

Here is the explanation of the fact: Cattle has as father Bull.

```
Each Cattle has as father at most one Bull.
It is possible that more than one Cattle has as father the same Bull.
Bull is derived from the entity Cattle.
```

Here is the explanation of the fact: Cattle was born on Date.

```
Each Cattle was born on exactly one Date.
It is possible that more than one Cattle was born on the same Date.
Date is a mandatory role constraint of the entity Cattle.
```

Here is the explanation of the fact: Cattle was sold on Date.

```
Each Cattle was sold on at most one Date.
It is possible that more than one Cattle was sold on the same Date.
```

Here is the explanation of the fact: HealthyCow was milked on MilkingDate.

```
Each HealthyCow was milked on at most one MilkingDate.
 This association with HealthyCow provides the preferred identification scheme for TheMilking.
It is possible that more than one HealthyCow was milked on the same MilkingDate.
HealthyCow is derived from the entity Cow.
UnhealthyCow is derived from the entity Cow.
TheMilking is a nested entity type from the fact HealthyCow was milked on MilkingDate.
```

Here is the explanation of the fact: Employee performed TheMilking.

```
Each Employee performed at most one TheMilking.
It is possible that more than one Employee performed the same TheMilking.
```

Here is the explanation of the fact: Employee has Salary.

```
Each Employee has exactly one Salary.
It is possible that more than one Employee has the same Salary.
Salary is a mandatory constraint of the entity Employee.
```

Here is the explanation of the fact: Employee has Job.

```
Each Employee has exactly one Job.
It is possible that more than one Employee has the same Job.
Job is a mandatory constraint of the enitity Employee.
```

Here is the explanation of the fact: Employee has Skill.

```
It is possible that for some Skills, more than one Employee has that Skills
 and that some Employee has more than one Skills.
In each population of Employee has Skills, each Skills, Employee combination     occurs at most once.
 This association with Skills, Employee provides the preferred identification scheme for
 EmployeeHasSkills.
```

Here is the explanation of the fact: Employee has milking parlor certificate.

```
In each population of Employee has milking parlor certificate, each Employee occurs at most once.
```

Here is the explanation of the fact: TheMilking took place on Parlor.

```
Each TheMilking took place on exactly one Parlor.
It is possible that more than one TheMilking took place on the same Parlor.
Parlor is a mandatory constraint of the entity TheMilking.
```

Here is the explanation of the fact: TheMilking has Lipidity.

```
Each TheMilking has exactly one Lipidity.
It is possible that more than one TheMilking has the same Lipidity.
Lipidity is a mandatory constraint of the entity TheMilking.
```

Here is the explanation of the fact: TheMilking has total Fat.

```
Each TheMilking has total exactly one Fat.
It is possible that more than one TheMilking has total the same Fat.
Fat is a mandatory constraint of the entity TheMilking.
```

Here is the explanation of the fact: TheMilking restulted in Volume.

```
Each TheMilking restulted in exactly one Volume.
It is possible that more than one TheMilking restulted in the same Volume.
Volume is a mandatory constraint of the entity TheMilking.
```

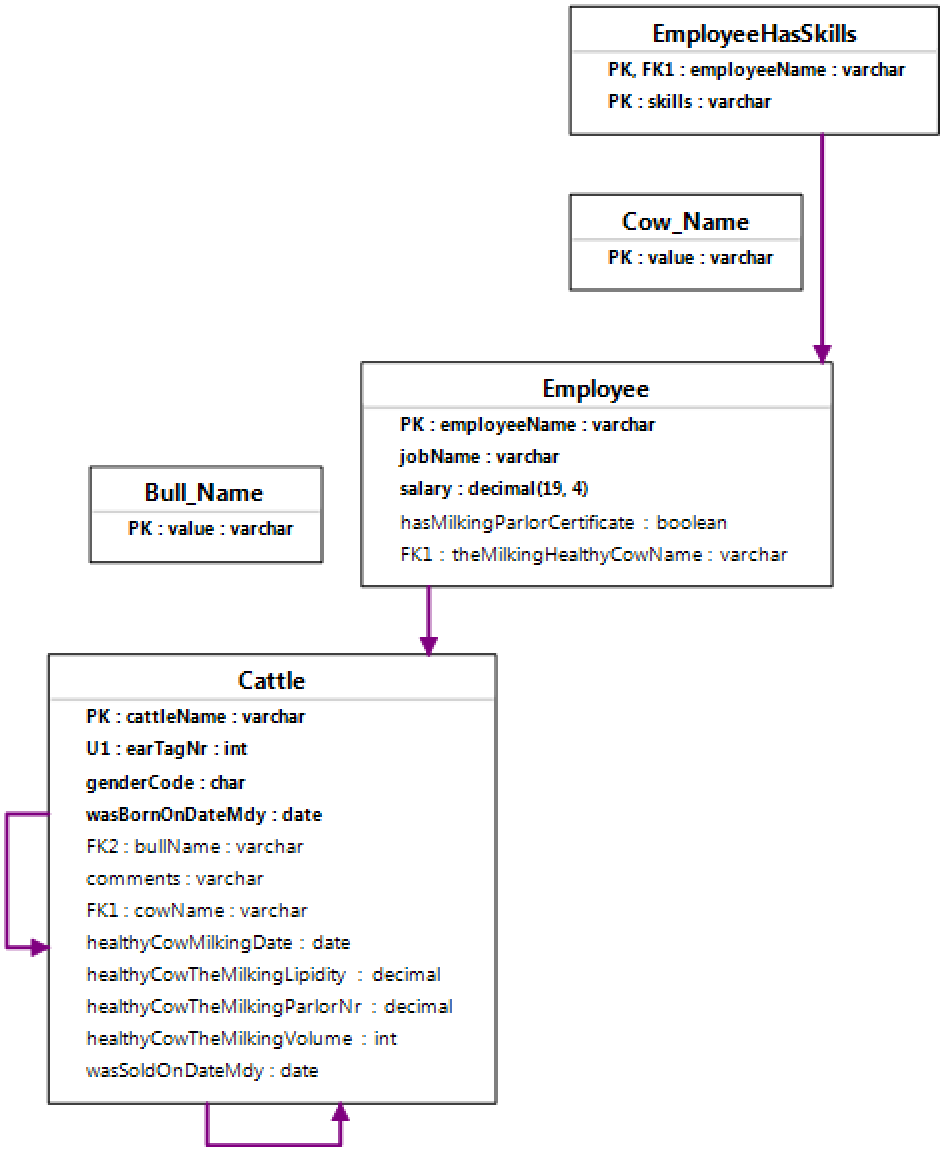
1.3 CSDP 6-7

Here we are going to show you all the sub-typing constraints, value/set comparison and other constraints that were used in our model.

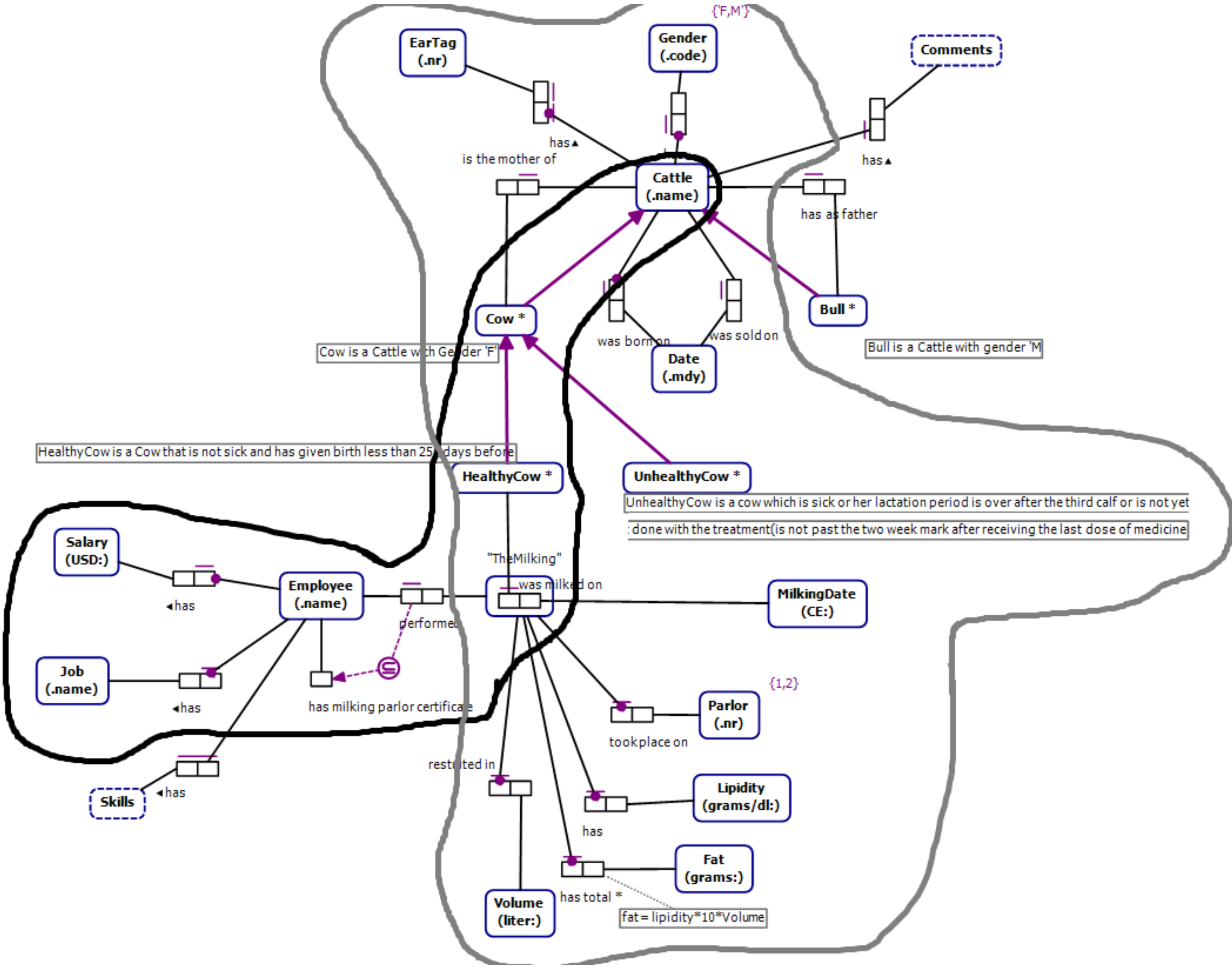
- The entity Gender has only 2 values, which are 'F' and 'M', which corresponds with Female and Male. There is no transgender cattle, so our chosen set of values should suffice.
- The entity Cow is a subtype of the entity Cattle which has as derivation note: Cow is a Cattle with Gender 'F'.
- The entity Bull is a subtype of the entity Cattle which has as derivation note: Bull is a Cattle with gender 'M.
- There is an exclusive or constraint between Cow with the subtype relation to Cattle and Bull with the subtype relation to Cattle, since Cattle can either be male or female.
- The entity HealthCow is a subtype of the entity cow which has as derivation note: HealthyCow is a Cow that is not sick or has given birth less than 250 days before.
- The entity UnhealthCow is a subtype of the entity cow which has as derivation note: UnhealthyCow is a cow which is sick or her lactation period is over after the third calf or is not yet done with the treatment(is not past the two week mark after receiving the last dose of medicine.
- There is an exclusive or constraint between HealthyCow with the subtype relation to Cow and UnhealthyCow with the subtype relation to Cow, because the Cow is either healthy(able to produce milk that is acceptable) or sick(not able to prodce milk that is accceptable.
- There is a subset constraint between the unary fact Employee has milking parlor certificate and the first role of the binary fact Employee performed TheMilking. This means that an Employee can only performe TheMilking if he has a milking parlor certificate.
- The entity Parlor has only 2 values, which are 1 and 2, which corresponds with the number of the Parlor.

1.4 RMap

With the model that we suggested, the Relational Map turned out to be the following:


EmployeeHasSkills
PK, FK1 : employeeName : varchar
PK : skills : varchar
Cow_Name
PK : value : varchar
Employee
PK : employeeName : varchar
jobName : varchar
salary : decimal(19, 4)
hasMilkingParlorCertificate : boolean
FK1 : theMilkingHealthyCowName : varchar
Bull_Name
PK : value : varchar
Cattle
PK : cattleName : varchar
U1 : earTagNr : int
genderCode : char
wasBornOnDateMdy : date
FK2 : bullName : varchar
comments : varchar
FK1 : cowName : varchar
healthyCowMilkingDate : date
healthyCowTheMilkingLipidity : decimal
healthyCowTheMilkingParlorNr : decimal
healthyCowTheMilkingVolume : int
wasSoldOnDateMdy : date

Based on the model this is the lassoing:



2 SQL assignments

2.1 DDL Statements

In order to generate the DDL statements, our group performed the steps suggested in the slides. We got the following code, except that the H2 tool provided gave some errors. We had to make minor changes to the code in order to make it work.

```


CREATE TABLE Employee
(
  employeeName nvarchar(max) NOT NULL,
  jobName nvarchar(max) NOT NULL,
  salary decimal(19,4) NOT NULL,
  hasMilkingParlorCertificate bit,
  theMilkingHealthyCowName nvarchar(max),
  CONSTRAINT Employee_PK PRIMARY KEY(employeeName)
);



CREATE TABLE Cow_Name
(
  "value" nvarchar(max) NOT NULL,
  CONSTRAINT Cow_Name_PK PRIMARY KEY("value")
);



CREATE TABLE Cattle
(
  cattleName nvarchar(max) NOT NULL,
  earTagNr int NOT NULL,
  genderCode nchar(4000) NOT NULL,
  wasBornOnDateMdy date NOT NULL,
  bullName nvarchar(max),
  comments nvarchar(max),
  cowName nvarchar(max),
  healthyCowMilkingDate date,
  healthyCowTheMilkingLipidity decimal(38,38),
  healthyCowTheMilkingParlorNr decimal(38,38),
  healthyCowTheMilkingVolume int,
  wasSoldOnDateMdy date,
  CONSTRAINT Cattle_PK PRIMARY KEY(cattleName),
  CONSTRAINT Cattle_UC UNIQUE(earTagNr),
  CONSTRAINT Cattle_genderCode_RoleValueConstraint1 CHECK (genderCode IN (N'F,M')),
  CONSTRAINT Cattle_healthyCowTheMilkingParlorNr_RoleValueConstraint2 CHECK (healthyCowTheMilkingParlorNr IN (1,2)),
  CONSTRAINT Cattle_TheMilking_MandatoryGroup CHECK (healthyCowTheMilkingParlorNr IS NOT NULL AND healthyCowTheMilkingVolume IS NOT NULL AND healthyCowTheMilkingLipidity IS NOT NULL AND healthyCowMilkingDate IS NOT NULL OR healthyCowTheMilkingParlorNr IS NULL AND healthyCowTheMilkingVolume IS NULL AND healthyCowTheMilkingLipidity IS NULL AND healthyCowMilkingDate IS NULL)
);



CREATE TABLE Bull_Name
(
  "value" nvarchar(max) NOT NULL,
  CONSTRAINT Bull_Name_PK PRIMARY KEY("value")
);



CREATE TABLE EmployeeHasSkills
(
  employeeName nvarchar(max) NOT NULL,
  skills nvarchar(max) NOT NULL,
  CONSTRAINT EmployeeHasSkills_PK PRIMARY KEY(skills, employeeName)
);



ALTER TABLE Employee ADD CONSTRAINT Employee_FK FOREIGN KEY (
    theMilkingHealthyCowName) REFERENCES Cattle (cattleName) ON DELETE NO ACTION
    ON UPDATE NO ACTION;


ALTER TABLE Cattle ADD CONSTRAINT Cattle_FK1 FOREIGN KEY (cowName) REFERENCES
    Cattle (cattleName) ON DELETE NO ACTION ON UPDATE NO ACTION;



ALTER TABLE Cattle ADD CONSTRAINT Cattle_FK2 FOREIGN KEY (bullName) REFERENCES
    Cattle (cattleName) ON DELETE NO ACTION ON UPDATE NO ACTION;



ALTER TABLE EmployeeHasSkills ADD CONSTRAINT EmployeeHasSkills_FK FOREIGN KEY (
    employeeName) REFERENCES Employee (employeeName) ON DELETE NO ACTION ON UPDATE
     NO ACTION;
```

2.2

The commands for adding the sample data in the database is the following:

- For adding the Employees table:

```
INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE )
VALUES ('Bronson Alcott', 'manager','2400', 'yes');

INSERT INTO EMPLOYEEHASSKILLS (EMPLOYEENAME ,SKILLS )
values ('Bronson Alcott', 'MSc. Agricultural engineering, tractor license,
milking parlor certificate');

INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY , HASMILKINGPARLORCERTIFICATE )
VALUES ('William Alcott', 'General Executive','2200','yes');
INSERT INTO EMPLOYEEHASSKILLS (EMPLOYEENAME ,SKILLS )
Values ('William Alcott', 'Bsc. Agri. Engineering, milking parlor certificate');

INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE  )
VALUES ('Abby May Alcott', 'Secretary & Cook','1800','no');

INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE  )
VALUES ('Heinrich Pestalozzi', 'Stableboy','1800','no');
INSERT INTO EMPLOYEEHASSKILLS (EMPLOYEENAME ,SKILLS )
VALUES ('Heinrich Pestalozzi', 'tractor license');
```

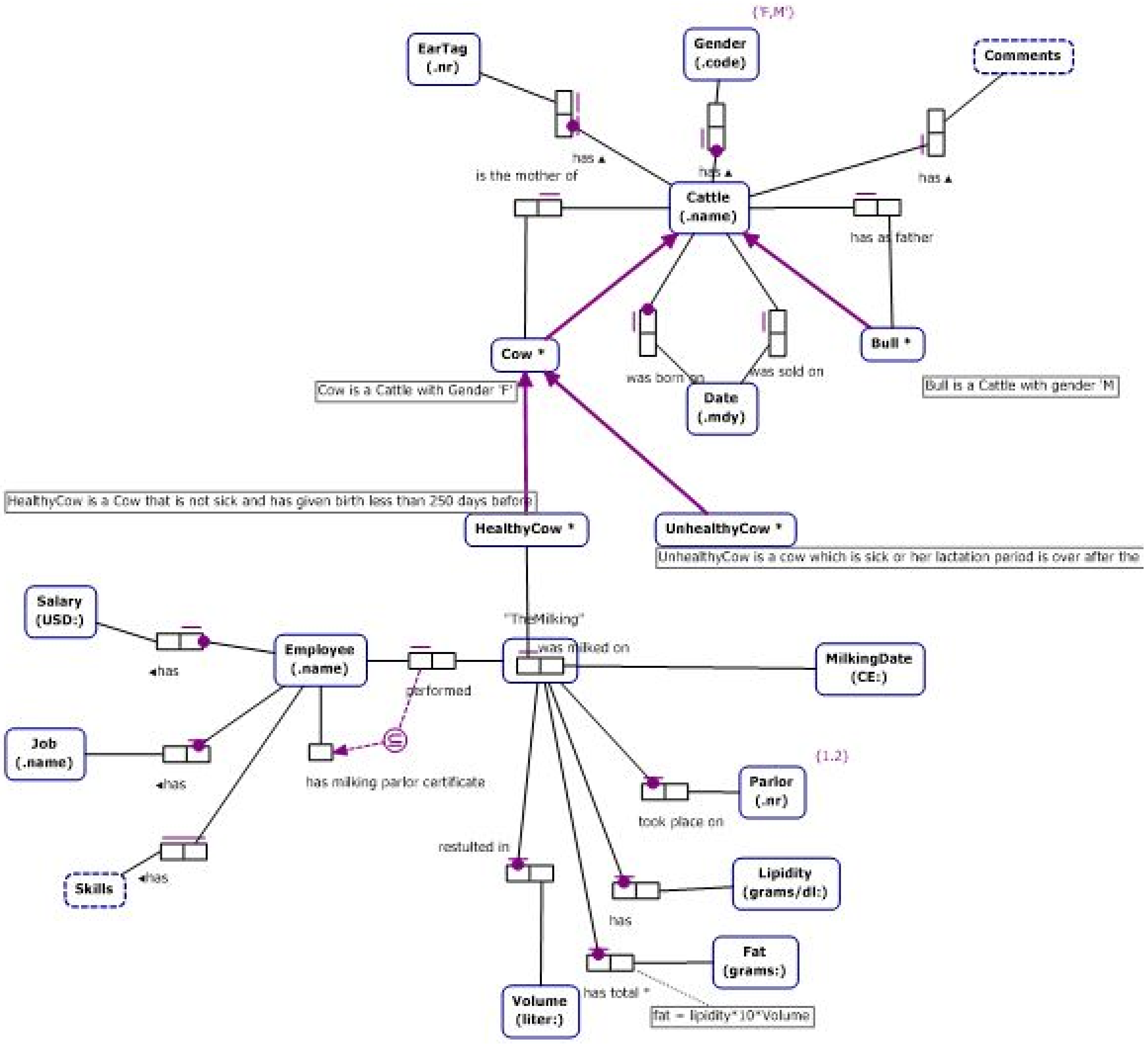
```
INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE  )
VALUES ('Lloyd Garrison', 'Stableboy','1800','yes');
INSERT INTO EMPLOYEEHASSKILLS (EMPLOYEENAME ,SKILLS )
VALUES('Lloyd Garrison','milking parlor certificate');

INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE  )
VALUES ('Louisa May Alcott', 'Shopkeeper','1800','no');

INSERT INTO EMPLOYEE (EMPLOYEENAME ,JOBNAME ,SALARY, HASMILKINGPARLORCERTIFICATE  )
VALUES ('George Sand', 'Stableboy','1700','no');
```

- Note: for the other tables, we ran into an error when inserting time in the table, which we could not solve. And we ran out of time.

The model is the following:


{'F,'M'}
EarTag (.nr)
Gender (.code)
Comments
has ▴
has ▴
has ▴
is the mother of
Cattle (.name)
has as father
Cow *
Cow is a Cattle with Gender 'F'
was born on
was sold on
Date (.mdy)
Bull *
Bull is a Cattle with gender 'M'
HealthyCow is a Cow that is not sick and has given birth less than 250 days before
HealthyCow *
UnhealthyCow *
UnhealthyCow is a cow which is sick or her lactation period is over after the
Salary (USD:)
◂has
Employee (.name)
"TheMilking"
was milked on
performed
MilkingDate (CE:)
Job (.name)
◂has
has milking parlor certificate
{1,2}
Parlor (.nr)
took place on
Skills
◂has
restulted in
Lipidity (grams/dl:)
has
Fat (grams:)
has total *
fat = lipidity*10*Volume
Volume (liter:)