Database Systems — CSci 4380
Midterm Exam #1 Answers
October 1, 2020

For questions 1, 2 and 3, use the following data model:

You are given a database for tracking fall leaf colors (foliage) in the U.S. This database records a number of locations with a given `name`, `state` and `altitude` (in feet). We store which trees are in each location in `TreeCoverage`. Each location has a number of different tree types (example of `treetype` values: `birch tree`, `maple tree`, etc.) For each `locationid`, `year`, and `treetype`, we store `coverage`, which is the percentage of this location that is covered with this tree (0-100). Finally the `FallColors` relation stores the peak level of a specific color at a given location for any given week of the year (a value between 1-52) and year. The term peak refers to quality of color, when `peakpercent` is 100, the colors are at their brightest. Some locations may never reach a `peakpercent` of 100% in a given year due to weather conditions. The lowest `peakpercent` is 0. Colors that are tracked in the database are `red`, `orange`, and `yellow`. The key for each relation is underlined.

```
Locations(locationid, name, state, altitude)
TreeCoverage(locationid, treetype, year, coverage)
FallColors(week, year, locationid, color, peakpercent)
```

Version 1 (v1).

Question 1 (10 points). Return the `week` and `year` values from `FallColors` for the location with `name` `Saratoga Springs` in which the `peakpercent` value for both `red` and `orange colors` was 80 or higher in the same `year`.

Answer.

```
Both answers below were accepted due to unclear wording.
```

Same week for both red and orange:

```
R1 = select_{name='Saratoga Springs'} Locations
R2 = project_{week, year, locationid} (select_{color='red' and peakpercent>=80} FallColors)
R3 = project_{week, year, locationid} (select_{color='orange' and peakpercent>=80} FallColors)
Result = project_{week, year} (R1 * R2 * R3)
```

Not necessarily same week for red and orange; return all weeks:

```
R1 = select_{name='Saratoga Springs'} Locations
R2 = project_{week, year, locationid} (select_{color='red' and peakpercent>=80} FallColors)
R3(week1, year, locationid) =
  project_{week, year, locationid} (select_{color='orange'  and peakpercent>=80} FallColors)
R5 = R1 * R2 * R4
R6(week, year) = project_{week1, year} (R5)
Result = R6 union (project_{week,year} R5)
```

Version 2 (v2).

Question 1 (10 points). Return the `locationid, name` of all locations in New York State (`state= NY`) that had `peakpercent` value of 60 or more for two different types of `colors` in `week` 50 of the same `year`.

Answer.

```
R1 = select_{peakpercent>=60 and week=50} FallColors
R2 = project_{color, locationid, year} R1
R3(color1, locationid1, year1) = R2
R4 = R2 join_{color<>color1 and locationid=locationid1 and year=year1} R3
Result = project_{locationid, name} (R4 * Locations)
```

Alternative that uses natural join:

```
R1 = select_{peakpercent>=60 and week=50} FallColors
R2 = project_{color, locationid, year} R1
R3(color1, locationid, year) = R2
R4 = select_{color<>color1} (R2 * R3)
Result = project_{locationid, name} (R4 * Locations)
```

Version 3 (v3).

Question 1 (10 points). Return the `locationid, name` of all locations at `altitude` 2000 feet or higher that have at least two different `treetypes` with 25 percent or more `coverage` value in the same `year`.

Answer.

```
R1 = select_{altitude>=2000} (Locations)
R2 = project_{treetype, locationid,year} (select_{coverage>=25} (TreeCoverage))
R3(treetype1, locationid1,year1) = R2
R4 = R2 join_{treetype<>treetype1 and locationid1=locationid and year1=year} R3
Result = project_{locationid, name} (R4 * Locations)
```

Alternative that uses natural join:

```
R1 = select_{altitude>=2000} (Locations)
R2 = project_{treetype, locationid,year} (select_{coverage>=25} (TreeCoverage))
R3(treetype1, locationid,year) = R2
R4 = select_{treetype<>treetype1} (R2*R3)
Result = project_{locationid, name} (R4 * Locations)
```

Version 1 (v1).

Question 2 (10 points). Return the `locationid, name` of all locations that have no `pine trees` (for `treetype`) and had 100 `peakpercent` of `orange` color in `year` 2019.

Answer.

```
Two possible interpretations were accepted for the first selection (R1 or R1').

R1 = project_{locationid} (select_{treetype='Pine Tree'} TreeType)
R1' = project_{locationid} (select_{treetype='Pine Tree' and year=2019} TreeType)

R2 = select_{peakpercent=100 and year=2019 and color='orange'} FallColors
R3 = project_{locationid} (R2)
R4 = R3 - R1
R5 = project_{locationid, name} (R4 * Locations)
```

Version 2 (v2).

Question 2 (10 points). Return the `locationid, name` of all locations that have `birch trees` (for `treetype`) and never had `yellow color` at 100 `peakpercent` value in `year 2019`.

Answer.

```
R1 = project_{locationid} (select_{treetype='Birch Tree'} TreeType)
R1' = project_{locationid} (select_{treetype='Birch Tree' and year=2019} TreeType)

R2 = select_{peakpercent=100 and year=2019 and color='yellow'} FallColors
R3 = project_{locationid} (R2)
R4 = R1 - R3
R5 = project_{locationid, name} (R4 * Locations)
```

Version 1 (v1).

Question 3 (10 points). Return the `locationid, name` of all locations in New York State (`state= NY`) at `altitude` lower than 1000 that has at least 20% `maple tree` coverage and 80 or higher `peakpercent` value for `red color` in `year 2019`.

Answer.

```
R1 = select_{state='NY' and altitude<1000} Locations
R2 = select_{treetype='Maple Tree' and coverage>= 20} TreeCoverage
R3 = select_{peakpercent>=80 and color='red' and year=2019} FallColors
R4 = project_{locationid,name} (R1*R2*R3)
```

Version 2 (v2).

Question 3 (10 points). Return the `locationid, name` of all locations in the `state` of `Vermont` that reached 100 `peakpercent` in red `color` at least once in `year` 2019 and have at least one `treetype` that also grows in at least one location in New York State (`state= NY`).

Answer.

```
R1 = select_{state='Vermont'} Locations
R2 = select_{peakpercent=100 and color='red' and year=2019} FallColors
R3 = project_{treetype} (select_{state='NY'} Locations * TreeCoverage)
R4 = project_{locationid, name} (R1*R2*R3*TreeCoverage)

R3: treetypes that grow in NY State
```

Version 1 (v1).

Question 4 (12 points). In this problem, we are given the following relation for our sheep database:

```
woolproduced(shearingid, shearingdate, sheepid, sheepname, ownerid, purchasedate,
             woolweight, woolquality)
```

Two different sheep can have the same `sheepname`, but not the same `sheepid`. Given a sheep (determined by `sheepid`), `woolquality` is fixed. Given a `shearingid`, there is a unique `shearingdate, sheepid, woolweight`. A sheep can have many owners over time. Given a `sheepid` and an `ownerid`, there is a unique `purchasedate`.

(a) Based on the above information, list all applicable functional dependencies.

(b) What are the key(s)?

(c) Is this relation in BCNF? 3NF? Explain why or why not.

(d) If it is not in 3NF, use 3NF decomposition to find relations that are in 3NF.

Answer.

```
Interpretation 1: sheep can have 1 name

(a) sheepid -> sheepname woolquality
    shearingid -> shearingdate sheepid woolweight
    sheepid ownerid -> purchasedate

(b) Key: shearingid ownerid

(c) Not in BCNF or 3NF. None of the FDs have a superkey on the left
and no prime attributes on the right.

(d) 3NF Decomposition:

S1(sheepid, sheepname, woolquality)
S2(shearingid, shearingdate, sheepid, woolweight)
S3(sheepid, ownerid, purchasedate)
S4(shearingid, ownerid)   ## for the key!
```

Alternative answer.

```
Interpretation 2: sheep can have 2 names

(a) sheepid -> woolquality
    shearingid -> shearingdate sheepid woolweight
    sheepid ownerid -> purchasedate

(b) Key:  sheepname shearingid ownerid

(c) Not in BCNF or 3NF. None of the FDs have a superkey on the left
and no prime attributes on the right.

(d)
S1(sheepid, woolquality)
S2(shearingid, shearingdate, sheepid, woolweight)
S3(sheepid, ownerid, purchasedate)
S4(sheepname, shearingid, ownerid)   ## for the key!
```

Version 1 (v1).

Question 5 (10 points). You are given a set $\mathcal{F} = \{AB \to DE, BD \to CF, CD \to A\}$ of functional dependencies for relation the $R(A, B, C, D, E, F)$.

Check all options below that are true for this relation:

```
R is in 3NF
R is in BCNF
AB->DE violates 3NF
AB->DE violates BCNF
BD->CF violates 3NF
BD->CF violates BCNF
CD->A violates 3NF
CD->A violates BCNF
AB is a key
BD is a key
CD is a key
Set F implies AB->F
Set F implies BD->A
Set F implies CD->B
None of these options is true
```

Answer.

```
F= {AB-> DE, BD -> CF, CD-> A} R(A,B,C,D,E,F).
Key: AB, BD   Not in BCNF (CD->A), in 3NF (A is a prime attr)

Correct:

R is in 3NF
CD->A violates BCNF
AB is a key
BD is a key
Set F implies AB->F
Set F implies BD->A

Incorrect:

R is in BCNF
AB->DE violates 3NF
AB->DE violates BCNF
BD->CF violates 3NF
BD->CF violates BCNF
CD->A violates 3NF
CD is a key
Set F implies CD->B
None of these options is true
```

Version 2 (v2).

Question 5 (10 points). You are given a set $\mathcal{F} = \{AB \to DE, BD \to CF, CD \to B\}$ of functional dependencies for the relation $R(A, B, C, D, E, F)$.

Check all options below that are true for this relation:

```
R is in 3NF
R is in BCNF
AB->DE violates 3NF
AB->DE violates BCNF
BD->CF violates 3NF
BD->CF violates BCNF
CD->B violates 3NF
CD->B violates BCNF
AB is a key
BD is a key
CD is a key
Set F implies AB->F
Set F implies BD->A
Set F implies CD->B
None of these options is true
```

Answer.

```
F= {AB-> DE, BD -> CF, CD-> B} R(A,B,C,D,E,F).
Key: AB, ACD  Not in BCNF (BD->CF, CD->B), not in 3NF (BD->CF)

Correct

AB is a key
Set F implies AB->F
BD->CF violates 3NF
BD->CF violates BCNF
CD->B violates BCNF
Set F implies CD->B


Incorrect
R is in 3NF
R is in BCNF
BD is a key
CD is a key
AB->DE violates 3NF
AB->DE violates BCNF
CD->B violates 3NF
Set F implies BD->A

None of these options is true
```

Version 3 (v3).

Question 5 (10 points). You are given a set $\mathcal{F} = \{AB \to DE, BD \to CF, CD \to B\}$ of functional dependencies for the relation $R(A, B, C, D, E, F, G)$.

Check all options below that are true for this relation:

```
R is in 3NF
R is in BCNF
AB->DE violates 3NF
AB->DE violates BCNF
BD->CF violates 3NF
BD->CF violates BCNF
CD->B violates 3NF
CD->B violates BCNF
AB is a key
BDG is a key
CDG is a key
Set F implies AB->F
Set F implies BD->A
Set F implies CD->B
None of these options is true
```

Answer.

```
F= {AB-> DE, BD -> CF, CD-> B} R(A,B,C,D,E,F,G).
Key: ABG, ACDG   Not in BCNF and 3NF, all violate BCNF,
     first two violate 3NF (E and F are not prime attributes)!

Correct:
AB->DE violates 3NF
AB->DE violates BCNF
BD->CF violates 3NF
BD->CF violates BCNF
CD->B violates BCNF
Set F implies AB->F
Set F implies CD->B

Incorrect:

R is in 3NF
R is in BCNF
CD->B violates 3NF
AB is a key
BDG is a key
CDG is a key
Set F implies BD->A
None of these options is true
```

Version 1 (v1).

Question 6 (10 points). You are given a set $\mathcal{F} = \{ABC \to D, BC \to E, BF \to CG, FG \to A\}$ of functional dependencies for the relation $R(A, B, C, D, E, F, G)$.

Using the Chase decomposition algorithm, determine if the following decomposition is lossless or not.

$R1(A, B, C, D)$
$R2(B, C, F)$
$R3(B, F, G)$
$R4(A, B, C, E)$

Show your work:

Answer.

```
a    b    c    d    e1   f1   g1
a2   b    c    d2   e2   f    g2
a3   b    c3   d3   e3   f    g
a    b    c    d4   e    f4   g4

apply ABC -> D, rows 1,4

a    b    c    d    e1   f1   g1
a2   b    c    d2   e2   f    g2
a3   b    c3   d3   e3   f    g
a    b    c    d    e    f4   g4

apply BC->E, rows 1,2,4

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g2
a3   b    c3   d3   e3   f    g
a    b    c    d    e    f4   g4

apply BF->CG, rows 2,3

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g
a3   b    c    d3   e3   f    g
a    b    c    d    e    f4   g4

Apply: FG ->A, rows 2,3

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g
a2   b    c    d3   e3   f    g
a    b    c    d    e    f4   g4

Apply: BC -> E rows 2,3

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g
a2   b    c    d3   e    f    g
a    b    c    d    e    f4   g4

FDs can be applied some more, but we cannot find a row without subscript.
This decomposition is lossy!
```

Version 2 (v2).

Question 6 (10 points). You are given a set $\mathcal{F} = \{ABC \to D, BC \to E, BF \to CG, FG \to A\}$ of functional dependencies for the relation $R(A, B, C, D, E, F, G)$.

Using the Chase decomposition algorithm, determine if the following decomposition is lossless or not.

$R1(A, B, C, D)$
$R2(B, C, F)$
$R3(A, B, F, G)$
$R4(A, B, C, E)$

Show your work.

Answer.

```
F = {ABC -> D, BC -> E, BF -> CG, FG -> A}
R(A,B,C,D,E,F,G)

R1(A,B,C,D)
R2(B,C,F)
R3(A,B,F,G)
R4(A,B,C,E)

a    b    c    d    e1   f1   g1
a2   b    c    d2   e2   f    g2
a    b    c3   d3   e3   f    g
a    b    c    d4   e    f4   g4

Apply ABC->D, rows 1,3,4

a    b    c    d    e1   f1   g1
a2   b    c    d2   e2   f    g2
a    b    c3   d    e3   f    g
a    b    c    d    e    f4   g4

Apply BC->E, rows 1,2,4

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g2
a    b    c3   d    e3   f    g
a    b    c    d    e    f4   g4

Apply, BF->CG, rows 2,3

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g
a    b    c    d    e3   f    g
a    b    c    d    e    f4   g4

Apply BC->E, all rows

a    b    c    d    e    f1   g1
a2   b    c    d2   e    f    g
a    b    c    d    e    f    g    <- no subscript
a    b    c    d    e    f4   g4

This is a lossless decomposition
```

Version 1 (v1).

Question 7 (12 points). In this problem, we are given the following relation for keeping track of the important turnip prices in Animal Crossing:

`turnipprices(turniptype, islandid, islandname, morningornot, price, nativeflower, villagername, nativebug)`

with the following functional dependencies:

`turniptype islandid morningornot` → `price`

`islandid` → `islandname villagername`

Note that islands have multiple native flowers and bugs. Answer the following questions:

(a) **What are the key(s)?**

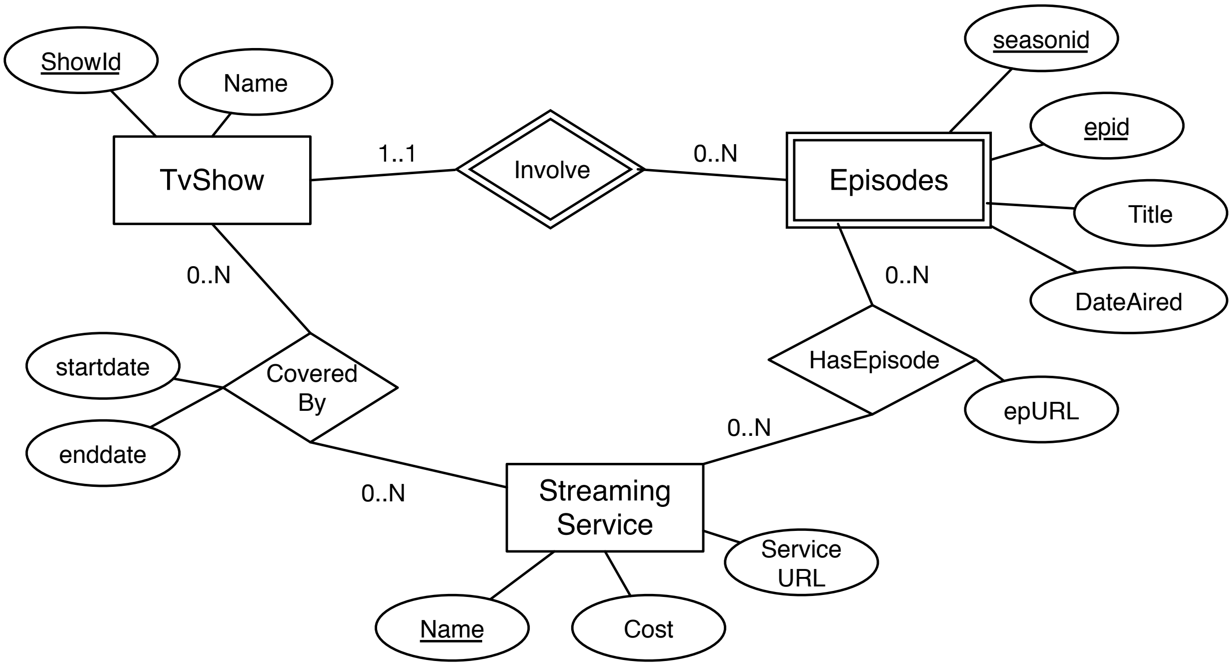
(b) **If this relation is not in BCNF, use BCNF decomposition to find relations that are in BCNF.**

(c) **Are there any relations in the result of BCNF decomposition that might not be in 4NF? Explain which one and very briefly why.**

Answer.

```
turnipprices(turniptype, islandid, islandname, morningornot, price, nativeflower, villagername, nativebug)

turniptype islandid morningornot -> price
islandid -> islandname villagername

(a) Key: turniptype islandid morningornot nativeflower nativebug

(b) Not in BCNF, all fds violate it (none have a superkey on the left)

Take out islandid -> islandname villagername

T1(islandid,islandname,villagername)  islandid -> islandname villagername , in BCNF

T2(turniptype, islandid, morningornot, price, nativeflower, nativebug)
    turniptype islandid morningornot -> price
    Key: turniptype islandid morningornot nativeflower nativebug
    Not in BCNF!, take out the FD

T21(turniptype, islandid, morningornot, price)
    turniptype islandid morningornot -> price, IN BCNF

T22(turniptype, islandid, morningornot, nativeflower, nativebug)
    No, FD, in BCNF

(c) This is in 4NF because flowers and bugs are multivalued attribute but
they do not relate to each other.
```

Version 1 (v1).

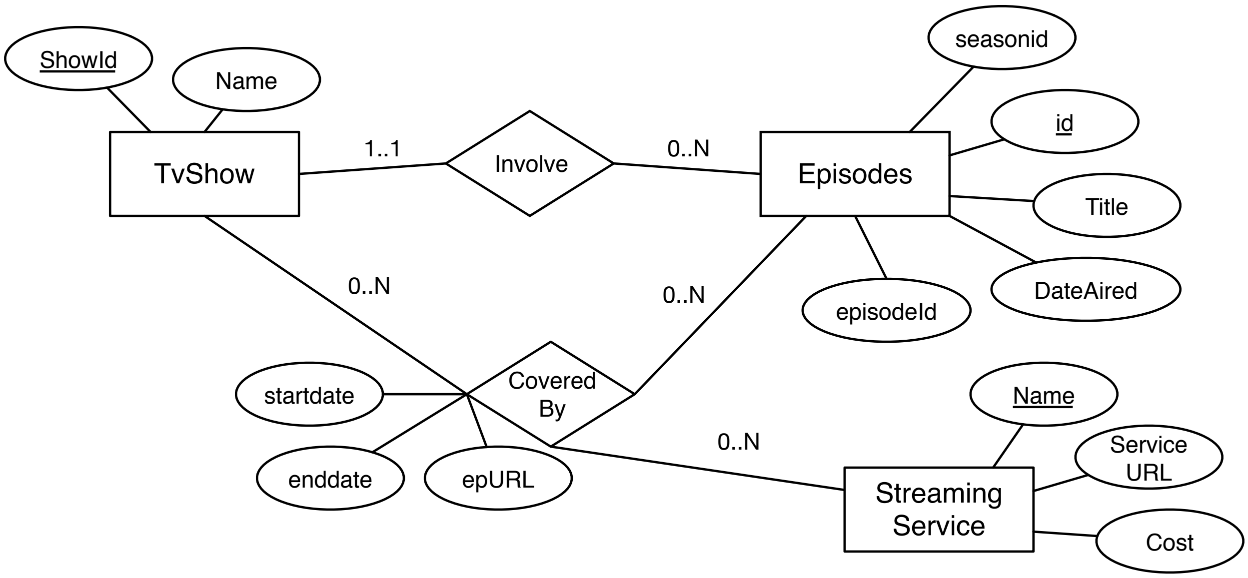


Question 8 (10 points). You are given the ER diagram above. List all relations resulting from the conversion of this diagram to the relational data model. Also list the key of each relation clearly.

Answer.

```
TVShow(ShowId, Name)
        Key: ShowID

Episodes(ShowId, seasonId, epid, Title, DateAired)
        Key: ShowId, seasonId, epid

StreamingService(Name, Cost, ServiceURL)
        Key: Name

CoveredBy(ShowId, StreamingService_Name, startdate, enddate)
        Key: ShowId, StreamingService_Name

HasEpisode(ShowId, seasonId, epid, StreamingService_Name, epURL)
        Key: ShowId, seasonId, epid, StreamingService_Name
```

Version 2 (v2).



Question 8 (10 points). You are given the ER diagram above. List all relations resulting from the conversion of this diagram to the relational data model. Also list the key of each relation clearly.

Answer.

```
TVShow(ShowId, Name)
        Key: ShowID

Episodes(id, seasonId, episodeid, Title, DateAired, ShowId)
        Key: id

StreamingService(Name, Cost, ServiceURL)
        Key: Name

CoveredBy(ShowId, StreamingService_Name, id, startdate, enddate, epURL)
        Key: ShowId, StreamingService_Name, id
```

Version 1 (v1).

Question 9 (16 points). Create an Entity-Relationship diagram for the following database, capturing all the requirements below precisely. Make sure you list all the relevant attributes, and underline the keys. For each relationship, mark the participation constraints clearly (one-to-one, one-to-many, or many-to-many). If you do not find a natural key for an entity, feel free to add an id attribute.

In this database, we will store a number of farms. Each farm has a name, state, address and url. Farm names are unique in each state. The database stores fairs identified by a unique name. Additionally, fairs have state, address and URL attributes. The database also stores events. Each event is the occurrence of one and only one fair in a given year. A fair can have at most one event in a given year. Events have start and end dates, and a ticket price. Farms participate in zero or more events and events feature one or more farms. Farms sell products in zero or more events. Products are identified by their name. A product may be the subgroup of at most one other product and have many subgroup products. There are also competitions in the database identified by an id. Each competition has a name and type and an associated event. Farms participate in zero or more competitions. A competition may have zero to 10 farms participating in them.

Answer.

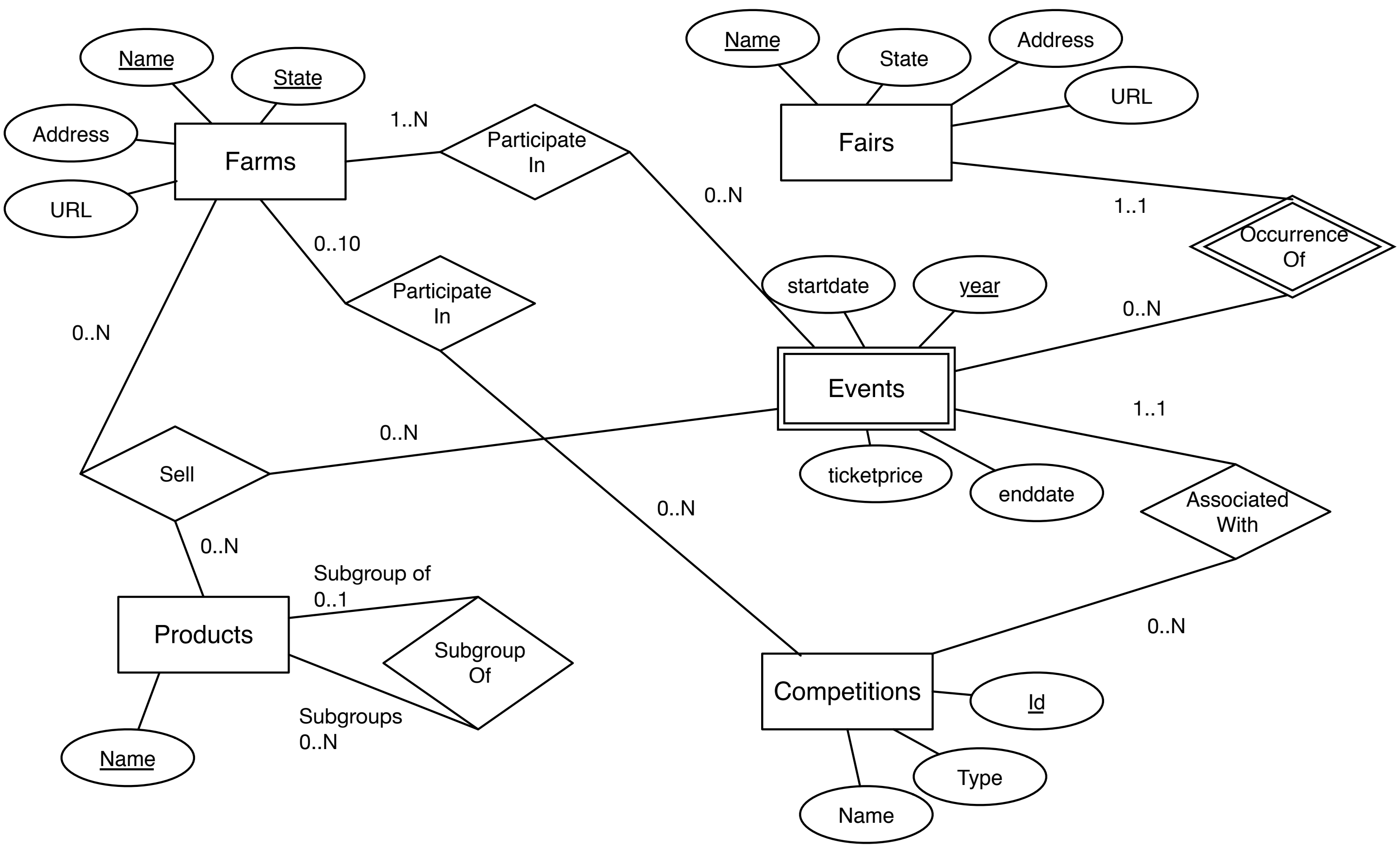


Figure 1: ER Diagram for Question 9 (v1)

Version 2 (v2).

Question 9 (16 points). Create an Entity-Relationship diagram for the following database, capturing all the requirements below precisely. Make sure you list all the relevant attributes, and underline the keys. For each relationship, mark the participation constraints clearly (one-to-one, one-to-many, or many-to-many). If you do not find a natural key for an entity, feel free to add an id attribute.

In this database, we store a number of farms. Each farm has a name, state, address, url and owner name. Farm names are unique for each state. Farms grow produce and raise animals. A produce is identified by its name. Produce also have type and hardiness score attributes. Animals are identified by their breed name (for example Merino Sheep). Additionally, each animal has a place of origin, type and a description. Farms grow multiple kinds of produce in each year and the same produce can be grown by multiple farms. What produce is grown may change from year to year for each farm. Farms raise many animals and animals are raised in zero to many farms.

The database also stores a number of farmer's markets. Each market is identified by an id and also stores its name, address, day of week, start and end times. Farms participate in zero or more farmer's markets and each market has at least 5 farms and can have many more. Markets allow zero or more kinds of produce and each produce can be sold in zero or more markets.

Answer.

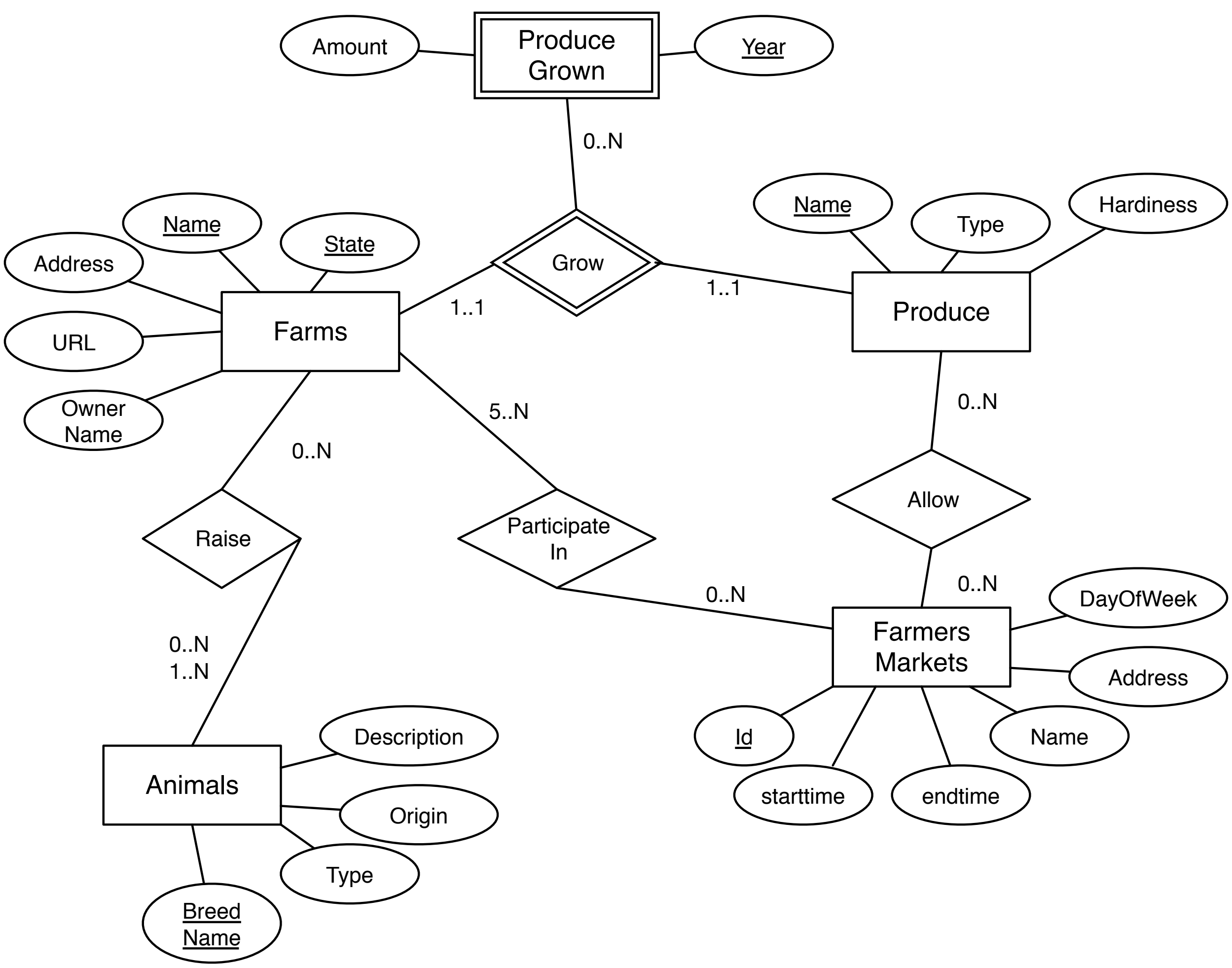


Figure 2: ER Diagram for Question 9 (v2)