University of British Columbia, Vancouver
Department of Computer Science

Milestone 2

03/01/2023

Group #10

Name	**Student Number**	**CS Alias (Userid)**	**Preferred E-mail Address**
Benjamin Farris	20530606	j8c9p	benner.farris@gmail.com
Jonathan Chow	33049743	e4i4a	jon.chowyyc@gmail.com
Allen Zhu	63577860	u6y8q	allenzhu2000@hotmail.com

By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, and then let us assign you to a TA for your project supervisor.)

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia.

Description

This project is a farm management system that keeps track of many farming utilities and processes such as facilities, fields, animals, crops, farmers, products, and buyers, with the intention of helping farm owners manage their farms more efficiently and effectively. The main aspects of the system are separated between staff, livestock, agriculture and the many components that are required to mass produce these items. All of these many entities will be linked to provide a user interface for the farmer in order to keep track of tasks needing to be accomplished, which include those listed above, in addition to keeping track of expenses and profits. Furthermore, important distributors will also have access to the system in order to purchase products to distribute to other manufacturers, which implies a needed access to the database in order to view available products.

Changes to ER Diagram

A revised version of the ER diagram from Milestone 1:

- Bolded “HAS” relationship (part of weak entity)
- Added partial key “recordID” to weak entity “Veterinary Records”
- Renamed attribute of “plantedOrHarvested” from “Crops entity” to “plantStatus” (enum that is either “planted” or “harvested”)
- Renamed attribute “plot#” of entity “Fields” to “plotNum”
- Added attribute “harvestable” to entity livestock

Schema

<u>Enum Definitions</u>

Enum Name	**Enum Elements**
CropType	canola, wheat, corn, potatoes, mustard, coconut,
AnimalType	chicken, sheep, cow, pig
ProductType	milk, eggs, beef, bacon, poultry, wool, manure
HealthStatusType	injured, sick, healthy, dead
CropStatusType	planted, harvested
CropVariantType	pollinated, hybrids
FieldState	empty, growing, resting

Following class conventions: <u>Underlined Attributes</u> are considered <u>PKs</u>, **bolded Attributes** are considered **FKs**. **<u>Bolded and Underlined Attributes</u>** are considered **<u>both PKs and FKs.</u>**
By default consider all our PKs and FKs as having the constraint of NOT NULL.

Note: All PKs are our Candidate Keys.

ENTITIES:

- Pen (
    facilityID: INTEGER,
    name: VARCHAR(30),
    age: INTEGER,
    maxCapacity: INTEGER NOT NULL,
    upkeep: INTEGER,
    typeOfAnimals: ENUM AnimalType
)
- Housing (
    facilityID: INTEGER,
    name: VARCHAR(30),
    age: INTEGER,
    maxCapacity: INTEGER,
    upkeep: INTEGER NOT NULL,
    rent: INTEGER,
    distanceToFields: INTEGER,
    distanceFromPens: INTEGER
)
- Farmers (
    farmerID: INTEGER,
    fullName: VARCHAR (40) NOT NULL,
    yearsOfEmployment: INTEGER,
    salary: INTEGER,
)
- Fields (
    plotNum: INTEGER,
    nutrientLevels: REAL,
    suitableCrops: ENUM CropType,
    capacity: INTEGER,
    state: ENUM FieldState,
    pesticidesAndChemicals: BOOLEAN
)
- Crops (
    cropType: ENUM CropType,
    variant: ENUM CropVariantType,
    plantStatus: ENUM CropStatusType,
    quantity: INTEGER
)
- Livestock (
    tagID: INTEGER PRIMARY KEY,

animalType: ENUM AnimalType,
age: INTEGER,
diet: ENUM CropType,
weight: REAL,
lastFed: DATE,
lastViolatedForHarvestedGoods: DATE,
harvestable: BOOLEAN
)

- LivestockProduce (
  productType: ENUM ProductType,
  quantity: INTEGER
)
- Buyers_DealsWith (
  buyerID: INTEGER,
  farmerID: INTEGER NOT NULL REFERENCES Farmers,
  address: VARCHAR(100),
  name: VARCHAR(40),
  phoneNumber: VARCHAR(30),
  date: DATE
)
- VeterinaryRecords_Has (
  tagID: INTEGER NOT NULL REFERENCES Livestock,
  recordID: INTEGER,
  date: DATE,
  healthStatus: ENUM HealthStatusType
)

RELATION SETS:

- Manages_Housing (
  facilityID: INTEGER REFERENCES Housing,
  farmerID: INTEGER REFERENCES Farmers,
  lastMaintained: Date
)
- Manages_Pen (
  facilityID: INTEGER REFERENCES Pen,
  farmerID: INTEGER REFERENCES Farmers,
  lastMaintained: Date
)
- Lives (
  facilityID: INTEGER REFERENCES Housing,
  farmerID: INTEGER REFERENCES Farmers,
  farmersHoused: INTEGER
)

- Contains (
    facilityID: INTEGER REFERENCES Pen,
    tagID: INTEGER REFERENCES Livestock,
    capacityOccupied: INTEGER
)
- Nurtures (
    farmerID: INTEGER REFERENCES Farmers,
    tagID: INTEGER REFERENCES Livestock,
    waterSpent: INTEGER,
    foodSpent: INTEGER
)
- Creates (
    productType: ENUM ProductType REFERENCES LivestockProduce,
    tagID: INTEGER REFERENCES Livestock,
    amount: INTEGER,
    dateProduced: DATE
)
- Tends (
    farmerID: INTEGER REFERENCES Farmers,
    plotNum: INTEGER REFERENCES Fields,
    waterSpent: INTEGER
)
- Buys (
    buyerID: INTEGER REFERENCES Buyers_DealWith,
    cropType: ENUM CropType,
    variant: VARCHAR(30),
    productType: ENUM ProductType REFERENCES LivestockProduce,
    quantity: INTEGER,
    price: INTEGER,
    transactionID: INTEGER
    FOREIGN KEY (**cropType**, **variant**) REFERENCES Crops
)
- IsGrowing (
    plotNum: INTEGER REFERENCES Fields,
    cropType: ENUM CropType,
    variant: ENUM CropVariantType,
    lastWatered: DATE,
    plantedDate: DATE,
    harvestDate: DATE,
    FOREIGN KEY (**cropType**, **variant**) REFERENCES Crops
)

Functional Dependencies

Livestock

- Weight, animalType → Diet
- animalType, age → harvestable
- tagID → weight, animalType, lastFed, age, lastViolatedForHarvestedGoods, diet, harvestable

Fields

- nutrientLevels → suitableCrops
- suitableCrops → pesticidesAndChemicals
- suitableCrops, plotNum → capacity
- plotNum → state, capacity, suitableCrops, pesticidesAndChemicals, nutrientLevels

Farmers

- yearsOfEmployment → salary
- farmerID → yearsOfEmployment, salary, fullName

Normalization

Livestock

FDs:

- Weight, animalType → Diet (1)
- animalType, age → harvestable (2)
- tagID → weight, animalType, lastFed, age, lastViolatedForHarvestedGoods, diet, harvestable (3)

To begin, we need to state the candidate key. The candidate key includes all attributes which are not found in the FDs above and the attributes that cannot be determined from the FDs. This means our candidate key is our primary key.

The candidate key and minimal key for Livestock is: {tagID}

Secondly, we see that FDs 1 and 2 violate BCNF, so we need to decompose them. The first decomposition is for FD (1). The initial relation is Livestock(tagID, lastFed, animalType, weight, diet, age, lastViolatedForHarvestedGoods, harvestable). Now we can split this into two separate relations.

Livestock_1(weight, animalType, diet) where both weight and animalType compose the primary key, and Livestock_2(tagID, lastFed, animalType, weight, age, lastViolatedForHarvestedGoods, harvestable).

From here, we still have 1 relation that has a violating FD for the Livestock relation. FD(2) is violated in Livestock_2. Therefore we need to decompose once more. Our starting relation in this case is Livestock_2(tagID, lastFed, animalType, weight, age, lastViolatedForHarvestedGoods, harvestable). Then we split:

Livestock_3(animalType, age, harvestable) and Livestock_4(tagID, lastFed, animalType, weight, age, lastViolatedForHarvestedGoods)

Now none of the relations are violating any FDs.

Therefore, we have the following three relations after normalization:

- Livestock_1(weight, animalType, diet),
- Livestock_3(animalType, age, harvestable),
- Livestock_4(tagID, lastFed, **animalType**, **weight**, **age**, lastViolatedForHarvestedGoods).

(For clarification, FKs: {**animalType**, **age**} and {**weight**, **animalType**}

Fields

FDs:

- nutrientLevels → suitableCrops (1)
- suitableCrops → pesticidesAndChemicals (2)
- suitableCrops, plotNum → capacity (3)
- plotNum → state, capacity, suitableCrops, pesticidesAndChemicals, nutrientLevels (4)

The candidate key and minimal key for Fields is: {plotNum}

We see that two FDs (1 and 2) violate BCNF, hence Fields is not in BCNF and we must decompose. Our initial relation is Fields(plotNum, nutrientLevels, suitableCrops, capacity, state, pesticidesAndChemicals).

We will start by decomposing on FD (1), which gives us the relations:

- Fields_1(nutrientLevels, suitableCrops)
- Fields_2(plotNum, nutrientLevels, capacity, state, pesticidesAndChemicals)

We still have a FD that violates BCNF, so we need to decompose again. This time, we decompose on the relation Fields_2(plotNum, nutrientLevels, capacity, state, pesticidesAndChemicals) around FD (2), which gives us the relations:

- Fields_3(suitableCrops, pesticidesAndChemicals)
- Fields_4(plotNum, nutrientLevels, capacity, state)

Now none of the relations are violating any FDs.

Therefore we have the following three relations after normalization:

- Fields_1(nutrientLevels, **suitableCrops**)
- Fields_3(suitableCrops, pesticidesAndChemicals)
- Fields_4(plotNum, **nutrientLevels**, capacity, state)

Farmers

FDs:

- yearsOfEmployment → salary (1)
- farmerID → yearsOfEmployment, salary, fullName (2)

The candidate key and minimal key for Farmers is: {farmerID}

We see that FD (1) violates BCNF, hence Farmers is not in BCNF and we must decompose. Our initial relation is Farmers(farmerID, fullName, yearsOfEmployment, salary).

Since there is only 1 FD that violates BCNF we decompose on FD (1), which gives us the relations:

- Farmers_1(yearsOfEmployment, salary)
- Farmers_2 (farmerID, fullName, **yearsOfEmployment**)

The relations are in BCNF and thus satisfying our normalization requirement.

Schema after Normalization

- Pen (
    facilityID : INTEGER,
    name: VARCHAR(30),
    age: INTEGER,
    maxCapacity: INTEGER NOT NULL,
    upkeep: INTEGER,
    typeOfAnimals: ENUM AnimalType
)
- Housing (
    facilityID : INTEGER,
    name: VARCHAR(30),
    age: INTEGER,
    maxCapacity: INTEGER,
    upkeep: INTEGER NOT NULL,
    rent: INTEGER,
    distanceToFields: INTEGER,
    distanceFromPens: INTEGER
)
- Farmers_1 (
    yearsOfEmployment : INTEGER,
    salary: INTEGER
)
- Farmers_2 (
    farmerID: INTEGER,
    fullName: VARCHAR (40) NOT NULL,
    yearsOfEmployment: INTEGER REFERENCES Farmers_1
)
- Fields_1 (
    nutrientLevels : REAL,

suitableCrops : ENUM CropType REFERENCES Fields_3
)

- Fields_3 (
  suitableCrops : ENUM CropType,
  pesticidesAndChemicals : BOOLEAN
)
- Fields_4 (
  plotNum : INTEGER,
  nutrientLevels : REAL REFERENCES Fields_1,
  capacity : INTEGER,
  state : ENUM FieldState
)
- Crops (
  cropType: ENUM CropType,
  variant: ENUM CropVariantType,
  plantStatus: ENUM CropStatusType,
  quantity: INTEGER
)
- Livestock_1 (
  weight : REAL,
  animalType : ENUM AnimalType,
  diet : ENUM CropType,
)
- Livestock_3 (
  animalType : ENUM AnimalType,
  age : INTEGER,
  harvestable : BOOLEAN
)
- Livestock_4 (
  tagID : INTEGER,
  lastFed : DATE,
  animalType : ENUM AnimalType,
  weight : REAL,
  age : INTEGER,
  lastViolatedForHarvestedGoods : DATE
  FOREIGN KEY (**weight**, **animalType**) REFERENCES Livestock_1
  FOREIGN KEY (**animalType**, **age**) REFERENCES Livestock_3
)
- LivestockProduce (
  productType : ENUM ProductType,
  quantity : INTEGER
)
- Buyers_DealsWith (
  buyerID: INTEGER,

farmerID: INTEGER NOT NULL REFERENCES Farmers_2,
address: VARCHAR(40),
name: VARCHAR(40),
phoneNumber: VARCHAR(30),
date: DATE
)

- VeterinaryRecords_Has (
  tagID: INTEGER NOT NULL REFERENCES Livestock_4,
  recordID: INTEGER,
  date: DATE,
  healthStatus: ENUM HealthStatusType
)

RELATION SETS:

- Manages_Housing (
  facilityID: INTEGER REFERENCES Housing,
  farmerID: INTEGER REFERENCES Farmers_2,
  lastMaintained: Date
)
- Manages_Pen (
  facilityID: INTEGER REFERENCES Pen,
  farmerID: INTEGER REFERENCES Farmers_2,
  lastMaintained: Date
)
- Lives (
  facilityID: INTEGER REFERENCES Housing,
  farmerID: INTEGER REFERENCES Farmers_2,
  farmersHoused: INTEGER
)
- Contains (
  facilityID: INTEGER REFERENCES Pen,
  tagID: INTEGER REFERENCES Livestock_4,
  capacityOccupied: INTEGER
)
- Nurtures (
  farmerID: INTEGER REFERENCES Farmers_2,
  tagID: INTEGER REFERENCES Livestock_4,
  waterSpent: INTEGER,
  foodSpent: INTEGER
)
- Creates (
  productType: ENUM ProductType REFERENCES LivestockProduce,
  tagID: INTEGER REFERENCES Livestock_4,
  amount: INTEGER,

dateProduced: DATE

)

- Tends (

  farmerID: INTEGER REFERENCES Farmers_2,
  plotNum: INTEGER REFERENCES Fields_4,
  waterSpent: INTEGER

)
- Buys (

  buyerID: INTEGER REFERENCES Buyers_DealWith,
  cropType: ENUM CropType,
  variant: VARCHAR(30),
  productType: ENUM ProductType REFERENCES LivestockProduce,
  quantity: INTEGER,
  price: INTEGER,
  transactionID: INTEGER
  FOREIGN KEY (**cropType**, **variant**) REFERENCES Crops

)
- IsGrowing (

  plotNum: INTEGER REFERENCES Fields_4,
  cropType: ENUM CropType,
  variant: ENUM CropVariantType,
  lastWatered: DATE,
  plantedDate: DATE,
  harvestDate: DATE,
  FOREIGN KEY (**cropType**, **variant**) REFERENCES Crops

)

SQL DDL Statements

Entities:

```
CREATE TABLE Pen (
    facilityID INTEGER PRIMARY KEY,
    name VARCHAR(30),
    age INTEGER,
    maxCapacity INTEGER NOT NULL,
    upkeep INTEGER,
    typeOfAnimals VARCHAR(10) CHECK( typeOfAnimals IN
('chicken','sheep','cow', 'pig'))
);
```

```
CREATE TABLE Housing (
    facilityID INTEGER PRIMARY KEY,
    name VARCHAR(30),
    age INTEGER,
    maxCapacity INTEGER,
    upkeep INTEGER NOT NULL,
    rent INTEGER,
    distanceToFields INTEGER,
    distanceFromPens INTEGER
);
```

```
CREATE TABLE Farmers_1 (
    yearsOfEmployment INTEGER PRIMARY KEY,
    salary INTEGER
);
```

```
CREATE TABLE Farmers_2 (
    farmerID  INTEGER PRIMARY KEY,
    fullName VARCHAR(50) NOT NULL,
    yearsOfEmployment INTEGER,
    FOREIGN KEY (yearsOfEmployment) REFERENCES Farmers_1
(yearsOfEmployment)
);
```

```
CREATE TABLE Crops (
    cropType VARCHAR(10) CHECK( cropType IN ('canola','wheat','corn',
'potatoes', 'mustard', 'coconut')),
    variant VARCHAR(10) CHECK( variant IN ('pollinated', 'hybrids')),
    plantStatus VARCHAR(10) CHECK( plantStatus IN ('planted',
'harvested')),
    quantity INTEGER,
    PRIMARY KEY (cropType, variant)
);
```

Note: Order matters for SQL query

```
CREATE TABLE Fields_3 (
    suitableCrops  VARCHAR(30) PRIMARY KEY,
    pesticidesAndChemicals  INTEGER
);
```

```
CREATE TABLE Fields_1 (
    nutrientLevels  REAL PRIMARY KEY,
    suitableCrops  VARCHAR(30),
    FOREIGN KEY (suitableCrops) REFERENCES Fields_3 (suitableCrops)
);
```

```
CREATE TABLE Fields_4 (
    plotNum  INTEGER PRIMARY KEY,
    nutrientLevels  REAL,
    capacity  INTEGER,
    state  VARCHAR(10) CHECK( state IN ('empty','growing','resting')),
    FOREIGN KEY (nutrientLevels) REFERENCES Fields_1 (nutrientLevels)
);
```

```
CREATE TABLE Livestock_1 (
    animalType  VARCHAR(10) CHECK( animalType IN ('chicken','sheep','cow',
'pig')),
    weight  REAL,
    diet  VARCHAR(10) CHECK( diet IN ('canola','wheat','corn', 'potatoes',
'mustard', 'coconut')),
    PRIMARY KEY (weight, animalType)
);
```

```
CREATE TABLE Livestock_3 (
    animalType  VARCHAR(10) CHECK( animalType IN ('chicken','sheep','cow',
'pig')),
    age  INTEGER,
    harvestable INTEGER,
    PRIMARY KEY(animalType, age)
);
```

```
CREATE TABLE Livestock_4 (
    tagID  INTEGER PRIMARY KEY,
    animalType  VARCHAR(10) CHECK( animalType IN ('chicken','sheep','cow',
'pig')),
    age  INTEGER,
    weight  REAL,
    lastFed  DATE,
    lastViolatedForHarvestedGoods DATE,
    FOREIGN KEY (weight, animalType) REFERENCES Livestock_1 (weight,
animalType),
    FOREIGN KEY (animalType, age) REFERENCES Livestock_3 (animalType, age)
);
```

*Note: This is part of many-to-many relation with full participation constraint, we have not yet seen this (*need Assertions).*

```
CREATE TABLE LivestockProduce (
    productType VARCHAR(10) CHECK( productType IN ('milk','eggs','beef',
'bacon', 'poultry', 'wool', 'manure')),
    quantity INTEGER,
    PRIMARY KEY (productType)
);
```

```
CREATE TABLE Buyers_DealsWith (
    buyerID INTEGER PRIMARY KEY,
    farmerID INTEGER NOT NULL,
    address VARCHAR(100),
    name VARCHAR(40),
    phoneNumber VARCHAR(30),
    date DATE,
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID)
);
```

```
CREATE TABLE VeterinaryRecords_Has (
    tagID INTEGER NOT NULL,
    recordID INTEGER,
    date DATE,
    healthStatus VARCHAR(30),
    PRIMARY KEY (tagID, recordID),
    FOREIGN KEY (tagID) REFERENCES Livestock_4 (tagID)
);
```

Relation Sets:

```
CREATE TABLE Manages_Housing (
    facilityID INTEGER,
    farmerID INTEGER,
    lastMaintained DATE,
    PRIMARY KEY (facilityID, farmerID),
    FOREIGN KEY (facilityID) REFERENCES Housing (facilityID),
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID)
);
```

```
CREATE TABLE Manages_Pen (
    facilityID INTEGER,
    farmerID INTEGER,
    lastMaintained DATE,
    PRIMARY KEY (facilityID, farmerID),
    FOREIGN KEY (facilityID) REFERENCES Pen (facilityID),
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID)
);
```

```
CREATE TABLE Lives (
    facilityID INTEGER,
    farmerID INTEGER,
    farmersHoused INTEGER,
    PRIMARY KEY (facilityID, farmerID),
    FOREIGN KEY (facilityID) REFERENCES Housing (facilityID),
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID)
);
```

```
CREATE TABLE Contains (
    facilityID INTEGER,
    tagID INTEGER,
    capacityOccupied INTEGER,
    PRIMARY KEY (facilityID, tagID),
    FOREIGN KEY (facilityID) REFERENCES Pen (facilityID),
    FOREIGN KEY (tagID) REFERENCES Livestock_4 (tagID)
);
```

```
CREATE TABLE Nurtures (
    farmerID INTEGER,
    tagID INTEGER,
    waterSpent INTEGER,
    foodSpent INTEGER,
    PRIMARY KEY (farmerID, tagID),
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID),
    FOREIGN KEY (tagID) REFERENCES Livestock_4 (tagID)
);
```

```
CREATE TABLE Creates (
    productType VARCHAR(10) CHECK( productType IN ('milk','eggs','beef',
'bacon', 'poultry', 'wool', 'manure')),
    tagID INTEGER,
    amount INTEGER,
    dateProduced DATE,
    PRIMARY KEY (productType, tagID),
    FOREIGN KEY (tagID) REFERENCES Livestock_4 (tagID),
    FOREIGN KEY (productType) REFERENCES LivestockProduce (productType)
);
```

```
CREATE TABLE Tends (
    farmerID INTEGER,
    plotNum INTEGER,
    waterSpent INTEGER,
    PRIMARY KEY (farmerID, plotNum),
    FOREIGN KEY (farmerID) REFERENCES Farmers_2 (farmerID),
    FOREIGN KEY (plotNum) REFERENCES Fields_4 (plotNum)
);
```

```
CREATE TABLE Buys (
    buyerID INTEGER,
    variant VARCHAR(10) CHECK( variant IN ('pollinated', 'hybrids')),
    cropType VARCHAR(10) CHECK( cropType IN ('canola','wheat','corn',
'potatoes', 'mustard', 'coconut')),
    productType VARCHAR(10) CHECK( productType IN ('milk','eggs','beef',
'bacon', 'poultry', 'wool', 'manure')),
    quantity INTEGER,
    price INTEGER,
    transactionID INTEGER,
    PRIMARY KEY (productType, buyerID, variant, cropType),
    FOREIGN KEY (buyerID) REFERENCES Buyers_DealsWith (buyerID),
    FOREIGN KEY (productType) REFERENCES LivestockProduce (productType),
    FOREIGN KEY (cropType, variant) REFERENCES Crops (cropType, variant)
);
```

```
CREATE TABLE IsGrowing (
    plotNum INTEGER,
    cropType VARCHAR(10) CHECK( cropType IN ('canola','wheat','corn',
'potatoes', 'mustard', 'coconut')),
    variant VARCHAR(10) CHECK( variant IN ('pollinated', 'hybrids')),
    lastWatered DATE,
    plantedDate DATE,
    harvestDate DATE,
    PRIMARY KEY (plotNum, variant, cropType),
    FOREIGN KEY (plotNum) REFERENCES Fields_4 (plotNum),
    FOREIGN KEY (cropType, variant) REFERENCES Crops (cropType, variant)
);
```

Insert Statements

Entities:

```
INSERT ALL
INTO Pen (facilityID, name, age, maxCapacity, upkeep, typeOfAnimals) VALUES
(2001, 'HomeOfDaBECKY', 1, 5, 120, 'cow');
INTO Pen (facilityID, name, age, maxCapacity, upkeep, typeOfAnimals) VALUES
(2002, 'YeetTown', 4, 20, 111, 'chicken');
INTO Pen (facilityID, name, age, maxCapacity, upkeep, typeOfAnimals) VALUES
(2003, 'MudPit', 8, 12, 50, 'pig');
INTO Pen (facilityID, name, age, maxCapacity, upkeep, typeOfAnimals) VALUES
(2004, 'Vegans', 2, 8, 130, 'sheep');
INTO Pen (facilityID, name, age, maxCapacity, upkeep, typeOfAnimals) VALUES
(2005, 'Chonky Charlies Cave', 3, 3, 301, 'cow');
```

```
INTO Housing (facilityID, name, age, maxCapacity,upkeep, rent,
distanceToFields, distanceFromPens) VALUES (3001, 'Harvest Homestead', 10,
4, 200, 500, 2, 1);
INTO Housing (facilityID, name, age, maxCapacity,upkeep, rent,
distanceToFields, distanceFromPens) VALUES (3002, 'The Grainery', 5, 6,
300, 700, 3, 2);
INTO Housing (facilityID, name, age, maxCapacity,upkeep, rent,
distanceToFields, distanceFromPens) VALUES (3003, 'Tillers\' Retreat', 20,
12, 500, 1000, 4, 3);
INTO Housing (facilityID, name, age, maxCapacity,upkeep, rent,
distanceToFields, distanceFromPens) VALUES (3004, 'Barnhouse Bliss', 15, 8,
400, 800, 5, 2);
INTO Housing (facilityID, name, age, maxCapacity,upkeep, rent,
distanceToFields, distanceFromPens) VALUES (3005, 'Crop Cottage', 25, 20,
600, 600, 4, 5);
```

```
INSERT INTO Crops (cropType, variant, plantStatus, quantity) VALUES
```

```
('canola', 'pollinated', 'planted', 100);
INSERT INTO Crops (cropType, variant, plantStatus, quantity) VALUES
('wheat', 'hybrids', 'harvested', 200);
INSERT INTO Crops (cropType, variant, plantStatus, quantity) VALUES
('corn', 'pollinated', 'planted', 150);
INSERT INTO Crops (cropType, variant, plantStatus, quantity) VALUES
('wheat', 'pollinated', 'planted', 50);
INSERT INTO Crops (cropType, variant, plantStatus, quantity) VALUES
('canola', 'hybrids', 'harvested', 300);
```

```
INTO Farmers_1 (yearsOfEmployment, salary) VALUES (1, 50000);
INTO Farmers_1 (yearsOfEmployment, salary) VALUES (2, 75000);
INTO Farmers_1 (yearsOfEmployment, salary) VALUES (3, 90000);
INTO Farmers_1 (yearsOfEmployment, salary) VALUES (5, 120000);
INTO Farmers_1 (yearsOfEmployment, salary) VALUES (10, 150000);
```

```
INSERT INTO Farmers_2 (farmerID, fullName, yearsOfEmployment) VALUES (1001,
'John Smith', 3);
INSERT INTO Farmers_2 (farmerID, fullName, yearsOfEmployment) VALUES (1002,
'Emily Brown', 1);
INSERT INTO Farmers_2 (farmerID, fullName, yearsOfEmployment) VALUES (1003,
'Michael Johnson', 5);
INSERT INTO Farmers_2 (farmerID, fullName, yearsOfEmployment) VALUES (1004,
'Maria Hernandez', 2);
INSERT INTO Farmers_2 (farmerID, fullName, yearsOfEmployment) VALUES (1005,
'David Kim', 10);
```

```
INSERT INTO Livestock_1 (animalType, weight, diet) VALUES ('chicken', 10,
'corn');
INSERT INTO Livestock_1 (animalType, weight, diet) VALUES ('chicken', 5,
'canola');
INSERT INTO Livestock_1 (animalType, weight, diet) VALUES ('cow', 70,
'wheat');
INSERT INTO Livestock_1 (animalType, weight, diet) VALUES ('sheep', 55,
'canola');
INSERT INTO Livestock_1 (animalType, weight, diet) VALUES ('pig', 45,
'corn');
```

```
INSERT INTO Livestock_3 (animalType, age, harvestable) VALUES ('chicken',
1, 0);
INSERT INTO Livestock_3 (animalType, age, harvestable) VALUES ('chicken',
3, 1);
INSERT INTO Livestock_3 (animalType, age, harvestable) VALUES ('cow', 4,
0);
INSERT INTO Livestock_3 (animalType, age, harvestable) VALUES ('sheep', 6,
1);
INSERT INTO Livestock_3 (animalType, age, harvestable) VALUES ('pig', 4,
1);
```

```
INSERT INTO Livestock_4(tagID, animalType, age,  weight, lastFed,
lastViolatedForHarvestedGoods) VALUES (4001, 'chicken', 1, 10,
'12/04/2022', '04/04/2022');
INSERT INTO Livestock_4(tagID, animalType, age,  weight, lastFed,
lastViolatedForHarvestedGoods) VALUES (4002, 'chicken', 3, 5,'12/04/2022',
'01/04/2022');
INSERT INTO Livestock_4(tagID, animalType, age,  weight, lastFed,
lastViolatedForHarvestedGoods) VALUES (4003, 'cow', 4, 70,'10/04/2022',
'12/04/2022')
INSERT INTO Livestock_4(tagID, animalType, age,  weight, lastFed,
lastViolatedForHarvestedGoods) VALUES (4004, 'sheep', 6, 55,'12/04/2022',
NULL);
INSERT INTO Livestock_4(tagID, animalType, age,  weight, lastFed,
lastViolatedForHarvestedGoods) VALUES (4005, 'pig', 4, 45, '11/04/2022',
NULL);
```

```
INSERT INTO Fields_3 (suitableCrops, pesticidesAndChemicals) VALUES
('corn', 0);
INSERT INTO Fields_3 (suitableCrops, pesticidesAndChemicals) VALUES
('wheat', 1);
INSERT INTO Fields_3 (suitableCrops, pesticidesAndChemicals) VALUES
('mustard', 0);
INSERT INTO Fields_3 (suitableCrops, pesticidesAndChemicals) VALUES
('potatoes', 1);
INSERT INTO Fields_3 (suitableCrops, pesticidesAndChemicals) VALUES
('canola', 1);
```

```
INSERT INTO Fields_1 (nutrientLevels, suitableCrops) VALUES (1002,
'canola');
INSERT INTO Fields_1 (nutrientLevels, suitableCrops) VALUES (300, 'corn');
INSERT INTO Fields_1 (nutrientLevels, suitableCrops) VALUES (150,
'potatoes');
INSERT INTO Fields_1 (nutrientLevels, suitableCrops) VALUES (670,
'mustard');
INSERT INTO Fields_1 (nutrientLevels, suitableCrops) VALUES (450, 'wheat');
```

```
INSERT INTO Fields_4 (plotNum, nutrientLevels, capacity, state) VALUES
(5001, 1002, 300000, 'empty');
INSERT INTO Fields_4 (plotNum, nutrientLevels, capacity, state) VALUES
(5002, 300, 1000000, 'growing');
INSERT INTO Fields_4 (plotNum, nutrientLevels, capacity, state) VALUES
(5003, 150, 3150, 'empty');
INSERT INTO Fields_4 (plotNum, nutrientLevels, capacity, state) VALUES
(5004, 670, 400, 'growing');
INSERT INTO Fields_4 (plotNum, nutrientLevels, capacity, state) VALUES
(5005, 450, 0, 'resting');
```

```
INSERT INTO LivestockProduce (productType, quantity) VALUES ('eggs', 21);
INSERT INTO LivestockProduce (productType, quantity) VALUES ('manure',
731);
INSERT INTO LivestockProduce (productType, quantity) VALUES ('wool', 121);
INSERT INTO LivestockProduce (productType, quantity) VALUES ('beef',
22);INSERT INTO LivestockProduce (productType, quantity) VALUES ('poultry',
26);
INSERT INTO LivestockProduce (productType, quantity) VALUES ('milk', 20);
```

```
INSERT INTO Buyers_DealsWith (buyerID, farmerID, address, name,
phoneNumber, date) VALUES (6001, 1002, '4523 Elmwood Avenue, Philadelphia,
PA 19103', 'Ethan Williams', '(604) 123-4567', '2023-06-11');
INSERT INTO Buyers_DealsWith (buyerID, farmerID, address, name,
phoneNumber, date) VALUES (6002, 1005, '7281 8th Street, Miami, FL 33130',
'Sophia Thompson', '(604) 555-1212', '2024-01-03');
INSERT INTO Buyers_DealsWith (buyerID, farmerID, address, name,
phoneNumber, date) VALUES (6003, 1001, '2128 Linden Avenue, Seattle, WA
98101', 'Luke Carter', '(604) 867-5309', '2022-11-24');
INSERT INTO Buyers_DealsWith (buyerID, farmerID, address, name,
phoneNumber, date) VALUES (6004, 1005, '9911 Oakwood Drive, San Francisco,
CA 94107', 'Olivia Wright', '(604) 987-6543', '2023-09-01');
INSERT INTO Buyers_DealsWith (buyerID, farmerID, address, name,
phoneNumber, date) VALUES (6005, 1005, '6316 Maple Street, Houston, TX
77030', 'Benjamin Cooper', '(604) 246-8242', '2024-03-12');
```

```
INSERT INTO VeterinaryRecords_Has (tagID, recordID, date, healthStatus)
VALUES (4001, 6001, '2022-07-12', 'healthy');
INSERT INTO VeterinaryRecords_Has (tagID, recordID, date, healthStatus)
VALUES (4002, 6002, '2022-09-05', 'sick');
INSERT INTO VeterinaryRecords_Has (tagID, recordID, date, healthStatus)
VALUES (4003, 6003, '2022-06-21', 'healthy');
INSERT INTO VeterinaryRecords_Has (tagID, recordID, date, healthStatus)
VALUES (4004, 6004, '2022-11-17', 'injured');
INSERT INTO VeterinaryRecords_Has (tagID, recordID, date, healthStatus)
VALUES (4005, 6005, '2022-08-08', 'healthy');
```

Relation Sets:

```
INSERT INTO Manages_Housing (facilityID, farmerID, lastMaintained) VALUES
(3001, 1001, '2022-01-15');
INSERT INTO Manages_Housing (facilityID, farmerID, lastMaintained) VALUES
(3002, 1002, '2022-01-25');
INSERT INTO Manages_Housing (facilityID, farmerID, lastMaintained) VALUES
(3003, 1003, '2022-01-20');
INSERT INTO Manages_Housing (facilityID, farmerID, lastMaintained) VALUES
(3004, 1004, '2022-01-12');
INSERT INTO Manages_Housing (facilityID, farmerID, lastMaintained) VALUES
(3005, 1005, '2022-01-18');
```

```
INSERT INTO Manages_Pen (facilityID, farmerID, lastMaintained) VALUES
(2001, 1001, '2022-01-15');
INSERT INTO Manages_Pen (facilityID, farmerID, lastMaintained) VALUES
(2002, 1002, '2022-01-25');
INSERT INTO Manages_Pen (facilityID, farmerID, lastMaintained) VALUES
(2003, 1003, '2022-01-20');
INSERT INTO Manages_Pen (facilityID, farmerID, lastMaintained) VALUES
(2004, 1004, '2022-01-12');
INSERT INTO Manages_Pen (facilityID, farmerID, lastMaintained) VALUES
(2005, 1005, '2022-01-18');
```

```
INSERT INTO Lives (facilityID, farmerID, farmersHoused) VALUES (3001, 1001,
4);
INSERT INTO Lives (facilityID, farmerID, farmersHoused) VALUES (3002, 1002,
6);
INSERT INTO Lives (facilityID, farmerID, farmersHoused) VALUES (3003, 1003,
12);
INSERT INTO Lives (facilityID, farmerID, farmersHoused) VALUES (3004, 1004,
8);
INSERT INTO Lives (facilityID, farmerID, farmersHoused) VALUES (3005, 1005,
20);
```

```
INSERT INTO Contains (facilityID, tagID, capacityOccupied) VALUES (2001,
4001, 20);
INSERT INTO Contains (facilityID, tagID, capacityOccupied) VALUES (2001,
4002, 30);
INSERT INTO Contains (facilityID, tagID, capacityOccupied) VALUES (2002,
4003, 25);
INSERT INTO Contains (facilityID, tagID, capacityOccupied) VALUES (2002,
4004, 35);
INSERT INTO Contains (facilityID, tagID, capacityOccupied) VALUES (2003,
4005, 40);
```

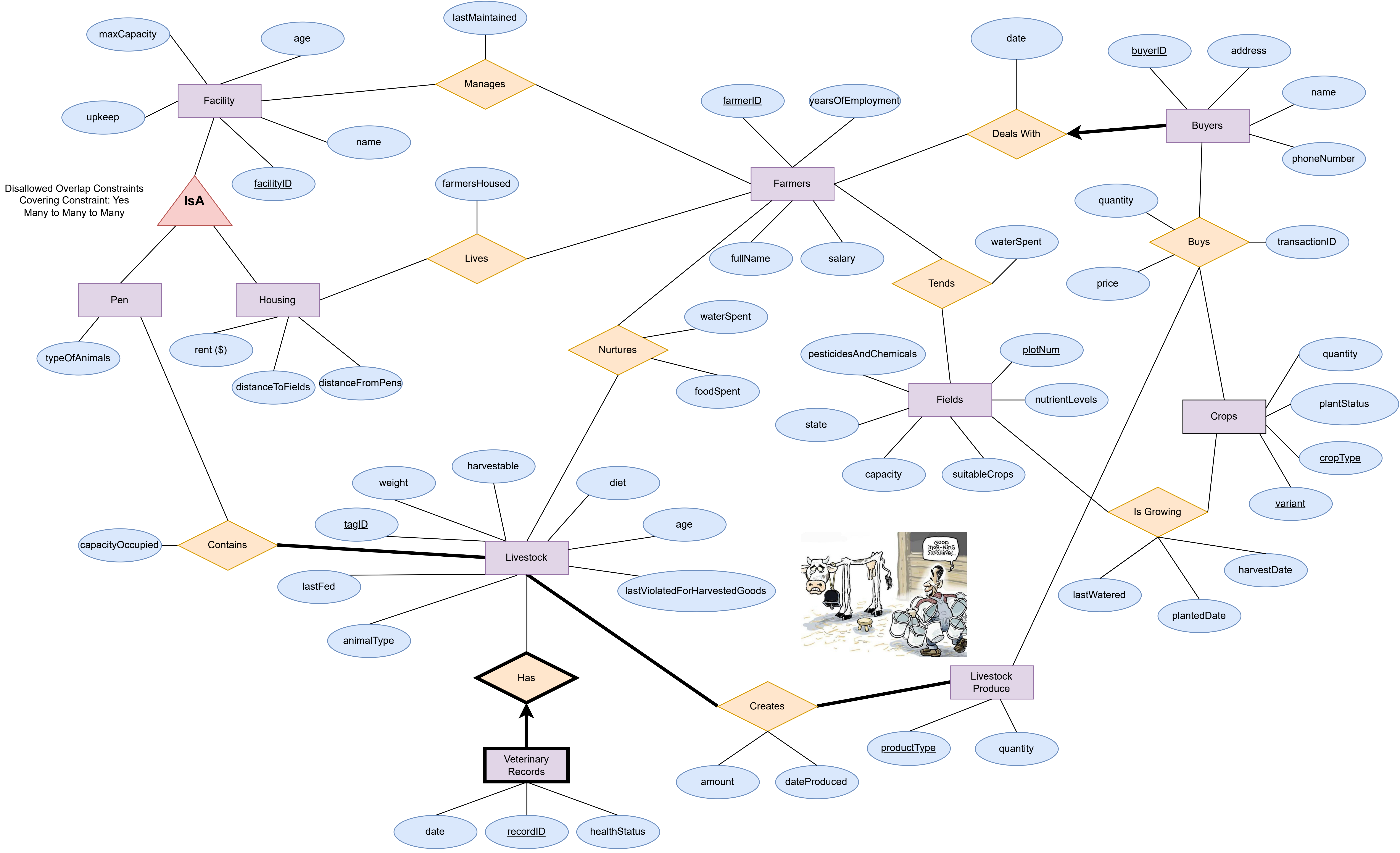
```
INSERT INTO Nurtures (farmerID, tagID, waterSpent, foodSpent) VALUES (1001,
4001, 10, 15);
INSERT INTO Nurtures (farmerID, tagID, waterSpent, foodSpent) VALUES (1002,
4002, 20, 25);
INSERT INTO Nurtures (farmerID, tagID, waterSpent, foodSpent) VALUES (1003,
4003, 15, 20);
INSERT INTO Nurtures (farmerID, tagID, waterSpent, foodSpent) VALUES (1004,
4004, 5, 10);
INSERT INTO Nurtures (farmerID, tagID, waterSpent, foodSpent) VALUES (1005,
4005, 25, 30);
```

```
INSERT INTO Creates (productType, tagID, amount, dateProduced) VALUES
('milk', 4001, 100, '2022-01-01');
INSERT INTO Creates (productType, tagID, amount, dateProduced) VALUES
('eggs', 4002, 50, '2022-01-02');
INSERT INTO Creates (productType, tagID, amount, dateProduced) VALUES
('beef', 4003, 200, '2022-01-05');
INSERT INTO Creates (productType, tagID, amount, dateProduced) VALUES
('poultry', 4004, 100, '2022-01-07');
INSERT INTO Creates (productType, tagID, amount, dateProduced) VALUES
('wool', 4005, 75, '2022-01-10');
```

```
INSERT INTO Tends (farmerID, plotNum, waterSpent) VALUES (1001, 5001, 10);
INSERT INTO Tends (farmerID, plotNum, waterSpent) VALUES (1002, 5002, 20);
INSERT INTO Tends (farmerID, plotNum, waterSpent) VALUES (1003, 5003, 15);
INSERT INTO Tends (farmerID, plotNum, waterSpent) VALUES (1004, 5004, 5);
INSERT INTO Tends (farmerID, plotNum, waterSpent) VALUES (1005, 5005, 25);
```

```
INSERT INTO Buys (buyerID, cropType, variant, productType, quantity, price,
transactionID) VALUES (6001, 'canola', 'hybrids', 'milk', 20, 50, 1);
INSERT INTO Buys (buyerID, cropType, variant, productType, quantity, price,
transactionID) VALUES (6002, 'wheat', 'pollinated', 'eggs', 30, 20, 2);
INSERT INTO Buys (buyerID, cropType, variant, productType, quantity, price,
transactionID) VALUES (6003, 'corn', 'pollinated', 'beef', 15, 100, 3);
INSERT INTO Buys (buyerID, cropType, variant, productType, quantity, price,
transactionID) VALUES (6004, 'wheat', 'pollinated', 'poultry', 10, 40, 4);
INSERT INTO Buys (buyerID, cropType, variant, productType, quantity, price,
transactionID) VALUES (6005, 'canola', 'pollinated', 'manure', 50, 5, 5);
```

```
INSERT INTO IsGrowing (plotNum, cropType, variant, lastWatered,
plantedDate, harvestDate) VALUES (5001, 'canola', 'pollinated',
'2023-01-28', '2023-01-01', '2023-04-01');
INSERT INTO IsGrowing (plotNum, cropType, variant, lastWatered,
plantedDate, harvestDate) VALUES (5002, 'wheat', 'hybrids', '2023-01-29',
'2023-01-02', '2023-04-02');
INSERT INTO IsGrowing (plotNum, cropType, variant, lastWatered,
plantedDate, harvestDate) VALUES (5003, 'corn', 'pollinated', '2023-01-30',
'2023-01-03', '2023-04-03');
 INSERT INTO IsGrowing (plotNum, cropType, variant, lastWatered,
plantedDate, harvestDate) VALUES (5004, 'wheat', 'pollinated',
'2023-01-28', '2023-01-04', '2023-04-04');
INSERT INTO IsGrowing (plotNum, cropType, variant, lastWatered,
plantedDate, harvestDate) VALUES (5005, 'canola', 'hybrids', '2023-01-29',
'2023-01-05', '2023-04-05');
```


maxCapacity
age
lastMaintained
date
buyerID
address
upkeep
Facility
Manages
farmerID
yearsOfEmployment
name
name
Deals With
Buyers
facilityID
Farmers
phoneNumber
Disallowed Overlap Constraints
Covering Constraint: Yes
Many to Many to Many
IsA
farmersHoused
quantity
fullName
salary
waterSpent
Buys
transactionID
Lives
Tends
price
Pen
Housing
waterSpent
typeOfAnimals
rent ($)
Nurtures
pesticidesAndChemicals
plotNum
quantity
distanceToFields
distanceFromPens
foodSpent
Fields
nutrientLevels
plantStatus
state
Crops
capacity
suitableCrops
cropType
harvestable
weight
diet
variant
tagID
age
Is Growing
capacityOccupied
Contains
Livestock
GOOD MOR-NING SUNSHINE!...
harvestDate
lastFed
lastViolatedForHarvestedGoods
lastWatered
plantedDate
animalType
Has
Livestock Produce
Creates
Veterinary Records
productType
quantity
amount
dateProduced
date
recordID
healthStatus