University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: 20/6/2023

Group Number: 66

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Amir Farah	39010608	o5y6m	amirsfarah@gmail.com
Engy Sadik	37749819	g0q0c	engymaged98.es@gmail.com
Zoe Yuen	99569139	v5s1d	zoeyuen100@gmail.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

2. Brief Summary

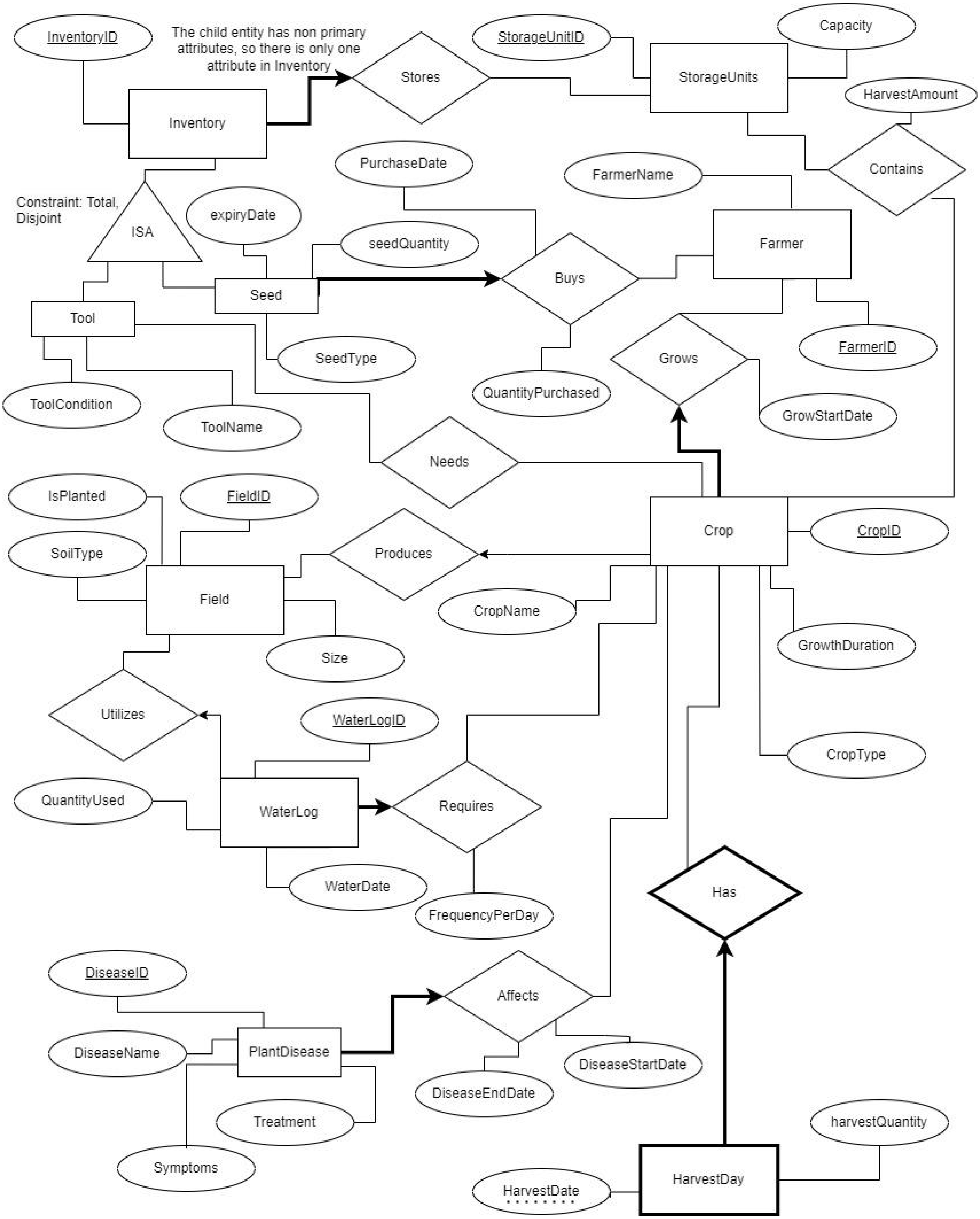
This project is a Farming and Agricultural Resource Management, designed to aid farmers to optimize their cultivation process. This application offers features for tracking crop and harvest details, monitoring seed and tool inventories, diagnosing plant diseases and efficiently managing farming spaces. Tailored for both novice and experienced farmers, the database aids in maximizing yields, boosting productivity and ensuring sustainable farming practices.

3. ER Diagram

Changes made since milestone 1

- We added a 7th entity (StorageUnits) and added the relationships it participates in (Stores and Contains), because we forgot to add our 7th entity in milestone 1
- We removed the ISA relationship for the fruit and vegetable subtypes because they did not have their own attributes or participate in fruit-specific or vegetable-specific relationships. Instead, we changed it to a CropType attribute which can indicate if the crop is a fruit, vegetable etc.
- Made some small participation constraint changes including making ‘Requires’ a many-to-one relationship between WaterLog and Crop since a crop can have multiple water logs.

ER diagram (on the next page)


The child entity has non primary attributes, so there is only one attribute in Inventory

Constraint: Total, Disjoint

4. Schema

Underline - Primary Key
Bold - Foreign Key

Entities:

1. Farmer(farmerID: integer (non null and unique), farmerName: char(20) (non null))
   - Candidate Key(s): farmerID
2. Tool(inventoryID: integer (non null and unique), **storageUnitID**: integer (non null), condition: char(20) (non null), toolName: char(20) (non null))
   - Candidate Key(s): inventoryID
3. Seed(inventoryID: integer (non null and unique), **farmerID**: integer (non null), seedQuantity: integer (non null and >= 0), seedType: char(20) (non null), purchaseDate: date (non null), quantityPurchased: integer (non null and >= 0) , expiryDate: date (not null))
   - Candidate Key(s): inventoryID
4. StorageUnits(storageUnitID: integer (non null and unique), capacity: float (non null and >= 0))
   - Candidate Key(s): storageUnitID
5. Crop(cropID: integer (non null and unique), **farmerID**: integer (non null), **fieldID**: integer (non null), growthDuration: integer (non null), cropType: char(20) (non null), cropName: char(20) (non null))
   - Candidate Key(s): cropID
6. WaterLog(waterLogID: integer (non null and unique), **cropID**: integer (non null) , **fieldID**: integer, waterDate: date (non null), waterQuantityUsed: integer (non null), frequencyPerDay: integer(non null and >= 0))
   - Candidate Key(s): waterLogID
7. Field(fieldID: integer (non null and unique), isPlanted: boolean, soilType: char(20) (non null), size: float (non null and >= 0))
   - Candidate Key(s): fieldID
8. PlantDisease(diseaseID: integer (non null and unique), **cropID**: integer (non null), diseaseName: char(20) (non null), treatment: char(20), symptoms : char(20) (non null), diseaseEndDate: date (non null), diseaseStartDate: date (non null))
   - Candidate Key(s): {diseaseID, cropID}
9. HarvestDay(**cropID**: integer (non null and unique), harvestDate: date (non null), harvestWeight: float (non null and >= 0))
   - Candidate Key(s): {cropID, harvestDate}

Relationships:

10. Contains(**storageUnitID**: integer, **cropID**: integer)
    - Candidate Key(s): {storageUnitID, cropID}
11. Needs(**cropID**: integer, **inventoryID**: integer)
    - Candidate Key(s): {cropID, inventoryID}

5. Functional Dependencies (FDs)

Farmer:

1. farmerID ⟶ farmerName

Inventory:

Tool:

1. inventoryID ⟶ toolCondition
2. inventoryID ⟶ toolName
3. inventoryID ⟶ storageUnitID

Seed:

4. inventoryID ⟶ seedQuantity
5. inventoryID ⟶ seedType
6. inventoryID ⟶ farmerID
7. inventoryID ⟶ QuantityPurchased
8. inventoryID ⟶ PurchaseDate

StorageUnit:

1. storageUnitID ⟶ capacity

Crop:

1. cropID ⟶ growthDuration
2. cropID ⟶ cropType
3. cropName⟶ cropType
4. cropID ⟶ cropName
5. cropID ⟶ farmerID
6. cropID ⟶ fieldID

WaterLog:

1. waterLogID ⟶ fieldID
2. waterLogID ⟶ waterDate
3. waterLogID ⟶ quantityUsed
4. waterLogID ⟶ CropID
5. waterLogID ⟶ frequencyPerDay

FieldID:

1. fieldID ⟶ isPlanted
2. fieldID ⟶ soilType
3. fieldID ⟶ size

PlantDisease:

1. diseaseID ⟶ diseaseName
2. diseaseID ⟶ treatment
3. diseaseID ⟶ symptoms
4. symptoms ⟶ treatment
5. diseaseName ⟶ treatment
6. diseaseName ⟶ symptoms
7. diseaseID ⟶ diseaseEndDate
8. diseaseID ⟶ diseaseStartDate
9. diseaseID ⟶ cropID

HarvestDay:

1. cropID, harvestDate ⟶ quantity

Contains:

1. storageUnitID ⟶ cropID
2. cropID ⟶ storageUnitID

Needs:

1. cropID ⟶ inventoryID
2. inventoryID ⟶ cropID

Requires:

1. cropID, waterLogID ⟶ frequencyPerDay
2. waterLogID ⟶ frequencyPerDay
3. cropID ⟶ waterLogID

6. Normalization in BCNF

Underline - Primary Key

Bold - Foreign Key

1. Farmer(farmerID: integer (non null and unique), farmerName: char(20) (non null))
    a. Candidate Key(s): farmerID
    b. farmerID ⟶ farmerName

farmerID^{+} = {farmerID, farmerName}

farmerID is already a superkey, hence relation Farmer is in BCNF.

2. Tool(inventoryID: integer (non null and unique), **storageUnitID**: integer (non null), condition: char(20) (non null), toolName: char(20) (non null))
    a. Candidate Key(s): inventoryID

b. inventoryID → toolName
c. inventoryID → condition
d. inventoryID → storageUnitID

inventoryID$^+$ = {inventoryID, toolName, condition, storageUnitID}
inventoryID is already the superkey, hence relation Tool is in BCNF

3. Seed(<u>inventoryID</u>: integer (non null and unique), **farmerID**: integer (non null), seedQuantity: integer (non null and >= 0), seedType: char(20) (non null), purchaseDate: date (non null), quantityPurchased: integer (non null and >= 0) , expiryDate: date (not null))
    a. Candidate Key(s): inventoryID
    b. inventoryID → seedQuantity
    c. inventoryID → seedType
    d. inventoryID → farmerID
    e. inventoryID → quantityPurchased
    f. inventoryID → purchaseDate
    g. inventoryID → expiryDate
    h. PurchaseDate → expiryDate

PurchaseDate$^+$ = {PurchaseDate, expiryDate}
inventoryID$^+$ = {inventoryID, seedQuantity, seedType, farmerID, purchaseDate, quantityPurchased, expiryDate}

Decompose on PurchaseDate→ expiryDate
Seed$_1$(<u>PurchaseDate</u>: date, expiryDate: date)
Seed$_2$(<u>inventoryID</u>: integer, seedQuantity: integer, seedType: char(20), **farmerID**: integer, **PurchaseDate**: date, quantityPurchased: integer)

Both Seed$_1$ and Seed$_2$ are in BCNF

4. StorageUnits(<u>storageUnitID</u>: integer (non null and unique), capacity: float (non null and >= 0))
    a. Candidate Key(s): storageUnitID
    b. storageUnitID → capacity

storageUnitID$^+$ = { storageUnitID, float}
storageUnitID is already the superkey, hence relation StorageUnit is in BCNF

5. Crop(<u>cropID</u>: integer (non null and unique), **farmerID**: integer (not null), **fieldID**: integer (not null), growthDuration: integer (non null), cropType: char(20) (non null), cropName: char(20) (non null))
    a. Candidate Key(s): No Candidate Key
    b. cropID → growthDuration

c. cropID ⟶ cropType
d. cropName⟶ cropType
e. cropID ⟶ cropName
f. cropID ⟶ farmerID
g. cropID ⟶ fieldID

cropName$^+$ = { cropName, cropType}
cropID$^+$ = { cropID, growthDuration, cropType, cropName, farmerID, fieldID}

Decompose on cropName⟶ cropType
Crop$_1$(cropName: char(20), cropType: char(20))
Crop$_2$(cropID: integer, **cropName**: char(20), growthDuration: integer, **farmerID**: integer, **fieldID**: integer)

Both Crop$_1$ and Crop$_2$ are in BCNF

6. WaterLog(waterLogID: integer (non null and unique), **cropID**: integer (non null) , **fieldID**: integer, waterDate: date (non null), waterQuantityUsed: integer (non null), frequencyPerDay: integer(non null and >= 0))
    a. Candidate Key(s): No Candidate Key
    b. waterLogID ⟶ fieldID
    c. waterLogID ⟶ waterDate
    d. waterLogID ⟶ quantityUsed
    e. waterLogID ⟶ CropID
    f. waterLogID ⟶ frequencyPerDay

waterLogID$^+$ = {waterLogID, fieldID, CropID, waterDate, quantityUsed, frequencyPerDay}
waterLogID is already the superkey, hence relation Waterlog in BCNF

7. Field(fieldID: integer (non null and unique), isPlanted: boolean, soilType: char(20) (non null), size: float (non null and >= 0))
    a. Candidate Key(s): No Candidate Key
    b. fieldID ⟶ isPlanted
    c. fieldID ⟶ soilType
    d. fieldID ⟶ size

fieldID$^+$ = {fieldID, isPlanted, soilType, size}
fieldID is already the superkey, hence relation Field is in BCNF

8. PlantDisease(diseaseID: integer (non null and unique), **cropID**: integer (not null), diseaseName: char(20) (non null), treatment: char(20), symptoms: char(20), diseaseEndDate: date, diseaseStartDate: date)
    a. Candidate Key(s): No Candidate Key

b. diseaseID → diseaseName
c. diseaseID → treatment
d. diseaseID → symptoms
e. symptoms → treatment
f. diseaseName → treatment
g. diseaseName → symptoms
h. diseaseID → diseaseEndDate
i. diseaseID → diseaseStartDate

symptoms$^+$ = {symptoms, treatment}
diseaseName$^+$ = {diseaseName, treatment, symptoms}
diseaseID$^+$ = {diseaseID, diseaseName, treatment, symptoms, diseaseEndDate, diseaseStartDate}

Decompose on symptoms → treatment
PlantDisease$_1$(symptoms: char(20), treatment: char(20))
PlantDisease$_2$(diseaseID: integer (non null and unique), **symptoms**: char(20), diseaseName: char(20) (non null) diseaseEndDate: date, diseaseStartDate: date)

Decompose on PlantDisease$_2$ on diseaseName → symptoms
PlantDisease$_3$(diseaseName: char(20), **symptoms**: char(20))
PlantDisease$_4$(diseaseID: integer, **diseaseName**: char(20), diseaseStartDate: integer, diseaseEndDate: integer)

PlantDisease$_1$, PlantDisease$_3$ and PlantDisease$_4$ are in BCNF

9. HarvestDay(**cropID**: integer, harvestDate: integer (non null), harvestWeight: float (non null and >= 0))
    a. Candidate Key(s): No Candidate Key
    b. cropID, harvestDate → quantity

cropID, harvestDate$^+$ = {cropID, harvestDate, quantity}
cropID, harvestDate is already a superkey, hence relation HarvestDay is in BCNF.

10. Contains(**storageUnitID**: integer, **cropID**: integer)
    a. Candidate Key(s): No Candidate Key
    b. cropID → storageUnitID

cropID$^+$ = {cropID, storageUnitID}
cropID is already a superkey, hence relation Contains is in BCNF.

11. Needs(**cropID**: integer, **inventoryID**: integer)
    a. Candidate Key(s): No Candidate Key

b. inventoryID ⟶ cropID

inventoryID$^+$ = {inventoryID, cropID}
inventoryID is already a superkey, hence relation Needs is in BCNF

7. SQL DDL statements

```
CREATE TABLE Farmer (
    farmerID INT NOT NULL UNIQUE,
    farmerName VARCHAR NOT NULL,
    PRIMARY KEY (farmerID)
);

CREATE TABLE Tool (
    inventoryID INT NOT NULL UNIQUE,
    storageUnitID INT NOT NULL UNIQUE,
    condition VARCHAR NOT NULL,
    toolName VARCHAR NOT NULL,
    PRIMARY KEY (inventoryID),
    FOREIGN KEY (storageUnitID) REFERENCES StorageUnit(storageUnitID)
        ON DELETE SET NULL
        ON UPDATE CASCADE
);

//we assume that all seeds have the same time period between purchase date and expiry date
CREATE TABLE Seed1 (
    PurchaseDate DATE NOT NULL,
    expiryDate DATE NOT NULL,
    PRIMARY KEY (PurchaseDate)
);

CREATE TABLE Seed2 (
    inventoryID INT NOT NULL UNIQUE,
    seedQuantity INT NOT NULL CHECK (seedQuantity >= 0),
    seedType VARCHAR(20) NOT NULL,
    farmerID INT NOT NULL,
    PurchaseDate DATE NOT NULL,
    quantityPurchased INT NOT NULL CHECK (quantityPurchased >= 0),
    PRIMARY KEY (inventoryID),
    FOREIGN KEY (farmerID) REFERENCES Farmer(farmerID)
        ON DELETE SET CASCADE
        ON UPDATE CASCADE,
        FOREIGN KEY (PurchaseDate) REFERENCES Seed1(PurchaseDate)
);


CREATE TABLE StorageUnits (
        storageUnitID INT NOT NULL UNIQUE,
        capacity FLOAT NOT NULL CHECK (capacity >= 0),
        PRIMARY KEY (storageUnitID)
);

CREATE TABLE Crop1 (
        cropName VARCHAR NOT NULL,
        cropType VARCHAR NOT NULL,
        PRIMARY KEY (cropName)
);

CREATE TABLE Crop2 (
        cropID INT NOT NULL UNIQUE,
        cropName VARCHAR NOT NULL,
        growthDuration INT NOT NULL,
        farmerID INT NOT NULL UNIQUE,
        fieldID INT NOT NULL UNIQUE,
        diseaseID INT NOT NULL UNIQUE,
        PRIMARY KEY (cropID),
        FOREIGN KEY (cropName) REFERENCES Crop1(cropName),
        FOREIGN KEY (farmerID) REFERENCES Farmer(farmerID),
                ON DELETE SET NULL
                ON UPDATE CASCADE
        FOREIGN KEY (fieldID) REFERENCES Field(fieldID),
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE WaterLog(
        waterLogID INT NOT NULL UNIQUE,
        fieldID INT NOT NULL,
        waterDate DATE NOT NULL,
        waterQuantityUsed INT NOT NULL,
        frequencyPerDay INT NOT NULL CHECK (frequencyPerDay >= 0),
        PRIMARY KEY (waterLogID),
        FOREIGN KEY (fieldID) REFERENCES Field(fieldID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
        FOREIGN KEY (cropID) REFERENCES Crop(cropID)
                ON DELETE SET CASCADE
                ON UPDATE CASCADE
):

CREATE TABLE Field (
        fieldID INT NOT NULL UNIQUE,
        isPlanted BOOLEAN,
        soilType VARCHAR NOT NULL,
        size FLOAT NOT NULL CHECK (size >= 0),
        PRIMARY KEY (fieldID)
);

CREATE TABLE PlantDisease1 (
        symptoms VARCHAR NOT NULL,
        treatment VARCHAR,
        PRIMARY KEY (symptoms)
);

CREATE TABLE PlantDisease3 (
        diseaseName VARCHAR NOT NULL,
        symptoms VARCHAR NOT NULL,
        PRIMARY KEY (diseaseName),
        FOREIGN KEY (symptoms) REFERENCES PlantDisease1(symptoms)
);

CREATE TABLE PlantDisease4 (
        diseaseID INT NOT NULL UNIQUE,
        cropID INT NOT NULL
        diseaseName VARCHAR NOT NULL,
        diseaseStartDate DATE NOT NULL,
        diseaseEndDate DATE NOT NULL,
        PRIMARY KEY (diseaseID),
        FOREIGN KEY (diseaseName) REFERENCES PlantDisease3(diseaseName)
        FOREIGN KEY (cropID) REFERENCES Crop2(cropID)
                ON DELETE CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE HarvestDay (
        cropID INT NOT NULL UNIQUE,
        harvestDate DATE NOT NULL,
        harvestWeight FLOAT NOT NULL CHECK (harvestWeight >= 0),
        PRIMARY KEY (cropID, harvestDate),
        FOREIGN KEY (cropID) REFERENCES Crop(cropID)
                ON DELETE SET CASCADE
                ON UPDATE CASCADE
);

CREATE TABLE Contains (
        storageUnitID INT NOT NULL,
        cropID INT NOT NULL,
        PRIMARY KEY (storageUnitID, cropID),
        FOREIGN KEY (storageUnitID) REFERENCES StorageUnits(storageUnitID),
        FOREIGN KEY (cropID) REFERENCES Crop(cropID)
);

CREATE TABLE Needs (
        cropID INT NOT NULL,
        inventoryID INT NOT NULL,
        PRIMARY KEY (cropID, inventoryID),
        FOREIGN KEY (cropID) REFERENCES Crop(cropID),
        FOREIGN KEY (inventoryID) REFERENCES Inventory(inventoryID)
);
```

8. INSERT statements

```
INSERT INTO Farmer (farmerID, farmerName) VALUES
(1, 'John Doe'),
(2, 'Ahmed Smith'),
(3, 'Robert Khaled'),
(4, 'Emily mahmood'),
(5, 'Abdullah Green'),
(6, 'Anna aswad');

INSERT INTO Tool (inventoryID, storageUnitID, condition, toolName) VALUES
(1001, 1, 'Good', 'Shovel'),
(1002, 2, 'Average', 'Hoe'),
(1003, 3, 'Excellent', 'Rake'),
(1004, 4, 'Good', 'Tiller'),
(1005, 5, 'Fair', 'Wheelbarrow'),
(1006, 6, 'Excellent', 'Pruner');
```

```
INSERT INTO Seed1 (PurchaseDate, expiryDate) VALUES
('2023-01-01', '2024-01-01'),
('2023-02-10', '2024-02-10'),
('2023-03-15', '2024-03-15'),
('2023-05-20', '2024-05-20'),
('2023-07-25', '2024-07-25'),
('2023-08-30', '2024-08-30');

INSERT INTO Seed2 (inventoryID, seedQuantity, seedType, farmerID, PurchaseDate, quantityPurchased)
VALUES
(2001, 100, 'Wheat', 1, '2023-01-01', 500),
(2002, 150, 'Corn', 2, '2023-02-10', 400),
(2003, 75, 'Rice', 3, '2023-03-15', 450),
(2004, 50, 'Barley', 4, '2023-05-20', 300),
(2005, 80, 'Soybean', 5, '2023-07-25', 350);

INSERT INTO StorageUnits (storageUnitID, capacity) VALUES
(1, 1000.5),
(2, 800.0),
(3, 500.25),
(4, 650.75),
(5, 750.0);


INSERT INTO Crop1 (cropName, cropType) VALUES
('Wheat', 'Grain'),
('Corn', 'Cereal'),
('Rice', 'Grain'),
('Barley', 'Cereal'),
('Soybean', 'Legume');

INSERT INTO Crop2 (cropID, cropName, growthDuration, farmerID, fieldID, diseaseID) VALUES
(3001, 'Wheat', 120, 1, 1, 1),
(3002, 'Corn', 90, 2, 2, 2),
(3003, 'Rice', 150, 3, 3, 3),
(3004, 'Barley', 75, 4, 4, 4),
(3005, 'Soybean', 100, 5, 5, 5);

INSERT INTO WaterLog (waterLogID, fieldID, waterDate, waterQuantityUsed, frequencyPerDay) VALUES
(1, 1, '2023-04-01', 50, 2),
(2, 2, '2023-04-02', 45, 2),
(3, 3, '2023-04-03', 55, 1),
(4, 4, '2023-04-04', 40, 3),
(5, 5, '2023-04-05', 60, 1);

INSERT INTO Field (fieldID, isPlanted, soilType, size) VALUES
(1, TRUE, 'Loamy', 5.0),
(2, TRUE, 'Sandy', 4.5),
(3, TRUE, 'Clay', 6.0),
(4, TRUE, 'Peaty', 3.5),
(5, TRUE, 'Chalky', 4.0);

INSERT INTO PlantDisease1 (symptoms, treatment) VALUES
('Yellowing leaves', 'Nutrient supplements'),
('Stunted growth', 'Pesticides'),
('Black mold on leaves', 'Antifungal sprays and reduced humidity'),
('Wilting stems', 'Proper watering and drainage'),
('Curled leaves', 'Pest control and reduction in environmental stress');

INSERT INTO PlantDisease3 (diseaseName, symptoms) VALUES
('Blight', 'Yellowing leaves'),
('Fungus', 'Stunted growth'),
('Mildew', 'White powdery spots on leaves and stems'),
('Leaf Spot', 'Brown or black spots on leaves'),
('Root Knot', 'Swelling or knots in roots');

INSERT INTO PlantDisease4 (diseaseID, cropID, diseaseName, diseaseStartDate, diseaseEndDate) VALUES
(1, 3001, 'Blight', '2023-05-01', '2023-05-10'),
(2, 3002, 'Fungus', '2023-05-05', '2023-05-15'),
(3, 3003, 'Mildew', '2023-06-01', '2023-06-11'),
(4, 3004, 'Leaf Spot', '2023-06-07', '2023-06-17'),
(5, 3005, 'Root Knot', '2023-06-15', '2023-06-25');

INSERT INTO HarvestDay (cropID, harvestDate, harvestWeight) VALUES
(3001, '2023-09-01', 500.0),
(3002, '2023-08-20', 450.0),
(3003, '2023-10-01', 650.0),
(3004, '2023-07-30', 400.0),
(3005, '2023-08-10', 550.0);

INSERT INTO Contains (storageUnitID, cropID) VALUES
(1, 3001),
(2, 3002),
(3, 3003),
(4, 3004),
(5, 3005);

INSERT INTO Needs (cropID, inventoryID) VALUES
(3001, 1),
(3002, 2),
(3003, 3),
(3004, 4),
(3005, 5);
```