

CPSC 304 Project Cover Page

Milestone #2

Date: October 22nd, 2025

Group Number: 31

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|-------------|----------------|-------------------|--------------------------|
| Cameron Jim | 64991375 | c5x9i | cjim02@student.ubc.ca |
| Ryan Vu | 93118289 | x7m7j | ryanvu@student.ubc.ca |
| Wit Lin | 46645636 | k6g0d | wlin18@student.ubc.ca |

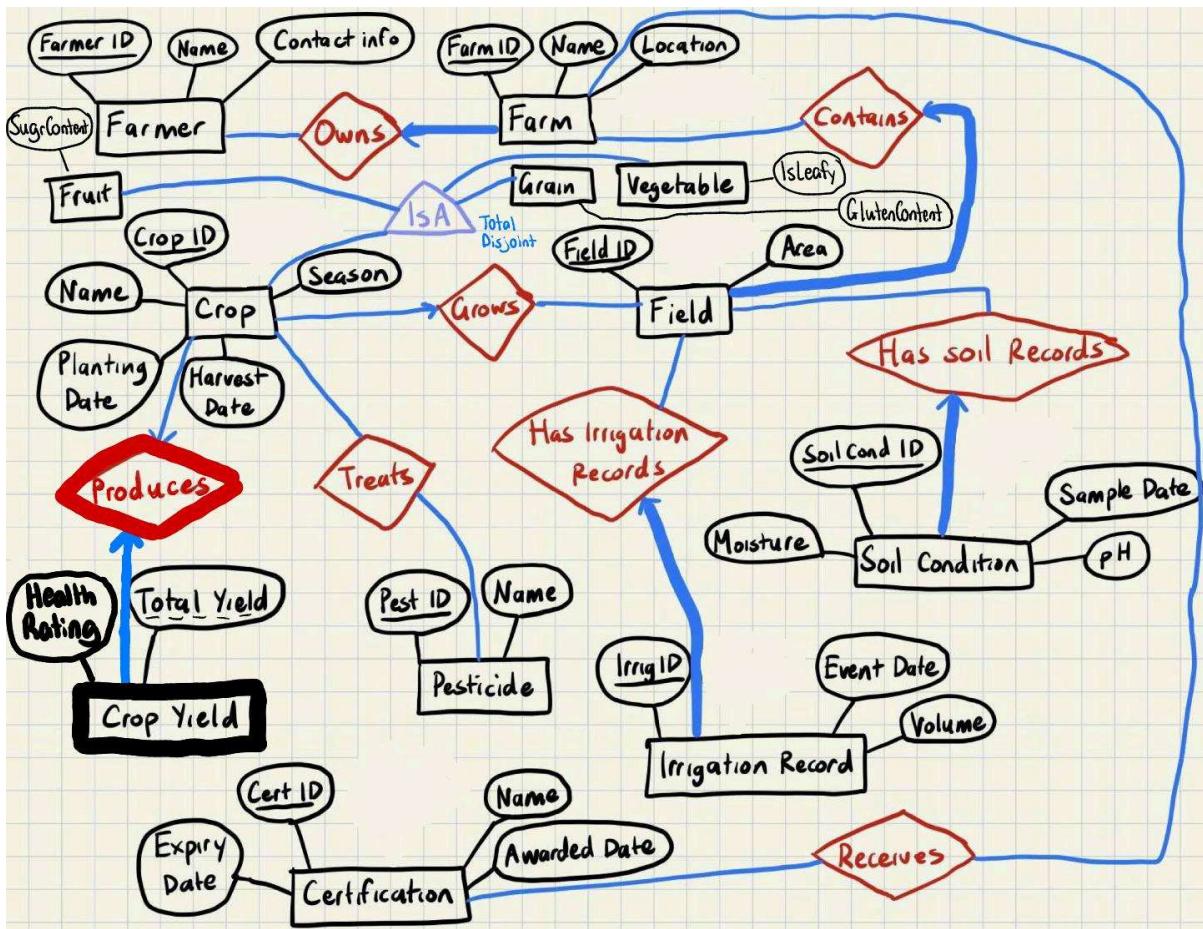
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address, 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.) Our project is based in the sustainable agriculture/farm management space. The system will help farms keep organized records about the crops they grow and the certifications their farm has (ie. organic, non-GMO, etc.). Data is seasonal and can change over time, thus a centralized platform will help farmers review yearly crop records.

3.) ER Diagram Changes:

- Added attributes to fully define IsA hierarchy (and the addition of total, disjoint)
- Completed weak entity drawing on diagram as well, made Total Yield a partial key.
- Removed some attributes since they were really foreign keys, and did not need to be associated with that specific entity.
- A health rating attribute was added to the entity Crop Yield to meet the requirement of attributes requirement of Milestone 1.



4.)

Farmer(FarmerID: INT, Name: VARCHAR(20), ContactInfo: VARCHAR(80))

- PK: FarmerID, CK: FarmerID

OwnsFarm(FarmID: INT, Name: VARCHAR(20), Location: VARCHAR(20), FarmerID: INT)

- PK: FarmID, FK: FarmerID, CK: FarmID, NOT NULL: FarmerID

ContainsField(FieldID: INT, FarmID: INT, Area: INT)

- PK: FieldID, FK: FarmID, CK: FieldID, NOT NULL: FarmID

GrowsCrop(CropID: INT, FieldID: INT, Season: VARCHAR(20), Name: VARCHAR(60),

PlantingDate: DATE, HarvestDate: DATE, FarmID: INT)

- PK: CropID, CK: CropID, FK: FieldID, FarmID

Grain(CropID: INT, GlutenContent: DECIMAL(10,2))

- PK: CropID, CK: CropID

Vegetable(CropID: INT, IsLeafy: BOOLEAN)

- PK: CropID, CK: CropID

Fruit(CropID: INT, SugarContent: DECIMAL(10,2))

- PK: CropID, CK: CropID

CropYieldProduces(CropID: INT, Total_Yield: DECIMAL(10,2), Health_Rating: INT)

- PK: CropID, FK: CropID, CK: CropID

Treats(CropID: INT, PestID: INT)

- PK: (CropID, PestID), FK: (CropID, PestID), CK: (CropID, PestID)

Pesticide(PestID: INT, Name: VARCHAR(60))

- PK: PestID, CK: PestID

IrrigationRecords(IrrigID: INT, FieldID: INT, EventDate: DATE, Volume: DECIMAL(10,2))

- PK: IrrigID, CK: IrrigID, FK: FieldID, NOT NULL: FieldID

SoilRecords(SoilCondID: INT, FieldID: INT, SampleDate: DATE, pH: DECIMAL(10,2), Moisture:

DECIMAL(10,2), Area: INT)

- PK: SoilCondID, CK: SoilCondID, FK: FieldID, Area, NOT NULL: FieldID

Receives(FarmID: INT, CertID: INT)

- PK: (FarmID, CertID), FK: (FarmID, FK: CertID), CK: (FarmID, FK: CertID)

Certification(CertID: INT, Name: VARCHAR(80), AwardedDate: DATE, Expiry Date: DATE)

- PK: CertID, CK: CertID

5.) Functional dependencies:

Farmer Entity:

Farmer ID → Name, Contact Info

Contact Info → Name

Farm Entity:

Farm ID → Name, Location, Farmer ID

- **Combined “Owns” Relation (many-to-one):**

- Farm ID → Farmer ID

Crop Entity:

Crop ID → Name, Field ID, Season, Planting Date, Harvest Date

Planting date → Season

Name → Planting Date, Harvest Date, Season

- **Combined “Grows” Relation (many-to-one):**

- Crop ID → Field ID

Pesticide entity:

Pest ID → Name

Field Entity:

Field ID → Farm ID, Area

- **Combined “Contains” Relation (many-to-one):**

- Field ID → Farm ID

Soil Condition Entity:

Soil Cond ID → Moisture, Sample Date, pH

Sample Date, pH → Moisture

Irrigation Record Entity:

Irrig ID → Event Date, Volume, Field ID

- **Combined “Has Irrigation Records” Relation (many-to-one):**

- Irrig ID → Field ID

Certification Entity:

Cert ID → Expiry Date, Name, Awarded Date

Awarded Date → Expiry Date

Has Soil Records Relation (many-to-one):

Soil Cond ID → Field ID

Produces Relation (one-to-one):

Crop ID → Health Rating, Total Yield

6.) Normalization to BCNF:

1. Decompose (SampleDate, pH) → Moisture on SoilRecords

$R_1 = (\underline{\text{SampleDate}}, \underline{\text{pH}}, \text{Moisture})$

$R_2 = (\underline{\text{SoilCondID}}, \text{FieldID}, \text{SampleDate}, \text{pH}), \text{Area}$

2. Decompose Awarded Date → Expiry Date on Certification

$R_1 = (\underline{\text{AwardedDate}}, \text{ExpiryDate})$

$R_2 = (\underline{\text{CertID}}, \text{Name}, \text{AwardedDate})$

3. Decompose Planting date → Season && Name → Planting Date, Harvest Date,

Season on GrowsCrops

$R_1 = (\underline{\text{PlantingDate}}, \text{Season})$

$R_2 = (\underline{\text{Name}}, \text{PlantingDate}, \text{HarvestDate})$

$R_3 = (\underline{\text{CropID}}, \text{FieldID}, \text{Name}, \text{FarmID})$

4. Decompose Contact Info → Name on Farmer

$R_1 = (\underline{\text{ContactInfo}}, \text{Name})$

$R_2 = (\underline{\text{FarmerID}}, \text{ContactInfo})$

*** Bold = PK

*** Underline = CK

Schema after normalization:

ContactInfoName(ContactInfo: VARCHAR(80), Name: VARCHAR(20))

- PK: ContactInfo, CK: ContactInfo

Farmer(FarmerID: INT, ContactInfo: VARCHAR(80))

- PK: FarmerID, CK: FarmerID

OwnsFarm(FarmID: INT, Name: VARCHAR(20), Location: VARCHAR(20), FarmerID: INT)

- PK: FarmID, FK: FarmerID, CK: FarmID, NOT NULL: FarmerID

ContainsField(FieldID: INT, FarmID: INT, Area: INT)

- PK: FieldID, FK: FarmID, CK: FieldID, NOT NULL: FarmID

SeasonByPlantDate(PlantingDate: DATE, Season: VARCHAR(20))

- PK: PlantingDate, CK: PlantingDate

CropType(Name: VARCHAR(60), PlantingDate: DATE, HarvestDate: DATE)

- PK: Name, CK: Name, FK: PlantingDate

GrowsCrop(CropID: INT, FieldID: INT, Name: VARCHAR(60))

- PK: CropID, CK: CropID, FK: FieldID, Name

Grain(CropID: INT, GlutenContent: DECIMAL(10,2))

- PK: CropID, CK: CropID

Vegetable(CropID: INT, IsLeafy: BOOLEAN)

- PK: CropID, CK: CropID

Fruit(CropID: INT, SugarContent: DECIMAL(10,2))

- PK: CropID, CK: CropID

CropYieldProduces(CropID: INT, Total_Yield: DECIMAL(10,2), Health_Rating: INT)

- PK: CropID, FK: CropID, CK: CropID

Treats(CropID: INT, PestID: INT)

- PK: (CropID, PestID), FK: (CropID, PestID), CK: (CropID, PestID)

Pesticide(PestID: INT, Name: VARCHAR(60))

- PK: PestID, CK: PestID

Treats(CropID: INT, PestID: INT)

- PK: (CropID, PestID), CK: (CropID, PestID), FK: CropID, PestID

IrrigationRecords(IrrigID: INT, FieldID: INT, EventDate: DATE, Volume: DECIMAL(10,2))

- PK: IrrigID, CK: IrrigID, FK: FieldID, NOT NULL: FieldID

MoistureByChemistry(SampleDate: DATE, pH: DECIMAL(4,2), Moisture: DECIMAL(6,2))

- PK: (SampleDate, pH), CK: (SampleDate, pH)

SoilRecords(SoilCondID: INT, FieldID: INT, SampleDate: DATE, pH: DECIMAL(4,2))

- PK: SoilCondID, CK: SoilCondID, FK: FieldID, NOT NULL: FieldID

AwardExpiry(AwardedDate: DATE, ExpiryDate: DATE)

- PK: AwardedDate, CK: AwardedDate

Receives(FarmID: INT, CertID: INT)

- PK: (FarmID, CertID), FK: (FarmID, FK: CertID), CK: (FarmID, FK: CertID)

Certification(CertID: INT, Name: VARCHAR(80), AwardedDate: DATE)

- PK: CertID, CK: CertID

7.) We only use ON DELETE CASCADE where the child doesn't make sense without the parent. That applied to Fruit, Vegetable, Grain, CropYieldProduces, and the Treats tables. Everywhere else, we leave the default behaviour so deletes are blocked if something still references the row. We also use foreign keys to enforce the FDs we normalized.

```
-- farmer table
-- 1) ContactInfoName(ContactInfo, Name)
-- 2) Farmer(FarmerID, ContactInfo)

-- contact info name table
CREATE TABLE ContactInfoName (
    ContactInfo  VARCHAR(80) PRIMARY KEY,
    Name          VARCHAR(20) NOT NULL
);

-- farmer table
CREATE TABLE Farmer (
    FarmerID      INT PRIMARY KEY,
    ContactInfo   VARCHAR(80) NOT NULL,
    CONSTRAINT fk_farmer_contact
        FOREIGN KEY (ContactInfo) REFERENCES ContactInfoName(ContactInfo),
    CONSTRAINT uq_farmer_contact UNIQUE (ContactInfo)
);

-- owns farm table
CREATE TABLE OwnsFarm (
    FarmID        INT PRIMARY KEY,
    Name          VARCHAR(20) NOT NULL,
    Location      VARCHAR(20) NOT NULL,
    FarmerID     INT NOT NULL,
    CONSTRAINT fk_farm_farmer
        FOREIGN KEY (FarmerID) REFERENCES Farmer(FarmerID)
);

-- contains field table
CREATE TABLE ContainsField (
    FieldID       INT PRIMARY KEY,
    FarmID        INT NOT NULL,
    Area          INT,
    CONSTRAINT fk_field_farm
        FOREIGN KEY (FarmID) REFERENCES OwnsFarm(FarmID)
);
```

```

-- grows crop table
-- 1) SeasonByPlantDate(PlantingDate → Season)
-- 2) CropType(Name → PlantingDate, HarvestDate, Season)
-- 3) GrowsCrop(CropID, FieldID, Name)

-- season by plant date table
CREATE TABLE SeasonByPlantDate (
    PlantingDate DATE PRIMARY KEY,
    Season        VARCHAR(20) NOT NULL
) ;

-- crop type table
CREATE TABLE CropType (
    Name          VARCHAR(60) PRIMARY KEY,
    PlantingDate  DATE NOT NULL,
    HarvestDate   DATE NOT NULL,
    CONSTRAINT fk_croptype_plantdate
        FOREIGN KEY (PlantingDate) REFERENCES
SeasonByPlantDate(PlantingDate)
) ;

-- grows crop table
CREATE TABLE GrowsCrop (
    CropID      INT PRIMARY KEY,
    FieldID     INT NOT NULL,
    Name         VARCHAR(60) NOT NULL,
    CONSTRAINT fk_crop_field
        FOREIGN KEY (FieldID) REFERENCES ContainsField(FieldID),
    CONSTRAINT fk_crop_name
        FOREIGN KEY (Name) REFERENCES CropType(Name)
) ;

-- grain table (ISA subtype of crop)
CREATE TABLE Grain (
    CropID      INT PRIMARY KEY,
    GlutenContent DECIMAL(10,2),
    CONSTRAINT fk_grain_crop
        FOREIGN KEY (CropID) REFERENCES GrowsCrop(CropID)
            ON DELETE CASCADE
) ;

-- vegetable table (ISA subtype of crop)
CREATE TABLE Vegetable (

```

```

CropID      INT PRIMARY KEY,
IsLeafy    NUMBER(1) NOT NULL,
CONSTRAINT ck_veg_bool CHECK (IsLeafy IN (0,1)),
CONSTRAINT fk_veg_crop
    FOREIGN KEY (CropID) REFERENCES GrowsCrop(CropID)
    ON DELETE CASCADE
);

-- fruit table (ISA subtype of crop)
CREATE TABLE Fruit (
    CropID      INT PRIMARY KEY,
    SugarContent DECIMAL(10,2),
    CONSTRAINT fk_fruit_crop
        FOREIGN KEY (CropID) REFERENCES GrowsCrop(CropID)
        ON DELETE CASCADE
);

-- crop yield produces table (weak entity)
CREATE TABLE CropYieldProduces (
    CropID      INT PRIMARY KEY,
    Total_Yield  DECIMAL(10,2) NOT NULL,
    Health_Rating INT NOT NULL,
    CONSTRAINT fk_yield_crop
        FOREIGN KEY (CropID) REFERENCES GrowsCrop(CropID)
        ON DELETE CASCADE
);

-- pesticide table
CREATE TABLE Pesticide (
    PestID      INT PRIMARY KEY,
    Name        VARCHAR(60)
);

-- treats table
CREATE TABLE Treats (
    CropID      INT,
    PestID      INT,
    PRIMARY KEY (CropID, PestID),
    CONSTRAINT fk_treats_crop
        FOREIGN KEY (CropID) REFERENCES GrowsCrop(CropID)
        ON DELETE CASCADE,
    CONSTRAINT fk_treats_pest
        FOREIGN KEY (PestID) REFERENCES Pesticide(PestID)
);

```

```

    ON DELETE CASCADE
) ;

-- irrigation records table
CREATE TABLE IrrigationRecords (
    IrrigID      INT PRIMARY KEY,
    FieldID      INT NOT NULL,
    EventDate    DATE,
    Volume       DECIMAL(10, 2),
    CONSTRAINT fk_irrig_field
        FOREIGN KEY (FieldID) REFERENCES ContainsField(FieldID)
) ;

-- soil records table
-- 1) MoistureByChemistry(SampleDate, pH → Moisture)
-- 2) SoilRecords(SoilCondID, FieldID, SampleDate, pH)

-- moisture by chemistry table
CREATE TABLE MoistureByChemistry (
    SampleDate DATE,
    pH          DECIMAL(4, 2),
    Moisture    DECIMAL(6, 2) NOT NULL,
    PRIMARY KEY (SampleDate, pH)
) ;

-- soil records table
CREATE TABLE SoilRecords (
    SoilCondID  INT PRIMARY KEY,
    FieldID     INT NOT NULL,
    SampleDate  DATE NOT NULL,
    pH          DECIMAL(4, 2) NOT NULL,
    CONSTRAINT fk_soil_field
        FOREIGN KEY (FieldID) REFERENCES ContainsField(FieldID),
    CONSTRAINT fk_soil_rule
        FOREIGN KEY (SampleDate, pH) REFERENCES
MoistureByChemistry(SampleDate, pH)
) ;

-- certification table
-- 1) AwardExpiry(AwardedDate → ExpiryDate)
-- 2) Certification(CertID, Name, AwardedDate)

-- award expiry table

```

```

CREATE TABLE AwardExpiry (
    AwardedDate DATE PRIMARY KEY,
    ExpiryDate DATE NOT NULL
);

-- certification table
CREATE TABLE Certification (
    CertID      INT PRIMARY KEY,
    Name        VARCHAR(80) NOT NULL,
    AwardedDate DATE NOT NULL,
    CONSTRAINT fk_cert_award
        FOREIGN KEY (AwardedDate) REFERENCES AwardExpiry(AwardedDate)
);

-- receives table
CREATE TABLE Receives (
    FarmID     INT,
    CertID     INT,
    PRIMARY KEY (FarmID, CertID),
    CONSTRAINT fk_recv_farm
        FOREIGN KEY (FarmID) REFERENCES OwnsFarm(FarmID),
    CONSTRAINT fk_recv_cert
        FOREIGN KEY (CertID) REFERENCES Certification(CertID)
);

```

8.)

```

-- ContactInfoName
INSERT INTO ContactInfoName (ContactInfo, Name) VALUES
('john.doe@example.com', 'John Doe');
INSERT INTO ContactInfoName (ContactInfo, Name) VALUES
('jack.smith@example.com', 'Jack Smith');
INSERT INTO ContactInfoName (ContactInfo, Name) VALUES
('brian.smith@example.com', 'Brian Smith');
INSERT INTO ContactInfoName (ContactInfo, Name) VALUES
('cora.doe@example.com', 'Cora Doe');
INSERT INTO ContactInfoName (ContactInfo, Name) VALUES
('eve.miller@example.com', 'Eve Miller');

-- Farmer
INSERT INTO Farmer (FarmerID, ContactInfo) VALUES (1,
'john.doe@example.com');

```

```
INSERT INTO Farmer (FarmerID, ContactInfo) VALUES (2,
'jack.smith@example.com');

INSERT INTO Farmer (FarmerID, ContactInfo) VALUES (3,
'brian.smith@example.com');

INSERT INTO Farmer (FarmerID, ContactInfo) VALUES (4,
'cora.doe@example.com');

INSERT INTO Farmer (FarmerID, ContactInfo) VALUES (5,
'eve.miller@example.com');

-- OwnsFarm
INSERT INTO OwnsFarm (FarmID, Name, Location, FarmerID) VALUES (101,
'Sunny Fields','British Columbia', 1);
INSERT INTO OwnsFarm (FarmID, Name, Location, FarmerID) VALUES (102,
'Green Valley','Alberta', 2);
INSERT INTO OwnsFarm (FarmID, Name, Location, FarmerID) VALUES (103,
'River Farm','Ontario', 3);
INSERT INTO OwnsFarm (FarmID, Name, Location, FarmerID) VALUES (104,
'Highland Acres','British Columbia', 4);
INSERT INTO OwnsFarm (FarmID, Name, Location, FarmerID) VALUES (105,
'Golden Harvest','Manitoba', 5);

-- ContainsField
INSERT INTO ContainsField (FieldID, FarmID, Area) VALUES (1001, 101,
25);
INSERT INTO ContainsField (FieldID, FarmID, Area) VALUES (1002, 102,
40);
INSERT INTO ContainsField (FieldID, FarmID, Area) VALUES (1003, 103,
30);
INSERT INTO ContainsField (FieldID, FarmID, Area) VALUES (1004, 104,
35);
INSERT INTO ContainsField (FieldID, FarmID, Area) VALUES (1005, 105,
25);

-- SeasonByPlantDate
INSERT INTO SeasonByPlantDate (PlantingDate, Season) VALUES (DATE
'2025-03-10', 'Spring');
INSERT INTO SeasonByPlantDate (PlantingDate, Season) VALUES (DATE
'2025-04-15', 'Spring');
INSERT INTO SeasonByPlantDate (PlantingDate, Season) VALUES (DATE
'2025-05-20', 'Summer');
INSERT INTO SeasonByPlantDate (PlantingDate, Season) VALUES (DATE
'2025-07-01', 'Summer');
```

```
INSERT INTO SeasonByPlantDate (PlantingDate, Season) VALUES (DATE
'2025-09-10', 'Fall');

-- CropType
INSERT INTO CropType (Name, PlantingDate, HarvestDate) VALUES ('Wheat',
DATE '2025-03-10', DATE '2025-07-20');
INSERT INTO CropType (Name, PlantingDate, HarvestDate) VALUES ('Corn',
DATE '2025-04-15', DATE '2025-09-05');
INSERT INTO CropType (Name, PlantingDate, HarvestDate) VALUES
('Tomato', DATE '2025-05-20', DATE '2025-08-25');
INSERT INTO CropType (Name, PlantingDate, HarvestDate) VALUES
('Lettuce', DATE '2025-07-01', DATE '2025-08-01');
INSERT INTO CropType (Name, PlantingDate, HarvestDate) VALUES ('Apple',
DATE '2025-09-10', DATE '2026-03-01');

-- GrowsCrop
INSERT INTO GrowsCrop (CropID, FieldID, Name) VALUES (201, 1001,
'Wheat');
INSERT INTO GrowsCrop (CropID, FieldID, Name) VALUES (202, 1002,
'Corn');
INSERT INTO GrowsCrop (CropID, FieldID, Name) VALUES (203, 1003,
'Tomato');
INSERT INTO GrowsCrop (CropID, FieldID, Name) VALUES (204, 1004,
'Lettuce');
INSERT INTO GrowsCrop (CropID, FieldID, Name) VALUES (205, 1005,
'Apple');

-- IsA tables
-- Grain crops
INSERT INTO Grain (CropID, GlutenContent) VALUES (201, 12.5);
INSERT INTO Grain (CropID, GlutenContent) VALUES (202, 0.00);

-- Vegetable crops
INSERT INTO Vegetable (CropID, IsLeafy) VALUES (203, 0);
INSERT INTO Vegetable (CropID, IsLeafy) VALUES (204, 1);

-- Fruit crops
INSERT INTO Fruit (CropID, SugarContent) VALUES (205, 14.2);

-- CropYieldProduces
INSERT INTO CropYieldProduces (CropID, Total_Yield, Health_Rating)
VALUES (201, 5000.00, 9);
```

```

INSERT INTO CropYieldProduces (CropID, Total_Yield, Health_Rating)
VALUES (202, 6500.50, 8);
INSERT INTO CropYieldProduces (CropID, Total_Yield, Health_Rating)
VALUES (203, 2200.75, 7);
INSERT INTO CropYieldProduces (CropID, Total_Yield, Health_Rating)
VALUES (204, 1800.00, 9);
INSERT INTO CropYieldProduces (CropID, Total_Yield, Health_Rating)
VALUES (205, 8000.25, 10);

INSERT INTO Pesticide(PestID, Name) VALUES (1, 'Mr Clean');
INSERT INTO Pesticide(PestID, Name) VALUES (2, 'Pest Killer');
INSERT INTO Pesticide(PestID, Name) VALUES (3, 'P-Cleaner Deluxe');
INSERT INTO Pesticide(PestID, Name) VALUES (4, 'Bye Bye Pests');
INSERT INTO Pesticide(PestID, Name) VALUES (5, 'Get Outta Here Pests');
INSERT INTO Pesticide(PestID, Name) VALUES (6, 'Pest Eliminator 3000');

INSERT INTO Treats(CropID, PestID) VALUES (201, 1);
INSERT INTO Treats(CropID, PestID) VALUES (201, 2);
INSERT INTO Treats(CropID, PestID) VALUES (202, 1);
INSERT INTO Treats(CropID, PestID) VALUES (202, 3);
INSERT INTO Treats(CropID, PestID) VALUES (203, 2);
INSERT INTO Treats(CropID, PestID) VALUES (205, 3);

-- IrrigationRecords
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (1, 1001, DATE '2023-03-01', 1500.00);
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (2, 1001, DATE '2023-04-15', 2000.00);
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (3, 1002, DATE '2023-05-10', 1800.00);
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (4, 1002, DATE '2023-06-20', 2200.00);
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (5, 1003, DATE '2023-07-05', 1600.00);
INSERT INTO IrrigationRecords (IrrigID, FieldID, EventDate, Volume)
VALUES (6, 1003, DATE '2023-08-18', 2100.00);

-- MoistureByChemistry
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-03-15', 6.5, 22.5);

```

```
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-04-20', 7.0, 18.0);
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-05-10', 5.8, 25.0);
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-06-25', 6.2, 20.0);
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-07-30', 7.5, 15.0);
INSERT INTO MoistureByChemistry (SampleDate, pH, Moisture) VALUES (DATE
'2023-08-15', 6.9, 19.5);

-- SoilRecords
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(1, 1001, DATE '2023-03-15', 6.5);
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(2, 1001, DATE '2023-04-20', 7.0);
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(3, 1002, DATE '2023-05-10', 5.8);
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(4, 1002, DATE '2023-06-25', 6.2);
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(5, 1003, DATE '2023-07-30', 7.5);
INSERT INTO SoilRecords (SoilCondID, FieldID, SampleDate, pH) VALUES
(6, 1003, DATE '2023-08-15', 6.9);

-- AwardExpiry
INSERT INTO AwardExpiry (AwardedDate, ExpiryDate) VALUES (DATE
'2021-10-16', DATE '2023-10-16');
INSERT INTO AwardExpiry (AwardedDate, ExpiryDate) VALUES (DATE
'2022-01-15', DATE '2024-01-15');
INSERT INTO AwardExpiry (AwardedDate, ExpiryDate) VALUES (DATE
'2023-02-15', DATE '2025-02-15');
INSERT INTO AwardExpiry (AwardedDate, ExpiryDate) VALUES (DATE
'2024-05-11', DATE '2026-05-11');
INSERT INTO AwardExpiry (AwardedDate, ExpiryDate) VALUES (DATE
'2024-12-30', DATE '2026-12-30');

-- Certification
INSERT INTO Certification (CertID, Name, AwardedDate) VALUES (1,
'Organic', DATE '2022-01-15');
```

```
INSERT INTO Certification (CertID, Name, AwardedDate) VALUES (2,
'Winner', DATE '2023-02-15');
INSERT INTO Certification (CertID, Name, AwardedDate) VALUES (3,
'BestFarm', DATE '2024-12-30');
INSERT INTO Certification (CertID, Name, AwardedDate) VALUES (4,
'PrettiestFarm', DATE '2024-05-11');
INSERT INTO Certification (CertID, Name, AwardedDate) VALUES (5,
'BestEnvironment', DATE '2021-10-16');

-- Receives
INSERT INTO Receives (FarmID, CertID) VALUES (101, 1);
INSERT INTO Receives (FarmID, CertID) VALUES (101, 2);
INSERT INTO Receives (FarmID, CertID) VALUES (102, 1);
INSERT INTO Receives (FarmID, CertID) VALUES (103, 3);
INSERT INTO Receives (FarmID, CertID) VALUES (104, 2);
INSERT INTO Receives (FarmID, CertID) VALUES (105, 1);
```

9.) No AI was used for this assignment