**University of British Columbia, Vancouver**
Department of Computer Science

# CPSC 304 Project Cover Page

Milestone #: 2

Date: 03/03/2025

Group Number: 4

| Name | Student Number | CS Alias (Userid) | Preferred E-mail Address |
|---|---|---|---|
| Hannah Baek | 35222264 | i8x2k | baekhannah852@gmail.com |
| Matt Skelton | 32367989 | a4t4d | matthew.skelton5@gmail.com |
| Ashley Wu | 81864530 | n4c7l | ashleywu2004@gmail.com |

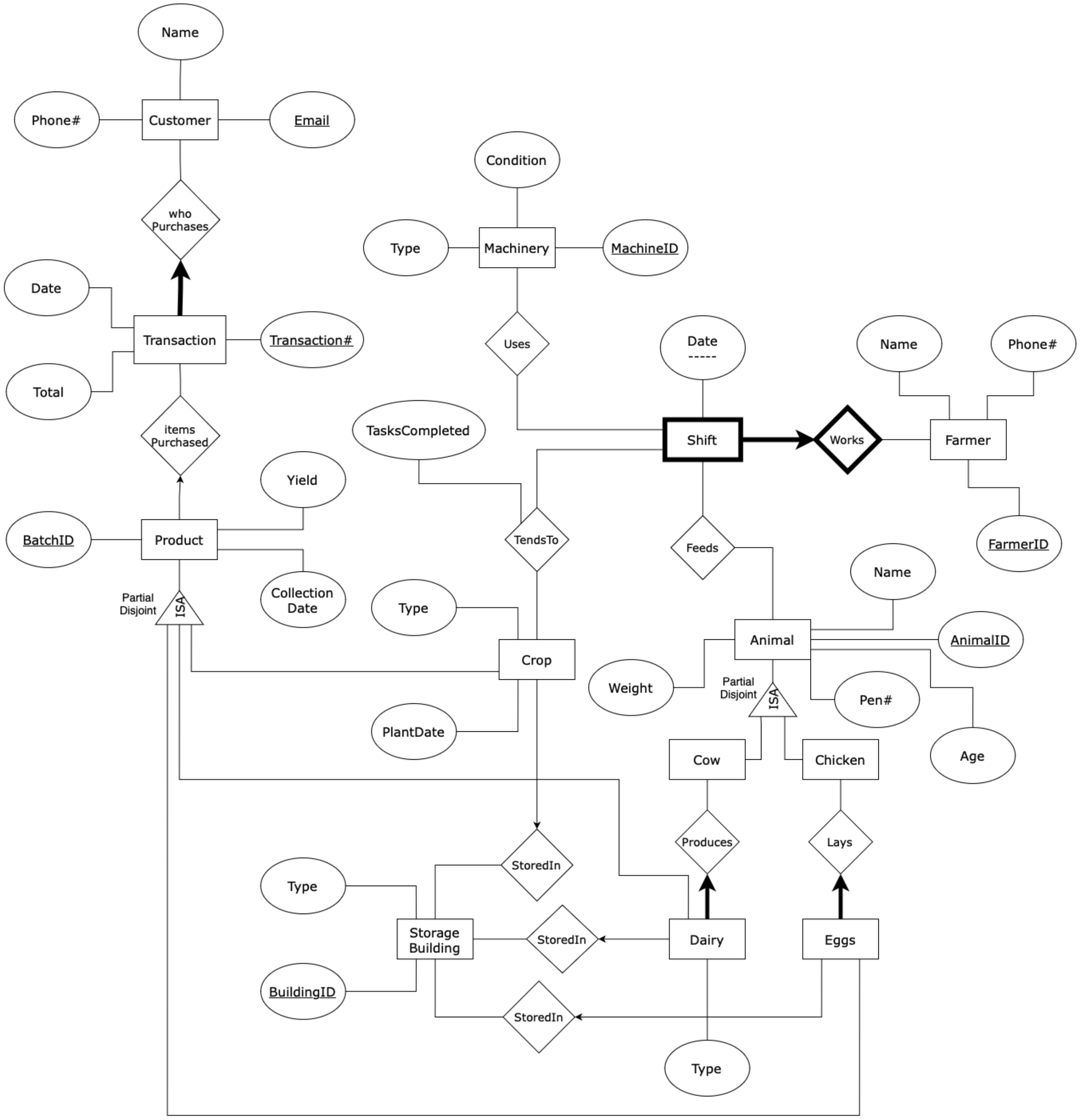
By typing our names and student numbers in the above table, we certify that the work in the attached assignment was performed solely by those whose names and student IDs are included above. (In the case of Project Milestone 0, the main purpose of this page is for you to let us know your e-mail address,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

# Deliverables

1. **Project Summary**
   Our application will be a tool for farm management, including workers, inventory, machinery, animals, and sales. It is intended to allow managers to monitor profits from the farm and ensure that many farm management operations are running smoothly.

2. **ER Diagram**



### Changes made to the ER diagram

- Total participation on Transaction because a transaction cannot happen without a customer.
- Many to one relationship with Transaction and Customer because a transaction can be completed by one customer.
- Changed relationship from ‘tendsTo’ (crops), ‘uses’ (machinery), and ‘feeds’ (animals) to allow the same farmer to perform these actions on the same entity during separate shifts.
- Added a ‘tasksCompleted’ attribute under ‘tendsTo’ because a farmer can keep record of what they did to that crop during their shift. It may just be a single task like “watering” or it may be multiple tasks.

### 3. Schema derived from ER diagram

**Entities** **Relationships** **Entity-Relationships**

- Customer(Email: VARCHAR, Name: VARCHAR, Phone#: CHAR(10))
  - Candidate Keys: Email
- Transaction(Transaction#: INTEGER, **Email:** VARCHAR, Date: DATE, Total: DECIMAL(19,4))
  - Candidate Keys: Transaction#
  - Email not null
- Products(BatchID: INTEGER, Yield: INTEGER, CollectionDate: DATE, **Transaction#:** INTEGER)
  - Candidate Keys: BatchID
- Farmer(FarmerID: INTEGER, Name: VARCHAR, Phone#: CHAR(10))
  - Candidate Keys: FarmerID
- Shift(**FarmerID:** INTEGER, Date: DATE)
  - Candidate Keys: {FarmerID, Date}
- Crop(**BatchID:** INTEGER, Type: VARCHAR, PlantDate: DATE, **BuildingID:** INTEGER)
  - Candidate Keys: BatchID
- CropMaintenance(TasksCompleted: VARCHAR, **FarmerID:** INTEGER, **Date:** DATE, **BatchID:** INTEGER)
  - Candidate Keys: {FarmerID, Date, BatchID}
- Machinery(MachineID: INTEGER, Type: VARCHAR, Condition: VARCHAR)
  - Candidate Keys: MachineID
- MachineryUsage(**FarmerID:** INTEGER, **MachineID:** INTEGER, **Date:** DATE)
  - Candidate Keys: {FarmerID, MachineID, Date}
- Animal(AnimalID: INTEGER, Name: VARCHAR, Age: INTEGER, Pen#: INTEGER, Weight: DECIMAL(8,2))
  - Candidate Keys: AnimalID
- AnimalFeedingLog(**FarmerID:** INTEGER, **AnimalID:** INTEGER**,** **Date:** DATE)
  - Candidate Keys: {FarmerID, AnimalID, Date}
- Cow(**AnimalID:** INTEGER)
  - Candidate Keys: AnimalID

- DairyRecords(**BatchID:** INTEGER, Type: VARCHAR, **AnimalID:** INTEGER, **BuildingID:** INTEGER)
  - Candidate Keys: BatchID
  - AnimalID not null
- Chicken(**AnimalID:** INTEGER)
  - Candidate Keys: AnimalID
- EggRecords(**BatchID:** INTEGER, **AnimalID:** INTEGER, **BuildingID:** INTEGER)
  - Candidate Keys: BatchID
  - AnimalID not null
- StorageBuilding(BuildingID**:** INTEGER, Type: VARCHAR)
  - Candidate Keys: BuildingID

## 4. Functional Dependencies

| Source Entity/Relations | FDs |
|---|---|
| Customer | cEmail → cName, cPhone# |
| Transaction | Transaction# → tDate, Total<br>Transaction# → cEmail, cName, cPhone# |
| Products | BatchID → Yield, Collection Date<br>BatchID → Transaction#, tDate, Total<br>BatchID → cName, cEmail, cPhone# |
| Machinery | MachineID → Condition, mType |
| Farmer | FarmerID → fPhone#, fName |
| Shift | FarmerID, sDate → fName, fPhone#<br>FarmerID, sDate, BatchID → TasksCompleted |
| Animal | AnimalID → aName, Age, Pen#, Weight |
| Storage Building | BuildingID → sbType |
| Dairy | BatchID → dType<br>BatchID → BuildingID, sbType<br>BatchID → AnimalID, aName, Age, Weight, Pen# |
| Eggs | BatchID → BuildingID, sbType<br>BatchID → AnimalID, aName, Age, Weight, Pen# |
| Crop | BatchID → crType, Plant Date<br>BatchID → BuildingID, sbType |

## 5. Normalization to 3NF

We chose to decompose into 3NF because dependency preservation is our main priority as this will help farmers efficiently retrieve data without requiring joining tables.

Customer(cEmail: VARCHAR, cName: VARCHAR, cPhone#: CHAR(10))
Transaction(Transaction#: INTEGER, **cEmail:** VARCHAR, tDate: DATE, Total: DECIMAL(10, 2))
Products(BatchID: INTEGER, Yield: INTEGER, Collection Date: DATE)
PurchasedProducts(**BatchID:** INTEGER, **Transaction#:** INTEGER)
Farmer(FarmerID: INTEGER, fName: VARCHAR, fPhone#: CHAR(10))
Crop(**BatchID:** INTEGER, crType:VARCHAR, PlantDate: DATE, **BuildingID:** INTEGER)
CropMaintenance(TasksCompleted: VARCHAR, **FarmerID:** INTEGER, **sDate:** DATE, **BatchID:** INTEGER)
Machinery(MachineID: INTEGER, mType: VARCHAR, Condition: VARCHAR)
MachineryUsage(**FarmerID**: INTEGER, **MachineID**: INTEGER, **sDate**: DATE)
Animal(AnimalID: INTEGER, aName: VARCHAR, Age: INTEGER, Pen#: INTEGER, Weight: DECIMAL(8, 2))
AnimalFeedingLog(**FarmerID**: INTEGER, **AnimalID**: INTEGER, **sDate**: DATE)
Cow(**AnimalID**: INTEGER)
Chicken(**AnimalID**: INTEGER)
StorageBuilding(BuildingID: INTEGER, sbType: VARCHAR)
EggRecords(**BatchID**: INTEGER, **AnimalID**: INTEGER, **BuildingID**: INTEGER)
DairyRecords(**BatchID**: INTEGER, dType: VARCHAR, **AnimalID**: INTEGER, **BuildingID**: INTEGER)
Shift(**FarmerID**: INTEGER, sDate: DATE)

Description of steps taken: most of the relationships were already in 3NF. The only one that needed to be decomposed was

Products(BatchID: INTEGER, Yield: INTEGER, Collection Date: DATE, **Transaction#**: INTEGER)

because of BatchID → Transaction#, on which it was decomposed into:

Products(BatchID: INTEGER, Yield: INTEGER, Collection Date: DATE)
PurchasedProducts(**BatchID**: INTEGER, **Transaction#**: INTEGER)

Both of these relationships are in 3NF, so we are done.

## 6. SQL DDL Statements

```
CREATE TABLE Customer (
        cEmail VARCHAR,
	cName VARCHAR,
	cPhone# CHAR(10),
	PRIMARY KEY (cEmail)
);

CREATE TABLE Transaction (
	Transaction# INTEGER,
	cEmail VARCHAR NOT NULL,
	tDate DATE,
	Total DECIMAL(10, 2),
	PRIMARY KEY (Transaction#),
	FOREIGN KEY (cEmail) REFERENCES Customer
);

CREATE TABLE Products (
	BatchID INTEGER,
	Yield INTEGER,
	CollectionDate DATE,
	PRIMARY KEY (BatchID)
);

CREATE TABLE PurchasedProducts (
	BatchID INTEGER,
	Transaction# INTEGER,
	PRIMARY KEY (BatchID, Transaction#),
	FOREIGN KEY (BatchID) REFERENCES Products
	FOREIGN KEY (Transaction#) REFERENCES Transaction
);

CREATE TABLE Farmer (
	FarmerID INTEGER,
	fName VARCHAR,
	fPhone# CHAR(10),
	PRIMARY KEY (FarmerID),
);

CREATE TABLE Shift (
	FarmerID INTEGER,
	sDate DATE,
	PRIMARY KEY (FarmerID, sDate),
	FOREIGN KEY (FarmerID) REFERENCES Farmer
);

CREATE TABLE Crop (
        BatchID INTEGER,
        crType VARCHAR,
        PlantDate DATE,
        BuildingID INTEGER,
        PRIMARY KEY (BatchID),
        FOREIGN KEY (BatchID) REFERENCES Products,
        FOREIGN KEY (BuildingID) REFERENCES StorageBuilding
);

CREATE TABLE CropMaintenance (
        TasksCompleted VARCHAR,
        FarmerID INTEGER,
        sDate DATE,
        BatchID INTEGER,
        PRIMARY KEY (FarmerID, sDate, BatchID),
        FOREIGN KEY (FarmerID, sDate) REFERENCES Shift,
        FOREIGN KEY (BatchID) REFERENCES Crop
);

CREATE TABLE Machinery (
        MachineID INTEGER,
        mType VARCHAR,
        Condition VARCHAR,
        PRIMARY KEY (MachineID)
);

CREATE TABLE MachineryUsage (
        FarmerID INTEGER,
        MachineID INTEGER,
        sDate DATE,
        PRIMARY KEY (FarmerID, sDate, MachineID),
        FOREIGN KEY (FarmerID, sDate) REFERENCES Shift,
        FOREIGN KEY (MachineID) REFERENCES Machinery
);

CREATE TABLE Animal (
        AnimalID INTEGER,
        aName VARCHAR,
        Age INTEGER,
        Pen# INTEGER,
        Weight DECIMAL(8, 2),
        PRIMARY KEY (AnimalID),
);
```

```
CREATE TABLE AnimalFeedingLog (
        FarmerID INTEGER,
        AnimalID INTEGER,
        sDate DATE,
        PRIMARY KEY (FarmerID, sDate, AnimalID),
        FOREIGN KEY (FarmerID, sDate) REFERENCES Shift,
        FOREIGN KEY (AnimalID) REFERENCES Animal
);

CREATE TABLE Cow (
        AnimalID INTEGER,
        PRIMARY KEY (AnimalID),
        FOREIGN KEY (AnimalID) REFERENCES Animal
);

CREATE TABLE DairyRecords (
        BatchID INTEGER,
        dType VARCHAR,
        AnimalID INTEGER NOT NULL,
        BuildingID INTEGER,
        PRIMARY KEY (BatchID),
        FOREIGN KEY (BatchID) REFERENCES Products,
        FOREIGN KEY (AnimalID) REFERENCES Cow,
        FOREIGN KEY (BuildingID) REFERENCES StorageBuilding
);

CREATE TABLE Chicken (
        AnimalID INTEGER,
        PRIMARY KEY (AnimalID),
        FOREIGN KEY (AnimalID) REFERENCES Animal
);

CREATE TABLE EggRecords (
        BatchID INTEGER,
        AnimalID INTEGER NOT NULL,
        BuildingID INTEGER,
        PRIMARY KEY (BatchID),
        FOREIGN KEY (BatchID) REFERENCES Products,
        FOREIGN KEY (AnimalID) REFERENCES Chicken,
        FOREIGN Key (BuildingID) REFERENCES StorageBuilding
);

CREATE TABLE StorageBuilding (
        BuildingID INTEGER,
        sbType VARCHAR,
        PRIMARY KEY (BuildingID),
);
```

## 7. Insert Statements

- Customer(cEmail: VARCHAR, cName: VARCHAR, cPhone#: CHAR(10))

```
INSERT INTO Customer
VALUES (“janesmith93@gmail.com”, “Jane Smith”, “604-122-3”);
INSERT INTO Customer
VALUES (“masonyurb03@gmail.com”, “Mason Yurb”, “778-322-3992”);
INSERT INTO Customer
VALUES (“romantaurk87@gmail.com”, “Roman Taurk”, “778-981-6432”);
INSERT INTO Customer
VALUES (“rbauli22@gmail.com”, “Raiya Bauli”, “604-261-0184”);
INSERT INTO Customer
VALUES (“parmkarla84@gmail.com”, “Karla Parm”, “604-918-3651”);
```

- Transaction(Transaction#: INTEGER, **cEmail:** VARCHAR, tDate: DATE, Total: DECIMAL(10, 2))

```
INSERT INTO Transaction
VALUES (1, "janesmith93@gmail.com", "2025-01-15", 1366.00);
INSERT INTO Transaction
VALUES (2, "masonyurb03@gmail.com", "2025-02-03", 289.99);
INSERT INTO Transaction
VALUES (3, "romantaurk87@gmail.com", "2025-02-10", 534.50);
INSERT INTO Transaction
VALUES (4, "rbauli22@gmail.com", "2025-02-20", 875.75);
INSERT INTO Transaction
VALUES (5, "parmkarla84@gmail.com", "2025-02-28", 425.25);
```

- Products(BatchID: INTEGER, Yield: INTEGER, Collection Date: DATE)

```
INSERT INTO Products VALUES (101, 150, '2025-01-15');
INSERT INTO Products VALUES (102, 60, '2025-02-03');
INSERT INTO Products VALUES (103, 220, '2025-02-10');
INSERT INTO Products VALUES (104, 77, '2025-02-20');
INSERT INTO Products VALUES (105, 345, '2025-02-28');
INSERT INTO Products VALUES (106, 80, '2025-01-17');
INSERT INTO Products VALUES (107, 250, '2025-02-05');
INSERT INTO Products VALUES (108, 90, '2025-03-10');
INSERT INTO Products VALUES (109, 170, '2025-02-22');
INSERT INTO Products VALUES (110, 125, '2025-08-28');
INSERT INTO Products VALUES (111, 64, '2025-01-15');
```

INSERT INTO Products VALUES (112, 205, '2025-02-02');
INSERT INTO Products VALUES (113, 95, '2025-09-15');
INSERT INTO Products VALUES (114, 144, '2025-01-22');
INSERT INTO Products VALUES (115, 78, '2025-05-25');

- PurchasedProducts(**BatchID**: INTEGER, **Transaction#:** INTEGER)
  INSERT INTO PurchasedProducts VALUES (111, 1);
  INSERT INTO PurchasedProducts VALUES (112, 2);
  INSERT INTO PurchasedProducts VALUES (113, 3);
  INSERT INTO PurchasedProducts VALUES (114, 4);
  INSERT INTO PurchasedProducts VALUES (115, 5);

- Farmer(FarmerID: INTEGER, fName: VARCHAR, fPhone#: CHAR(10))
  INSERT INTO Farmer VALUES (100, “Jerry Jam”, “778-333-9898”);
  INSERT INTO Farmer VALUES (101, “Carrey Nicol”, “778-541-6312”);
  INSERT INTO Farmer VALUES (102, “Richelle Peters”, “778-232-1431”);
  INSERT INTO Farmer VALUES (103, “Tony Manning”, “778-619-6543”);
  INSERT INTO Farmer VALUES (104, “Peeta Parker”, “778-691-4234”);

- Shift(**FarmerID**: INTEGER, sDate: DATE)
  INSERT INTO Shift VALUES (100, '2024-05-12');
  INSERT INTO Shift VALUES (101, '2025-01-22');
  INSERT INTO Shift VALUES (102, '2025-01-23');
  INSERT INTO Shift VALUES (100, '2025-01-24');
  INSERT INTO Shift VALUES (103, '2025-01-24');

- Crop(**BatchID:** INTEGER, crType: VARCHAR, PlantDate: DATE, **BuildingID:** INTEGER)
  INSERT INTO Crop VALUES (111, "Corn", '2024-04-02', 104);
  INSERT INTO Crop VALUES (112, "Wheat", '2024-12-02', 104);
  INSERT INTO Crop VALUES (113, "Carrot", '2024-11-02', 104);
  INSERT INTO Crop VALUES (114, "Soybean", '2024-10-02', 104);
  INSERT INTO Crop VALUES (115, "Pumpkin", '2025-01-24', 104);

- CropMaintenance(TasksCompleted: VARCHAR, **FarmerID:** INTEGER, **sDate:** DATE, **BatchID:** INTEGER)
  INSERT INTO CropMaintenance VALUES ("Watering and weeding", 100, '2024-05-12', 20);
  INSERT INTO CropMaintenance VALUES ("Harvesting", 101, '2025-01-22', 30);
  INSERT INTO CropMaintenance VALUES ("Spraying pesticides and thinning", 100, '2025-01-24', 40);
  INSERT INTO CropMaintenance VALUES ("Harvesting", 102, '2025-01-23', 50);
  INSERT INTO CropMaintenance VALUES ("Planting", 103, '2025-01-24', 60);

- Machinery(MachineID: INTEGER, mType: VARCHAR, Condition: VARCHAR)
  ```
  INSERT INTO Machinery VALUES (1, “Tractor”, “Fair. Replaced front right wheel”);
  INSERT INTO Machinery VALUES (2, “Planter”,”Good”);
  INSERT INTO Machinery VALUES (3, “Solar Panel”,”New”);
  INSERT INTO Machinery VALUES (4, “Weeding Machine”, “Good, cleaned exterior”);
  INSERT INTO Machinery VALUES (5, “Compost Machine”,”Fair. Replaced and
  maintenance on left compartment”);
  ```

- MachineryUsage(**FarmerID**: INTEGER, **MachineID**: INTEGER, **sDate**: DATE)
  ```
  INSERT INTO MachineryUsage VALUES (100, 1);
  INSERT INTO MachineryUsage VALUES (101, 3);
  INSERT INTO MachineryUsage VALUES (102, 4);
  INSERT INTO MachineryUsage VALUES (103, 5);
  INSERT INTO MachineryUsage VALUES (104, 2);
  ```

- Animal(AnimalID: INTEGER, aName: VARCHAR, Age: INTEGER, Pen#: INTEGER, Weight: DECIMAL(8,2))
  ```
  INSERT INTO Animal VALUES (300, “Pecky”, 8, 121, 2.42);
  INSERT INTO Animal VALUES (301, “Snoopy”, 4, 123, 505.45);
  INSERT INTO Animal VALUES (302, “Woodstock”, 4, 123, 2.75);
  INSERT INTO Animal VALUES (303, “Mooshu”, 10, 124, 484.84);
  INSERT INTO Animal VALUES (304, “Sir Loin”, 15, 125, 575.25);
  INSERT INTO Animal VALUES (305, “Yolks”, 3, 127, 1.66);
  INSERT INTO Animal VALUES (306, “Mooana”, 12, 128, 468.32);
  INSERT INTO Animal VALUES (307, “Feathers”, 8, 125, 1.81);
  INSERT INTO Animal VALUES (308, “Clucks”, 6, 130, 3.03);
  INSERT INTO Animal VALUES (309, “Milkshake”, 11, 133, 625.75);
  ```

- AnimalFeedingLog(**FarmerID**: INTEGER, **AnimalID**: INTEGER, **sDate**: DATE)
  ```
  INSERT INTO AnimalFeedingLog VALUES (101, 300, '2025-02-11');
  INSERT INTO AnimalFeedingLog VALUES (102, 301, '2025-02-12');
  INSERT INTO AnimalFeedingLog VALUES (100, 302, '2025-02-26');
  INSERT INTO AnimalFeedingLog VALUES (104, 303, '2025-02-26');
  INSERT INTO AnimalFeedingLog VALUES (103, 304, '2025-03-01');
  ```

- Cow(**AnimalID**: INTEGER)
  ```
  INSERT INTO Cow VALUES (301);
  INSERT INTO Cow VALUES (303);
  INSERT INTO Cow VALUES (304);
  INSERT INTO Cow VALUES (306);
  INSERT INTO Cow VALUES (309);
  ```

- DairyRecords(**BatchID**: INTEGER, dType: VARCHAR, **AnimalID**: INTEGER, **BuildingID**: INTEGER)

```
INSERT INTO DairyRecords VALUES (101, “Milk”, 303, 101);
INSERT INTO DairyRecords VALUES (103, “Milk”, 304, 101);
INSERT INTO DairyRecords VALUES (105, “Butter”, 304, 102);
INSERT INTO DairyRecords VALUES (107, “Butter”, 306, 102);
INSERT INTO DairyRecords VALUES (109, “Milk”, 309, 101);
```

- Chicken(**AnimalID**: INTEGER)

  ```
  INSERT INTO Chicken VALUES (300);
  INSERT INTO Chicken VALUES (302);
  INSERT INTO Chicken VALUES (305);
  INSERT INTO Chicken VALUES (307);
  INSERT INTO Chicken VALUES (308);
  ```

- EggRecords(**BatchID**: INTEGER, **AnimalID**: INTEGER, **BuildingID**: INTEGER)

  ```
  INSERT INTO EggRecords VALUES (102, 300, 100);
  INSERT INTO EggRecords VALUES (104, 307, 100);
  INSERT INTO EggRecords VALUES (106, 307, 100);
  INSERT INTO EggRecords VALUES (108, 308, 100);
  INSERT INTO EggRecords VALUES (110, 308, 100);
  ```

- StorageBuilding(BuildingID: INTEGER, sbType: VARCHAR)

  ```
  INSERT INTO StorageBuilding VALUES (100, “Egg Cellar”);
  INSERT INTO StorageBuilding VALUES (101, “Milk House”);
  INSERT INTO StorageBuilding VALUES (102, “Butter Place”);
  INSERT INTO StorageBuilding VALUES (103, “Feed Shed”);
  INSERT INTO StorageBuilding VALUES (104, “Crop Silo”);
  ```

## 8. Acknowledgement of AI tools

NONE