JOYCE WOZNICA DATABASE PROJECT

Horse Records

Joyce Woznica
jlwoznic@syr.edu

Contents

List of Figures

Project – Part 1

Part 1 – Project Overview and Entity Diagrams

Updated Summary

For my project, I elected to create a database that will help me keep track of my horses' vaccination requirements, dental visits, chiropractor, and other medical records. The information in this database will allow me to answer the following types of inquiries:

- When is <HORSE> due for a <VACCINE>?
- What vet did <HORSE>'s last <VACCINE>?
- When did <HORSE> get the last dose of <VACCINE>?
- How much money did I spent on vaccine(s) in <YEAR>?
- What is the average price for <VACCINE>?
- How much was spent on <HORSE> for vaccines in <YEAR>?

This data is very important and could actually prove to be useful in the industry in general (for other horse owners).

Background

Certain vaccines are due every 6 months, some once per year, dental visits are usually semi-annual where the chiropractor visits are quarterly. Each horse might get a different batch of a specific vaccine and this can be important if there is a reaction of some kind to the vaccine.

It is very hard to keep track with multiple horses getting vaccines in different locations from different vets. It is important for all horse owners to keep track of when a horse is vaccinated and when he/she is due for his/her next vaccine. It is very important not to miss the proper vaccines as it puts not only that horse in jeopardy, but any horses they come in contact with at their own location and any shows that they might attend.

Entities

- Horse
  - Horse Microchip
  - Horse Name
  - Show Name
  - Foal Date
  - Breed
- Barn
  - Barn Name
  - Barn Address
  - Barn City
  - Barn State
  - Barn Zip
  - Barn Phone
- Vaccine
  - Name
  - Cycle

- Veterinarian
  - Last Name
  - First Name
  - Cell Phone
  - License Number
  - Clinic Associated
- Vet Clinic
  - Clinic Name
  - Clinic Address
  - City
  - State
  - Zip
  - WebSite
- Horse/Vaccine
  - Horse
  - Vaccine
  - Batch
  - Date Given
  - Vet Used
  - Clinic Used

Relationships

- One horse can have 0 to many vaccines.
- One veterinarian can provide 0 to many vaccines to 0 to many horses.
- One clinic can have one or more veterinarians that gives vaccines.
- A veterinarian has to provide all vaccines.
- One clinic can have one to many veterinarians.
- One horse can be located at one and only one barn.
- One barn can have 0 to many horses.

One of the most important relationships is where everything comes together mapping the vet, the clinic, the horse, the vaccine and the dates and batches.

Miscellaneous Information

- The date the vaccine is given and the “batch” is very important as all the scheduling and future dates stem from this information.
- Each vaccine should have an “cycle” with it to note how often it should be given. Possibly cycles used for this project are “Annual,” “Semi-Annual,” “Quarterly”.
- It is important to know the due date of the next vaccine.
- Costs can also be captured to determine if there are vets that might charge a lesser price.

Example Real Data

I have moved all my data to the spreadsheet included with this submission. I used the techniques with sorting data in Excel to help normalize my data.

Entity Diagram

Note: Changed to reflect changes to database column names and primary keys

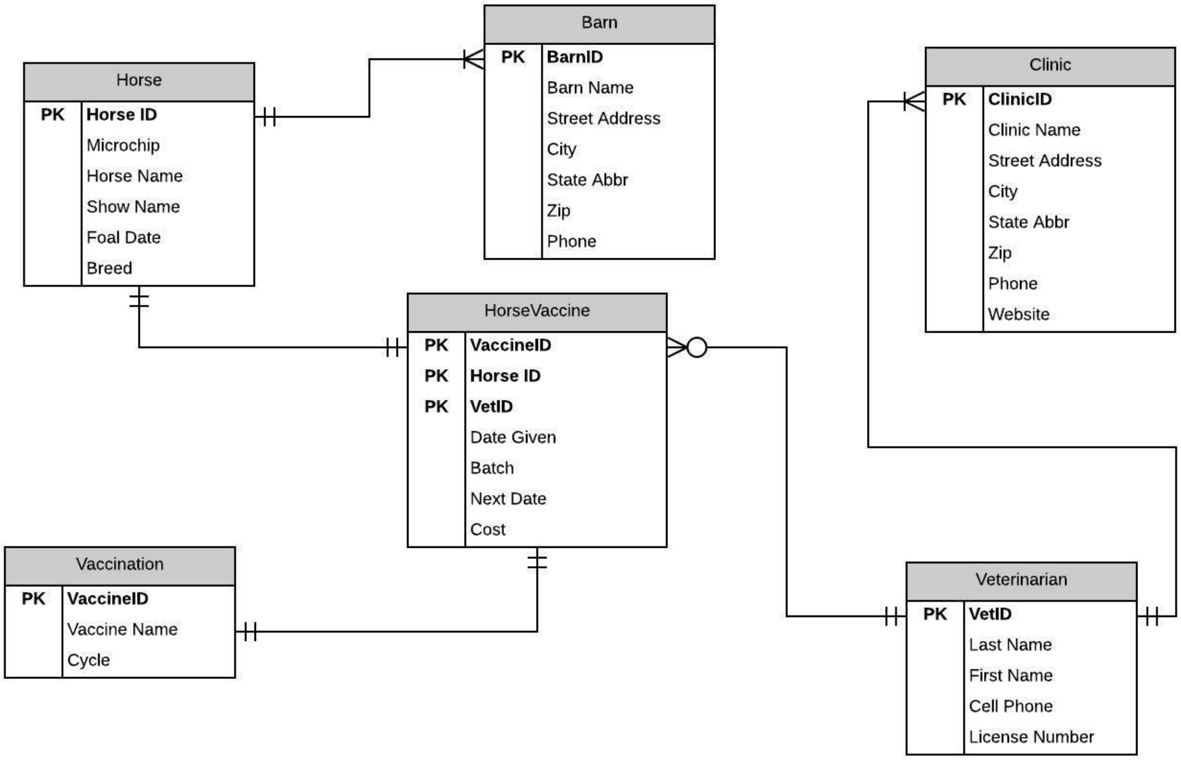


Figure 1: Horse Record Database Entity Diagram

Normalized Model

Note: Changed to reflect changes to database column names and primary keys

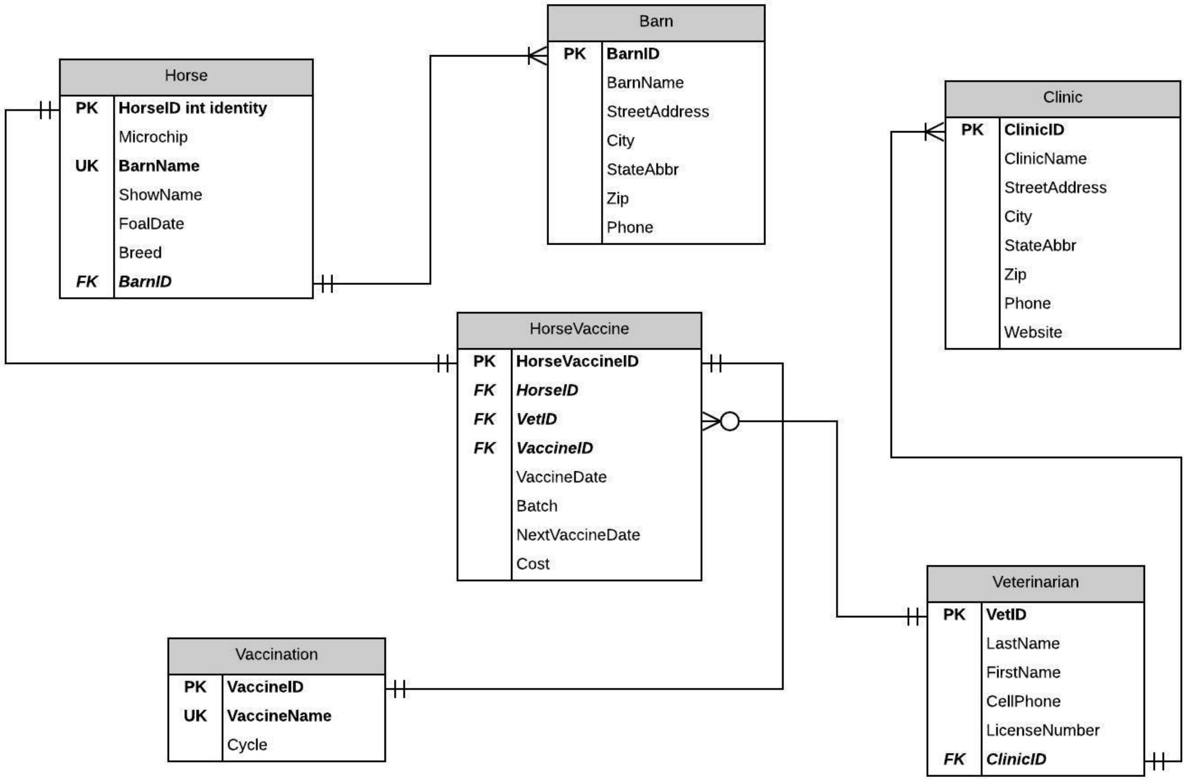


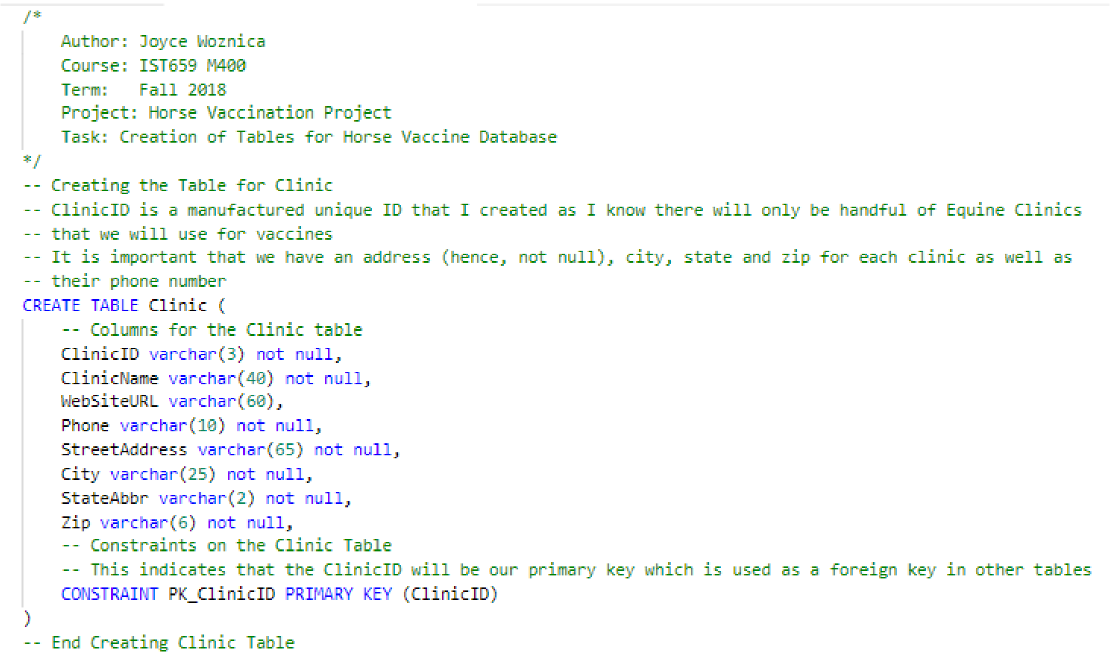
Figure 2: Horse Record Database Normalized Entity Diagram

Project – Part 2

Physical Database Design

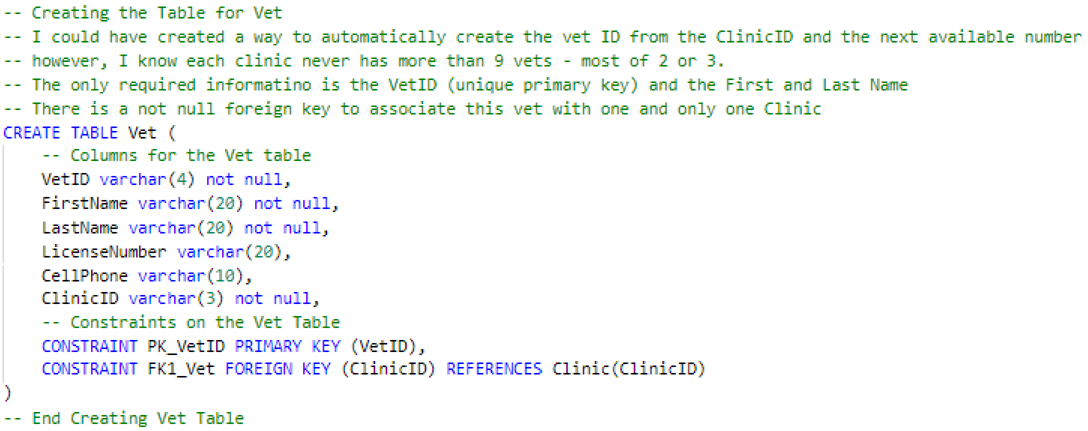
Create Tables

The physical database design is shown in the Create Table SQL code that appears in the following figures. However, please note that one table was changed (***HorseVaccine***) to provide some clarity around the column *VaccineScheduledDate.*



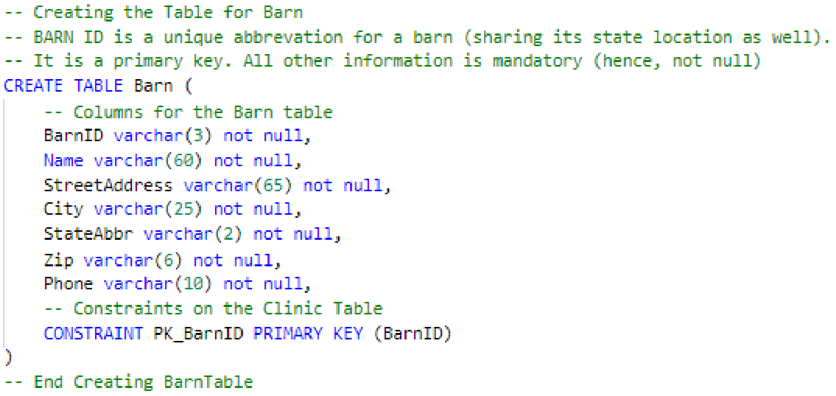
```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Creation of Tables for Horse Vaccine Database
*/
-- Creating the Table for Clinic
-- ClinicID is a manufactured unique ID that I created as I know there will only be handful of Equine Clinics
-- that we will use for vaccines
-- It is important that we have an address (hence, not null), city, state and zip for each clinic as well as
-- their phone number
CREATE TABLE Clinic (
    -- Columns for the Clinic table
    ClinicID varchar(3) not null,
    ClinicName varchar(40) not null,
    WebSiteURL varchar(60),
    Phone varchar(10) not null,
    StreetAddress varchar(65) not null,
    City varchar(25) not null,
    StateAbbr varchar(2) not null,
    Zip varchar(6) not null,
    -- Constraints on the Clinic Table
    -- This indicates that the ClinicID will be our primary key which is used as a foreign key in other tables
    CONSTRAINT PK_ClinicID PRIMARY KEY (ClinicID)
)
-- End Creating Clinic Table
```

Figure 3: Create Clinic Table SQL Commands



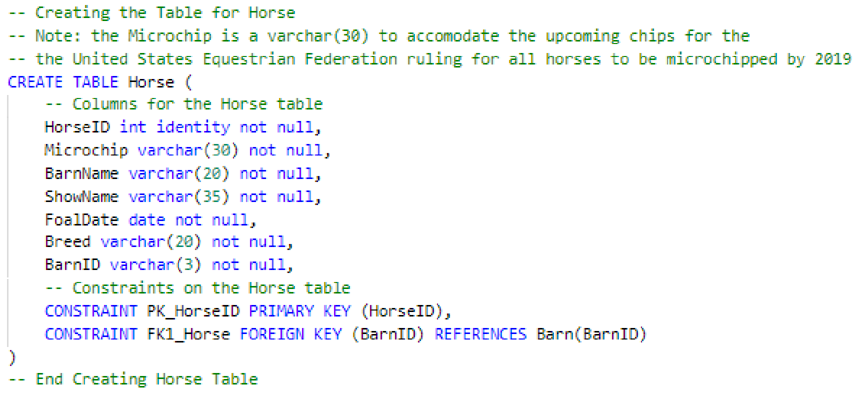
```
-- Creating the Table for Vet
-- I could have created a way to automatically create the vet ID from the ClinicID and the next available number
-- however, I know each clinic never has more than 9 vets - most of 2 or 3.
-- The only required informatino is the VetID (unique primary key) and the First and Last Name
-- There is a not null foreign key to associate this vet with one and only one Clinic
CREATE TABLE Vet (
    -- Columns for the Vet table
    VetID varchar(4) not null,
    FirstName varchar(20) not null,
    LastName varchar(20) not null,
    LicenseNumber varchar(20),
    CellPhone varchar(10),
    ClinicID varchar(3) not null,
    -- Constraints on the Vet Table
    CONSTRAINT PK_VetID PRIMARY KEY (VetID),
    CONSTRAINT FK1_Vet FOREIGN KEY (ClinicID) REFERENCES Clinic(ClinicID)
)
-- End Creating Vet Table
```

Figure 4: Create Vet Table SQL Commands



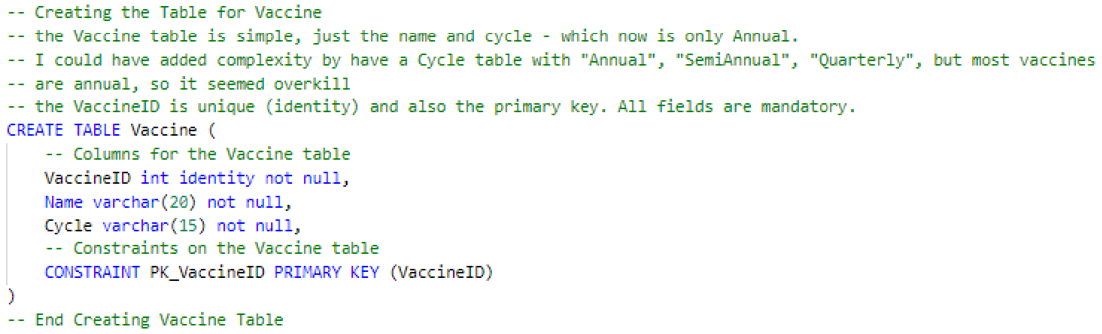
```
-- Creating the Table for Barn
-- BARN ID is a unique abbrevation for a barn (sharing its state location as well).
-- It is a primary key. All other information is mandatory (hence, not null)
CREATE TABLE Barn (
    -- Columns for the Barn table
    BarnID varchar(3) not null,
    Name varchar(60) not null,
    StreetAddress varchar(65) not null,
    City varchar(25) not null,
    StateAbbr varchar(2) not null,
    Zip varchar(6) not null,
    Phone varchar(10) not null,
    -- Constraints on the Clinic Table
    CONSTRAINT PK_BarnID PRIMARY KEY (BarnID)
)
-- End Creating BarnTable
```

Figure 5: Create Barn Table SQL Commands



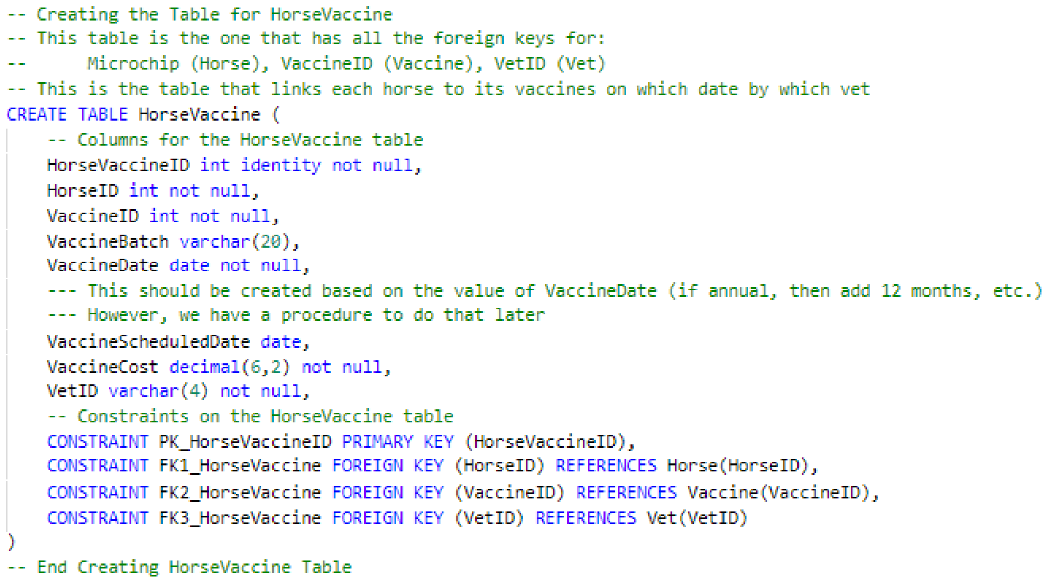
```
-- Creating the Table for Horse
-- Note: the Microchip is a varchar(30) to accomodate the upcoming chips for the
-- the United States Equestrian Federation ruling for all horses to be microchipped by 2019
CREATE TABLE Horse (
    -- Columns for the Horse table
    HorseID int identity not null,
    Microchip varchar(30) not null,
    BarnName varchar(20) not null,
    ShowName varchar(35) not null,
    FoalDate date not null,
    Breed varchar(20) not null,
    BarnID varchar(3) not null,
    -- Constraints on the Horse table
    CONSTRAINT PK_HorseID PRIMARY KEY (HorseID),
    CONSTRAINT FK1_Horse FOREIGN KEY (BarnID) REFERENCES Barn(BarnID)
)
-- End Creating Horse Table
```

Figure 6: Create Horse Table SQL Commands



```
-- Creating the Table for Vaccine
-- the Vaccine table is simple, just the name and cycle - which now is only Annual.
-- I could have added complexity by have a Cycle table with "Annual", "SemiAnnual", "Quarterly", but most vaccines
-- are annual, so it seemed overkill
-- the VaccineID is unique (identity) and also the primary key. All fields are mandatory.
CREATE TABLE Vaccine (
    -- Columns for the Vaccine table
    VaccineID int identity not null,
    Name varchar(20) not null,
    Cycle varchar(15) not null,
    -- Constraints on the Vaccine table
    CONSTRAINT PK_VaccineID PRIMARY KEY (VaccineID)
)
-- End Creating Vaccine Table
```

Figure 7: Create Vaccine Table SQL Commands

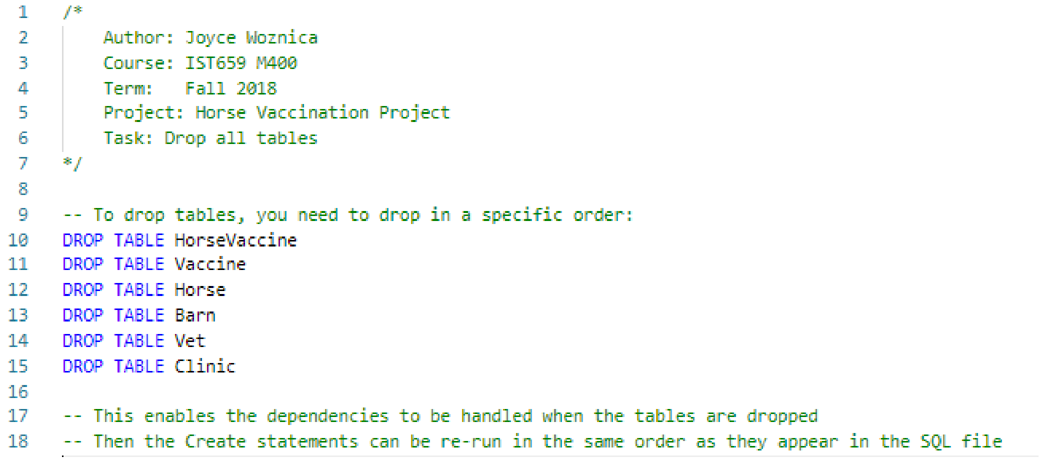


```
-- Creating the Table for HorseVaccine
-- This table is the one that has all the foreign keys for:
--      Microchip (Horse), VaccineID (Vaccine), VetID (Vet)
-- This is the table that links each horse to its vaccines on which date by which vet
CREATE TABLE HorseVaccine (
    -- Columns for the HorseVaccine table
    HorseVaccineID int identity not null,
    HorseID int not null,
    VaccineID int not null,
    VaccineBatch varchar(20),
    VaccineDate date not null,
    --- This should be created based on the value of VaccineDate (if annual, then add 12 months, etc.)
    --- However, we have a procedure to do that later
    VaccineScheduledDate date,
    VaccineCost decimal(6,2) not null,
    VetID varchar(4) not null,
    -- Constraints on the HorseVaccine table
    CONSTRAINT PK_HorseVaccineID PRIMARY KEY (HorseVaccineID),
    CONSTRAINT FK1_HorseVaccine FOREIGN KEY (HorseID) REFERENCES Horse(HorseID),
    CONSTRAINT FK2_HorseVaccine FOREIGN KEY (VaccineID) REFERENCES Vaccine(VaccineID),
    CONSTRAINT FK3_HorseVaccine FOREIGN KEY (VetID) REFERENCES Vet(VetID)
)
-- End Creating HorseVaccine Table
```

Figure 8: Create HorseVaccine Table SQL Commands

Drop Tables

For removing the data, I created the following SQL file, *DropTables4Project.sql* which is shown below. This file represents the order that tables must be dropped due to dependencies. I tested these drop statements multiple times by creating the tables and then dropping them and then recreating.



```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Drop all tables
*/

-- To drop tables, you need to drop in a specific order:
DROP TABLE HorseVaccine
DROP TABLE Vaccine
DROP TABLE Horse
DROP TABLE Barn
DROP TABLE Vet
DROP TABLE Clinic

-- This enables the dependencies to be handled when the tables are dropped
-- Then the Create statements can be re-run in the same order as they appear in the SQL file
```

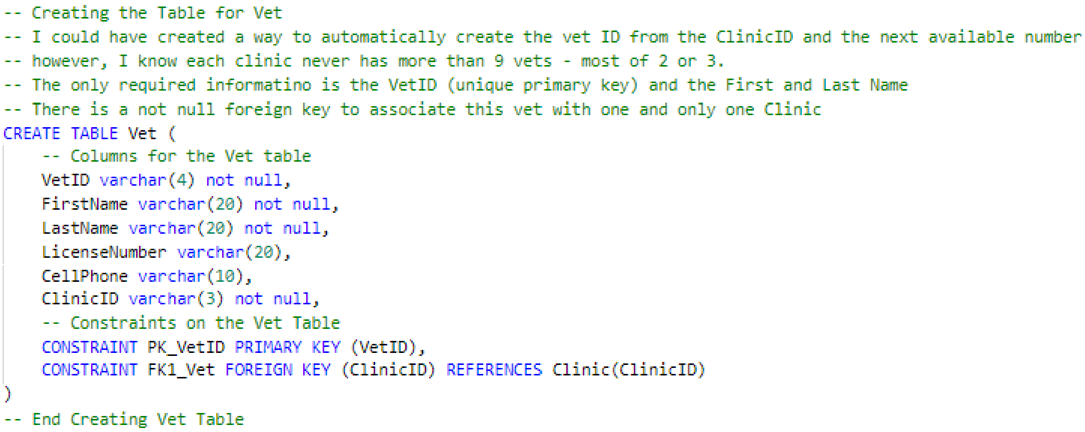
Figure 9: Drop Tables in Required Order SQL Commands

Data Creation

I created insert files for each individual table so it was easier for me to open and close files when needed. I did these in a specific order to verify my dependencies were correct and the data was correct.

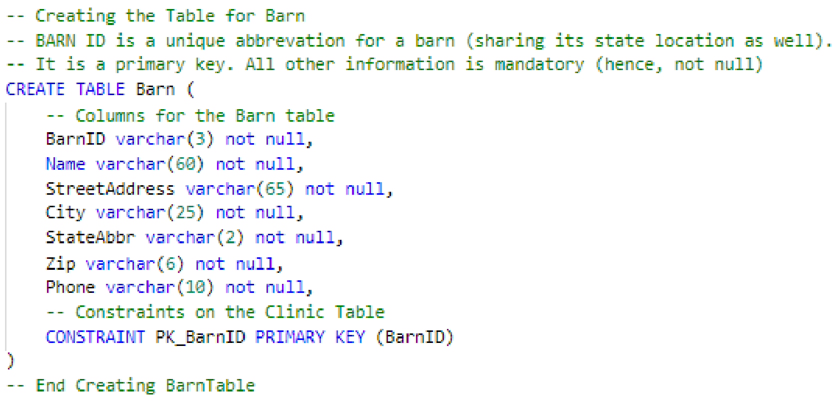
- *01-InsertintoVaccine.sql*
- *02-InsertintoBarn.sql*
- *03-InsertintoHorse.sql*
- *04-InsertintoClinic.sql*
- *05-InsertintoVet.sql*
- *06-InsertintoHorseVaccine.sql*

In generating these INSERT statements, I made some mistakes and had to drop tables and recreate them (see above in the section entitled "



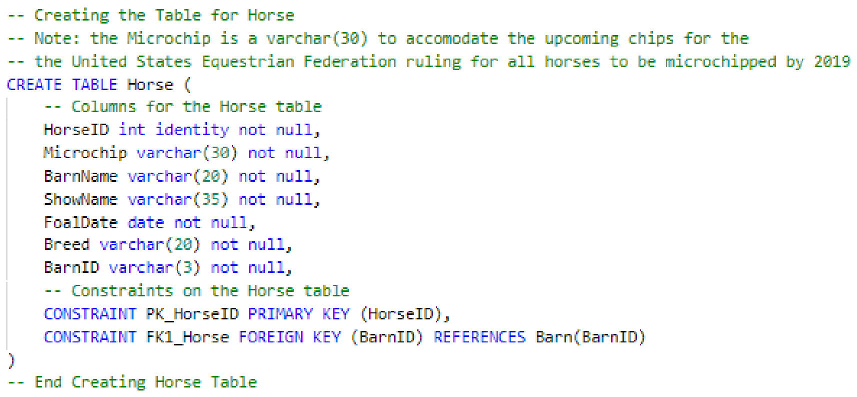
```
-- Creating the Table for Vet
-- I could have created a way to automatically create the vet ID from the ClinicID and the next available number
-- however, I know each clinic never has more than 9 vets - most of 2 or 3.
-- The only required informatino is the VetID (unique primary key) and the First and Last Name
-- There is a not null foreign key to associate this vet with one and only one Clinic
CREATE TABLE Vet (
    -- Columns for the Vet table
    VetID varchar(4) not null,
    FirstName varchar(20) not null,
    LastName varchar(20) not null,
    LicenseNumber varchar(20),
    CellPhone varchar(10),
    ClinicID varchar(3) not null,
    -- Constraints on the Vet Table
    CONSTRAINT PK_VetID PRIMARY KEY (VetID),
    CONSTRAINT FK1_Vet FOREIGN KEY (ClinicID) REFERENCES Clinic(ClinicID)
)
-- End Creating Vet Table
```

Figure 4: Create Vet Table SQL Commands



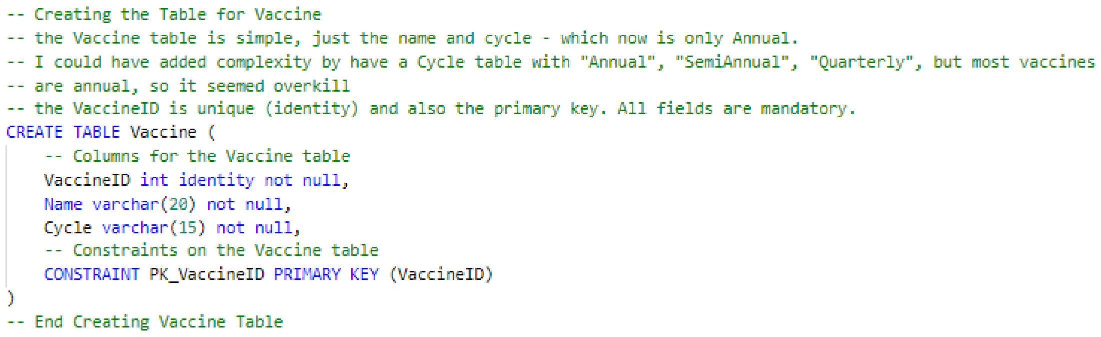
```
-- Creating the Table for Barn
-- BARN ID is a unique abbrevation for a barn (sharing its state location as well).
-- It is a primary key. All other information is mandatory (hence, not null)
CREATE TABLE Barn (
    -- Columns for the Barn table
    BarnID varchar(3) not null,
    Name varchar(60) not null,
    StreetAddress varchar(65) not null,
    City varchar(25) not null,
    StateAbbr varchar(2) not null,
    Zip varchar(6) not null,
    Phone varchar(10) not null,
    -- Constraints on the Clinic Table
    CONSTRAINT PK_BarnID PRIMARY KEY (BarnID)
)
-- End Creating BarnTable
```

Figure 5: Create Barn Table SQL Commands



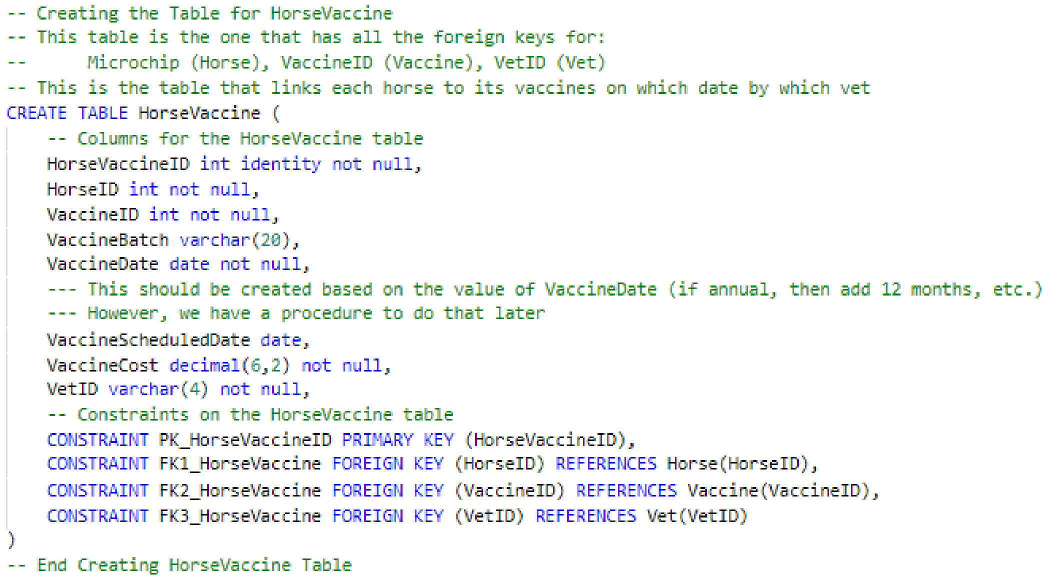
```
-- Creating the Table for Horse
-- Note: the Microchip is a varchar(30) to accomodate the upcoming chips for the
-- the United States Equestrian Federation ruling for all horses to be microchipped by 2019
CREATE TABLE Horse (
    -- Columns for the Horse table
    HorseID int identity not null,
    Microchip varchar(30) not null,
    BarnName varchar(20) not null,
    ShowName varchar(35) not null,
    FoalDate date not null,
    Breed varchar(20) not null,
    BarnID varchar(3) not null,
    -- Constraints on the Horse table
    CONSTRAINT PK_HorseID PRIMARY KEY (HorseID),
    CONSTRAINT FK1_Horse FOREIGN KEY (BarnID) REFERENCES Barn(BarnID)
)
-- End Creating Horse Table
```

Figure 6: Create Horse Table SQL Commands



```
-- Creating the Table for Vaccine
-- the Vaccine table is simple, just the name and cycle - which now is only Annual.
-- I could have added complexity by have a Cycle table with "Annual", "SemiAnnual", "Quarterly", but most vaccines
-- are annual, so it seemed overkill
-- the VaccineID is unique (identity) and also the primary key. All fields are mandatory.
CREATE TABLE Vaccine (
    -- Columns for the Vaccine table
    VaccineID int identity not null,
    Name varchar(20) not null,
    Cycle varchar(15) not null,
    -- Constraints on the Vaccine table
    CONSTRAINT PK_VaccineID PRIMARY KEY (VaccineID)
)
-- End Creating Vaccine Table
```

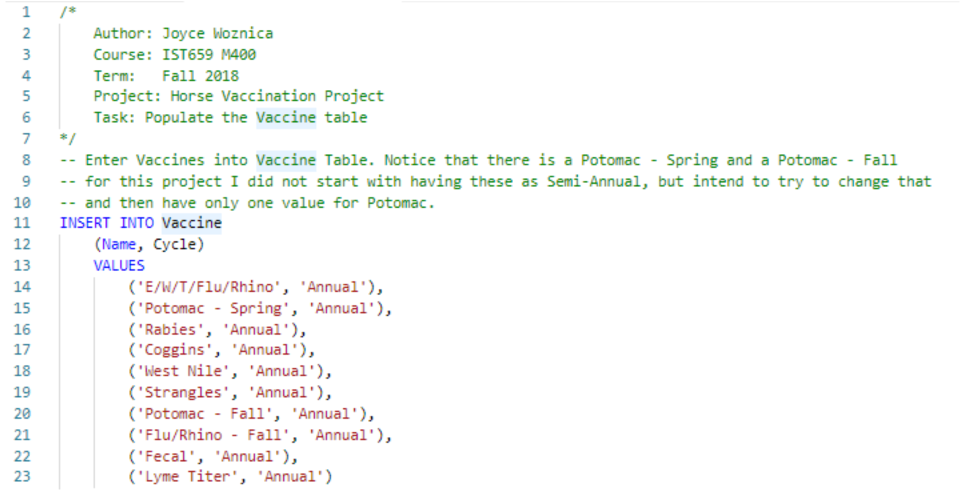
Figure 7: Create Vaccine Table SQL Commands



```sql
-- Creating the Table for HorseVaccine
-- This table is the one that has all the foreign keys for:
--      Microchip (Horse), VaccineID (Vaccine), VetID (Vet)
-- This is the table that links each horse to its vaccines on which date by which vet
CREATE TABLE HorseVaccine (
    -- Columns for the HorseVaccine table
    HorseVaccineID int identity not null,
    HorseID int not null,
    VaccineID int not null,
    VaccineBatch varchar(20),
    VaccineDate date not null,
    --- This should be created based on the value of VaccineDate (if annual, then add 12 months, etc.)
    --- However, we have a procedure to do that later
    VaccineScheduledDate date,
    VaccineCost decimal(6,2) not null,
    VetID varchar(4) not null,
    -- Constraints on the HorseVaccine table
    CONSTRAINT PK_HorseVaccineID PRIMARY KEY (HorseVaccineID),
    CONSTRAINT FK1_HorseVaccine FOREIGN KEY (HorseID) REFERENCES Horse(HorseID),
    CONSTRAINT FK2_HorseVaccine FOREIGN KEY (VaccineID) REFERENCES Vaccine(VaccineID),
    CONSTRAINT FK3_HorseVaccine FOREIGN KEY (VetID) REFERENCES Vet(VetID)
)
-- End Creating HorseVaccine Table
```

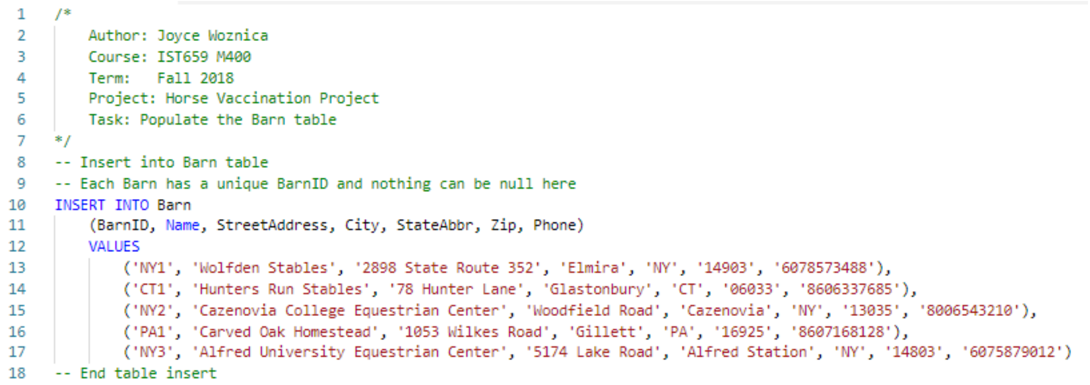
Figure 8: Create HorseVaccine Table SQL Commands

Drop Tables on page 7). Here are some excerpts from these files.



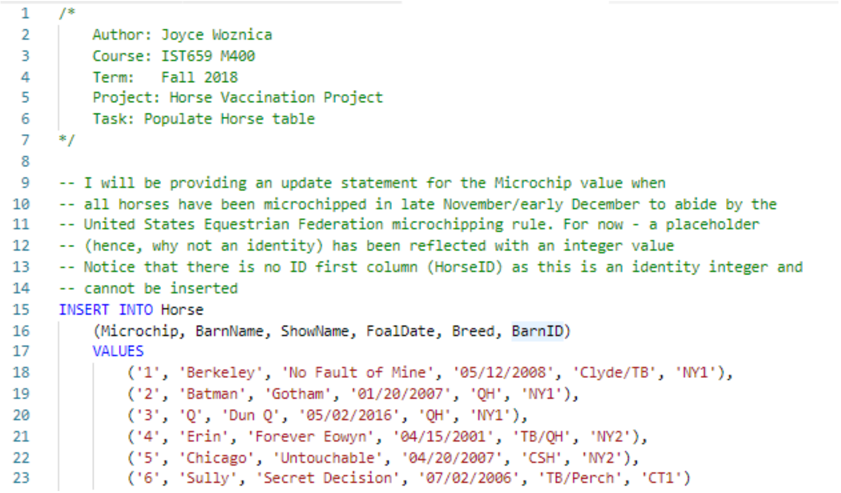
```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate the Vaccine table
*/
-- Enter Vaccines into Vaccine Table. Notice that there is a Potomac - Spring and a Potomac - Fall
-- for this project I did not start with having these as Semi-Annual, but intend to try to change that
-- and then have only one value for Potomac.
INSERT INTO Vaccine
    (Name, Cycle)
    VALUES
        ('E/W/T/Flu/Rhino', 'Annual'),
        ('Potomac - Spring', 'Annual'),
        ('Rabies', 'Annual'),
        ('Coggins', 'Annual'),
        ('West Nile', 'Annual'),
        ('Strangles', 'Annual'),
        ('Potomac - Fall', 'Annual'),
        ('Flu/Rhino - Fall', 'Annual'),
        ('Fecal', 'Annual'),
        ('Lyme Titer', 'Annual')
```

Figure 10: Inserts to Load the Vaccine Table



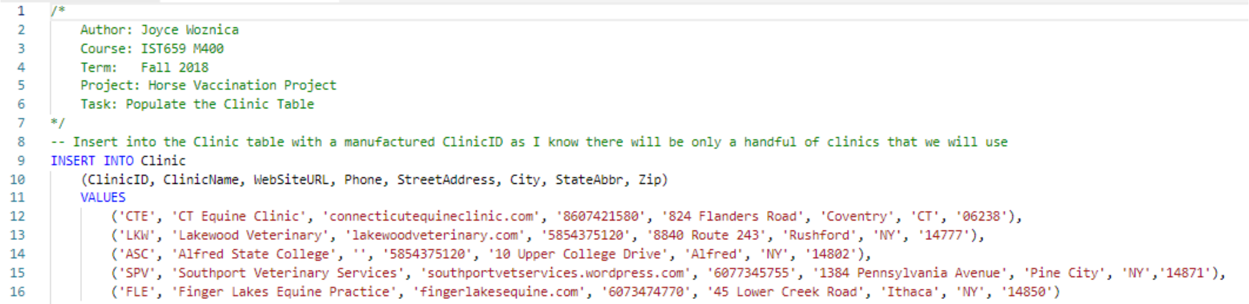
```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate the Barn table
*/
-- Insert into Barn table
-- Each Barn has a unique BarnID and nothing can be null here
INSERT INTO Barn
    (BarnID, Name, StreetAddress, City, StateAbbr, Zip, Phone)
    VALUES
        ('NY1', 'Wolfden Stables', '2898 State Route 352', 'Elmira', 'NY', '14903', '6078573488'),
        ('CT1', 'Hunters Run Stables', '78 Hunter Lane', 'Glastonbury', 'CT', '06033', '8606337685'),
        ('NY2', 'Cazenovia College Equestrian Center', 'Woodfield Road', 'Cazenovia', 'NY', '13035', '8006543210'),
        ('PA1', 'Carved Oak Homestead', '1053 Wilkes Road', 'Gillett', 'PA', '16925', '8607168128'),
        ('NY3', 'Alfred University Equestrian Center', '5174 Lake Road', 'Alfred Station', 'NY', '14803', '6075879012')
-- End table insert
```

Figure 11: Inserts to Load the Barn Table



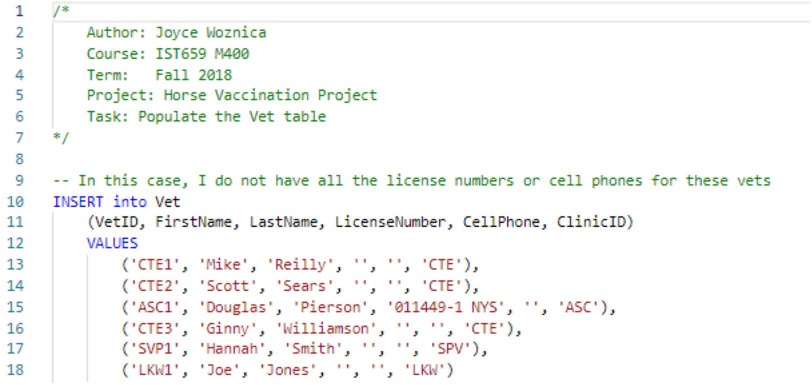
```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate Horse table
*/

-- I will be providing an update statement for the Microchip value when
-- all horses have been microchipped in late November/early December to abide by the
-- United States Equestrian Federation microchipping rule. For now - a placeholder
-- (hence, why not an identity) has been reflected with an integer value
-- Notice that there is no ID first column (HorseID) as this is an identity integer and
-- cannot be inserted
INSERT INTO Horse
    (Microchip, BarnName, ShowName, FoalDate, Breed, BarnID)
    VALUES
        ('1', 'Berkeley', 'No Fault of Mine', '05/12/2008', 'Clyde/TB', 'NY1'),
        ('2', 'Batman', 'Gotham', '01/20/2007', 'QH', 'NY1'),
        ('3', 'Q', 'Dun Q', '05/02/2016', 'QH', 'NY1'),
        ('4', 'Erin', 'Forever Eowyn', '04/15/2001', 'TB/QH', 'NY2'),
        ('5', 'Chicago', 'Untouchable', '04/20/2007', 'CSH', 'NY2'),
        ('6', 'Sully', 'Secret Decision', '07/02/2006', 'TB/Perch', 'CT1')
```

Figure 12: Inserts to Load the Horse Table



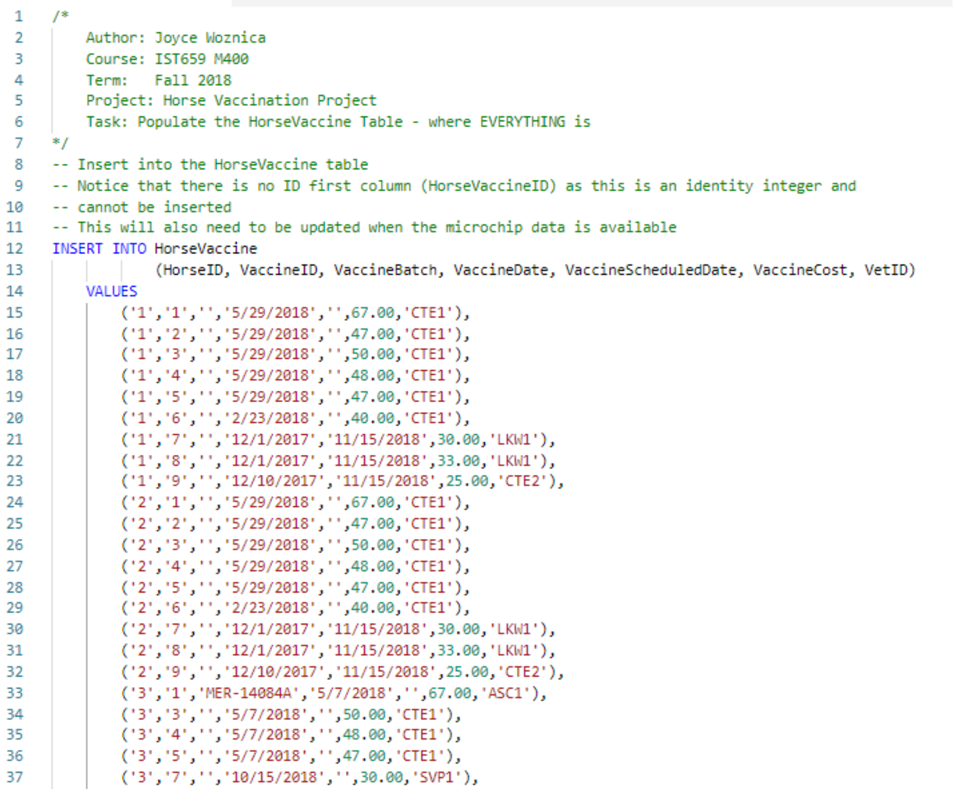
```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate the Clinic Table
*/
-- Insert into the Clinic table with a manufactured ClinicID as I know there will be only a handful of clinics that we will use
INSERT INTO Clinic
    (ClinicID, ClinicName, WebSiteURL, Phone, StreetAddress, City, StateAbbr, Zip)
    VALUES
        ('CTE', 'CT Equine Clinic', 'connecticutequineclinic.com', '8607421580', '824 Flanders Road', 'Coventry', 'CT', '06238'),
        ('LKW', 'Lakewood Veterinary', 'lakewoodveterinary.com', '5854375120', '8840 Route 243', 'Rushford', 'NY', '14777'),
        ('ASC', 'Alfred State College', '', '5854375120', '10 Upper College Drive', 'Alfred', 'NY', '14802'),
        ('SPV', 'Southport Veterinary Services', 'southportvetservices.wordpress.com', '6077345755', '1384 Pennsylvania Avenue', 'Pine City', 'NY','14871'),
        ('FLE', 'Finger Lakes Equine Practice', 'fingerlakesequine.com', '6073474770', '45 Lower Creek Road', 'Ithaca', 'NY', '14850')
```

Figure 13: Inserts to Load the Clinic Table



```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate the Vet table
*/

-- In this case, I do not have all the license numbers or cell phones for these vets
INSERT into Vet
    (VetID, FirstName, LastName, LicenseNumber, CellPhone, ClinicID)
    VALUES
        ('CTE1', 'Mike', 'Reilly', '', '', 'CTE'),
        ('CTE2', 'Scott', 'Sears', '', '', 'CTE'),
        ('ASC1', 'Douglas', 'Pierson', '011449-1 NYS', '', 'ASC'),
        ('CTE3', 'Ginny', 'Williamson', '', '', 'CTE'),
        ('SVP1', 'Hannah', 'Smith', '', '', 'SPV'),
        ('LKW1', 'Joe', 'Jones', '', '', 'LKW')
```

Figure 14: Inserts to Load the Vet Table



```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Populate the HorseVaccine Table - where EVERYTHING is
*/
-- Insert into the HorseVaccine table
-- Notice that there is no ID first column (HorseVaccineID) as this is an identity integer and
-- cannot be inserted
-- This will also need to be updated when the microchip data is available
INSERT INTO HorseVaccine
            (HorseID, VaccineID, VaccineBatch, VaccineDate, VaccineScheduledDate, VaccineCost, VetID)
    VALUES
        ('1','1','','5/29/2018','',67.00,'CTE1'),
        ('1','2','','5/29/2018','',47.00,'CTE1'),
        ('1','3','','5/29/2018','',50.00,'CTE1'),
        ('1','4','','5/29/2018','',48.00,'CTE1'),
        ('1','5','','5/29/2018','',47.00,'CTE1'),
        ('1','6','','2/23/2018','',40.00,'CTE1'),
        ('1','7','','12/1/2017','11/15/2018',30.00,'LKW1'),
        ('1','8','','12/1/2017','11/15/2018',33.00,'LKW1'),
        ('1','9','','12/10/2017','11/15/2018',25.00,'CTE2'),
        ('2','1','','5/29/2018','',67.00,'CTE1'),
        ('2','2','','5/29/2018','',47.00,'CTE1'),
        ('2','3','','5/29/2018','',50.00,'CTE1'),
        ('2','4','','5/29/2018','',48.00,'CTE1'),
        ('2','5','','5/29/2018','',47.00,'CTE1'),
        ('2','6','','2/23/2018','',40.00,'CTE1'),
        ('2','7','','12/1/2017','11/15/2018',30.00,'LKW1'),
        ('2','8','','12/1/2017','11/15/2018',33.00,'LKW1'),
        ('2','9','','12/10/2017','11/15/2018',25.00,'CTE2'),
        ('3','1','MER-14084A','5/7/2018','',67.00,'ASC1'),
        ('3','3','','5/7/2018','',50.00,'CTE1'),
        ('3','4','','5/7/2018','',48.00,'CTE1'),
        ('3','5','','5/7/2018','',47.00,'CTE1'),
        ('3','7','','10/15/2018','',30.00,'SVP1'),
```

Figure 15: Some Representative Inserts to Load the HorseVaccine Table

Data Manipulation

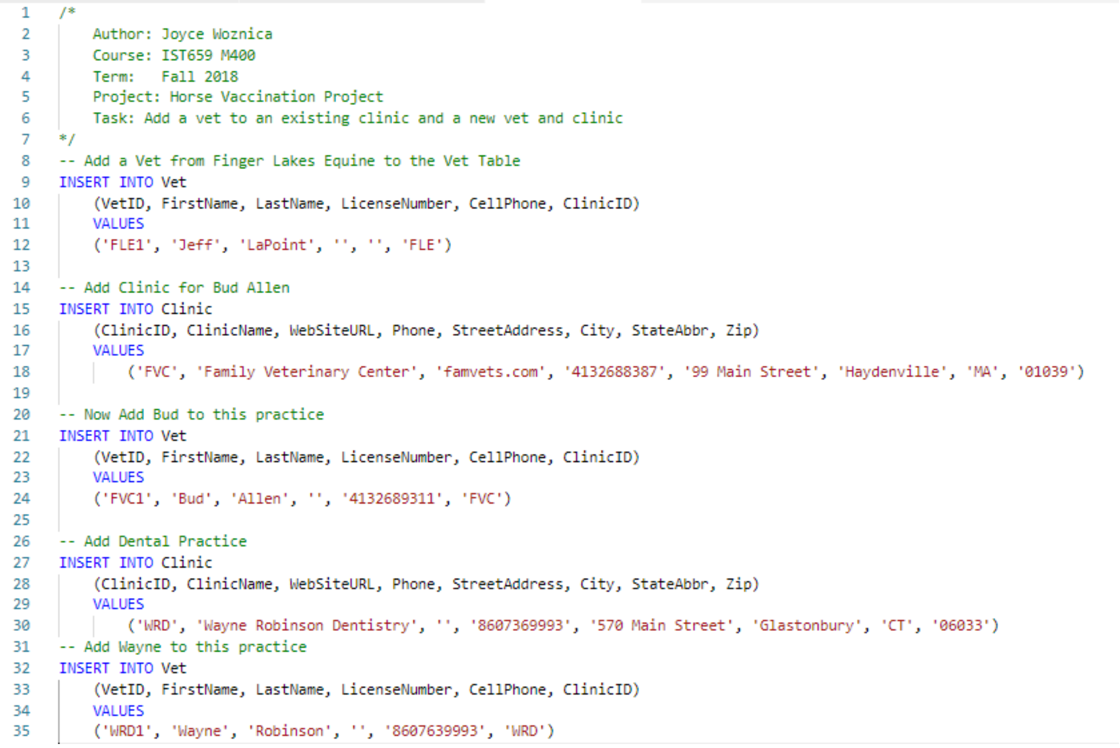
I have provided some updates, deletions, alters, etc. to the data based on what I had to do in the short term to provide changes to the table data.

Adding some Data to the Tables

I decided to provide some additional information to the Clinic and Vet tables as well as the Vaccine table to support vaccines recently done in the last two weeks. In order to do that, I created a couple of general SQL scripts with INSERT statements to accommodate those changes.

AddVet.sql

This adds 3 vets and 2 clinics to associated tables.

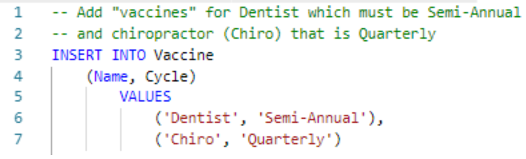


```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Add a vet to an existing clinic and a new vet and clinic
*/
-- Add a Vet from Finger Lakes Equine to the Vet Table
INSERT INTO Vet
    (VetID, FirstName, LastName, LicenseNumber, CellPhone, ClinicID)
    VALUES
    ('FLE1', 'Jeff', 'LaPoint', '', '', 'FLE')

-- Add Clinic for Bud Allen
INSERT INTO Clinic
    (ClinicID, ClinicName, WebSiteURL, Phone, StreetAddress, City, StateAbbr, Zip)
    VALUES
        ('FVC', 'Family Veterinary Center', 'famvets.com', '4132688387', '99 Main Street', 'Haydenville', 'MA', '01039')

-- Now Add Bud to this practice
INSERT INTO Vet
    (VetID, FirstName, LastName, LicenseNumber, CellPhone, ClinicID)
    VALUES
    ('FVC1', 'Bud', 'Allen', '', '4132689311', 'FVC')

-- Add Dental Practice
INSERT INTO Clinic
    (ClinicID, ClinicName, WebSiteURL, Phone, StreetAddress, City, StateAbbr, Zip)
    VALUES
        ('WRD', 'Wayne Robinson Dentistry', '', '8607369993', '570 Main Street', 'Glastonbury', 'CT', '06033')
-- Add Wayne to this practice
INSERT INTO Vet
    (VetID, FirstName, LastName, LicenseNumber, CellPhone, ClinicID)
    VALUES
    ('WRD1', 'Wayne', 'Robinson', '', '8607639993', 'WRD')
```

Figure 16: AddVet INSERT statements to Add Data to Vet and Clinic Tables

AddDentistandChiro.sql

This function adds entries into the ***Vaccine*** table for the *Chiropractor* on a *Quarterly* basis and the Dentist on a Semi-Annual basis.

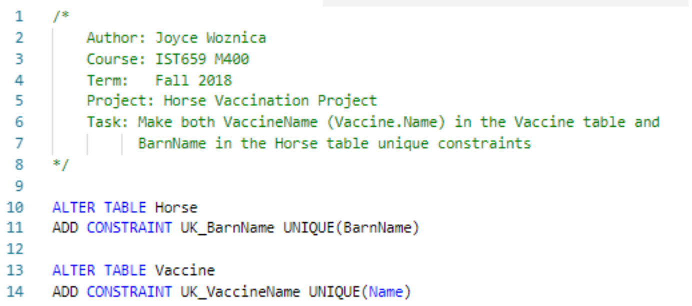


```sql
-- Add "vaccines" for Dentist which must be Semi-Annual
-- and chiropractor (Chiro) that is Quarterly
INSERT INTO Vaccine
    (Name, Cycle)
        VALUES
            ('Dentist', 'Semi-Annual'),
            ('Chiro', 'Quarterly')
```

Figure 17: AddDentistandChiro INSERT statements to Add Data to Vaccine Table

MakeBarnNameandVaccineNameUnique.sql

This file updates the ***Horse*** table construct and the ***Vaccine*** table construct to make *BarnName* and *Name* UNIQUE constraints, respectively.



```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Make both VaccineName (Vaccine.Name) in the Vaccine table and
          BarnName in the Horse table unique constraints
*/

ALTER TABLE Horse
ADD CONSTRAINT UK_BarnName UNIQUE(BarnName)

ALTER TABLE Vaccine
ADD CONSTRAINT UK_VaccineName UNIQUE(Name)
```

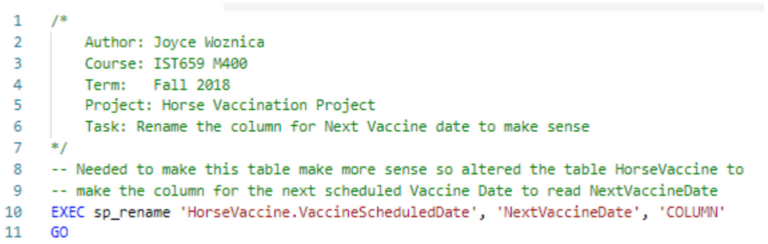
Figure 18: MakeBarnNameandVaccineNameUnique ALTER TABLE statements

Functions, Procedures, Views, and More

In addition to just update the tables a bit with various information, I also created several functions, procedures, views and other items to manipulate the data and to make certain procedures and functions available for what will become the UI that I plan to create.

RenameScheduleDate.sql

This alters a table (using the MSSQL “sp_rename” function) to change a column name in the ***HorseVaccine*** table.

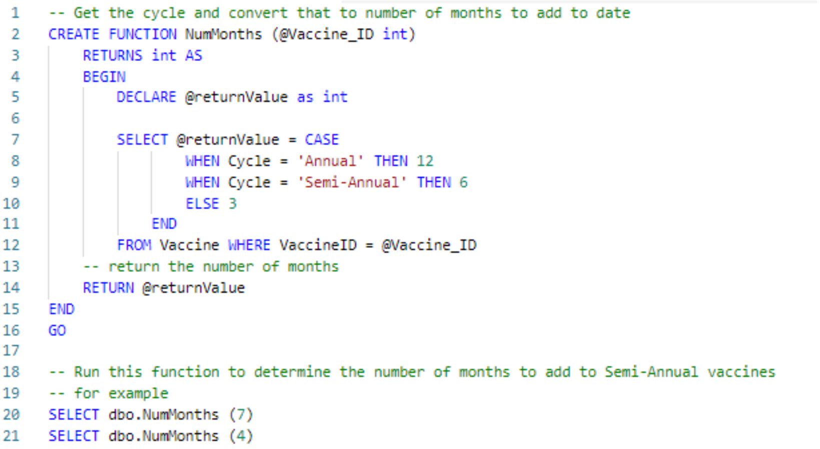


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Rename the column for Next Vaccine date to make sense
*/
-- Needed to make this table make more sense so altered the table HorseVaccine to
-- make the column for the next scheduled Vaccine Date to read NextVaccineDate
EXEC sp_rename 'HorseVaccine.VaccineScheduledDate', 'NextVaccineDate', 'COLUMN'
GO
```

Figure 19: RenameScheduleDate Execution of MSSQL Function to Modify the HorseVaccine Table

NumMonthsFunction.sql

This is a function that will determine the number of months to add to the *VaccineDate* to derive the *ScheduledVaccineDate* based on the *Cycle* in the ***Vaccine*** table. It accepts the *VaccineID*. Included is how to execute this function for a specific vaccine.

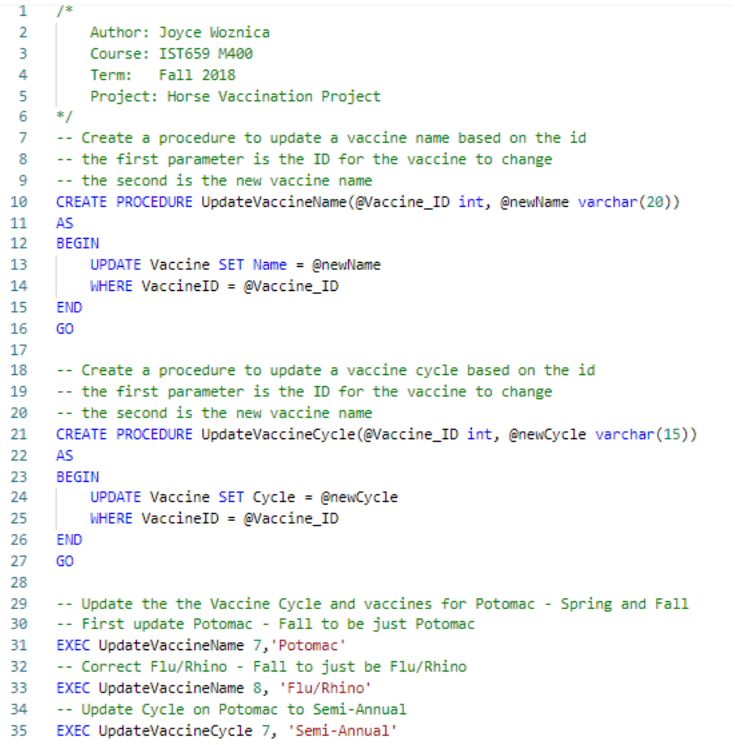


```
-- Get the cycle and convert that to number of months to add to date
CREATE FUNCTION NumMonths (@Vaccine_ID int)
    RETURNS int AS
    BEGIN
        DECLARE @returnValue as int

        SELECT @returnValue = CASE
                WHEN Cycle = 'Annual' THEN 12
                WHEN Cycle = 'Semi-Annual' THEN 6
                ELSE 3
            END
        FROM Vaccine WHERE VaccineID = @Vaccine_ID
    -- return the number of months
    RETURN @returnValue
END
GO

-- Run this function to determine the number of months to add to Semi-Annual vaccines
-- for example
SELECT dbo.NumMonths (7)
SELECT dbo.NumMonths (4)
```

Figure 20: NumMonths Function to Determine NextVaccineDate

VaccineSchedules.sql

This code provides two stored procedures that provide updates to the ***Vaccine*** table. One for changing the name of a vaccine and one for modifying the cycle for a vaccine.

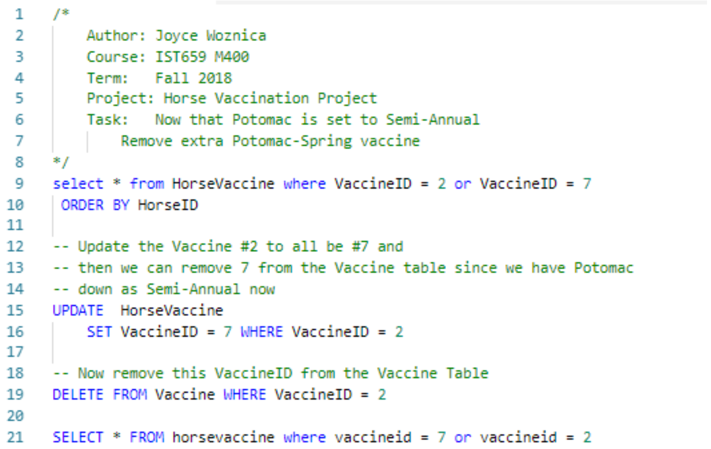


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
*/
-- Create a procedure to update a vaccine name based on the id
-- the first parameter is the ID for the vaccine to change
-- the second is the new vaccine name
CREATE PROCEDURE UpdateVaccineName(@Vaccine_ID int, @newName varchar(20))
AS
BEGIN
    UPDATE Vaccine SET Name = @newName
    WHERE VaccineID = @Vaccine_ID
END
GO

-- Create a procedure to update a vaccine cycle based on the id
-- the first parameter is the ID for the vaccine to change
-- the second is the new vaccine name
CREATE PROCEDURE UpdateVaccineCycle(@Vaccine_ID int, @newCycle varchar(15))
AS
BEGIN
    UPDATE Vaccine SET Cycle = @newCycle
    WHERE VaccineID = @Vaccine_ID
END
GO

-- Update the the Vaccine Cycle and vaccines for Potomac - Spring and Fall
-- First update Potomac - Fall to be just Potomac
EXEC UpdateVaccineName 7,'Potomac'
-- Correct Flu/Rhino - Fall to just be Flu/Rhino
EXEC UpdateVaccineName 8, 'Flu/Rhino'
-- Update Cycle on Potomac to Semi-Annual
EXEC UpdateVaccineCycle 7, 'Semi-Annual'
```

Figure 21: VaccineSchedules Stored Procedures to Allow Updates of the Vaccine Table

RemoveExtraVaccine.sql

This code is provided in this section because of the order I needed to do things based on my original design (which I elected not to change as I learned so much by taking that approach). This renames all the 'Potomac – Spring' (VaccineID #2) to 7 (Just 'Potomac') in the ***HorseVaccine*** table, then I was able to delete the extra 'Potomac' entry and use the code shown in this document to make 'Potomac' a Semi-Annual vaccine instead of having two similar entries once per year.



```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task:   Now that Potomac is set to Semi-Annual
            Remove extra Potomac-Spring vaccine
*/
select * from HorseVaccine where VaccineID = 2 or VaccineID = 7
 ORDER BY HorseID

-- Update the Vaccine #2 to all be #7 and
-- then we can remove 7 from the Vaccine table since we have Potomac
-- down as Semi-Annual now
UPDATE  HorseVaccine
    SET VaccineID = 7 WHERE VaccineID = 2

-- Now remove this VaccineID from the Vaccine Table
DELETE FROM Vaccine WHERE VaccineID = 2

SELECT * FROM horsevaccine where vaccineid = 7 or vaccineid = 2
```

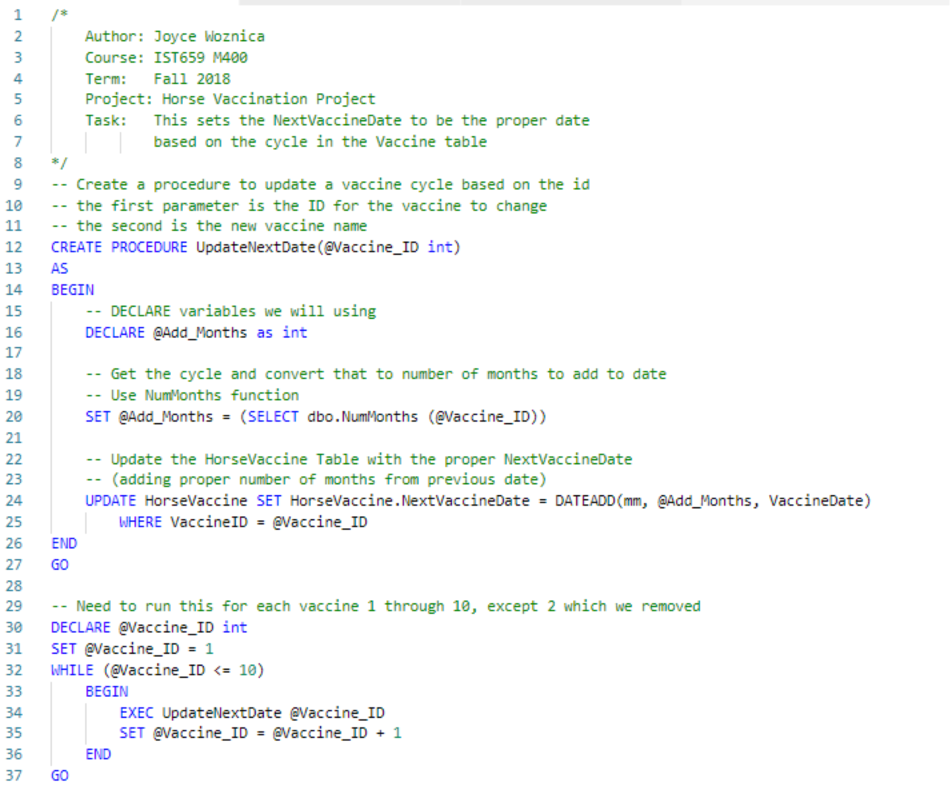
Figure 22: RemoveExtraVaccine UPDATE, DELETE to Update Vaccine, HorseVaccine Tables

UpdateNextVaccineDate.sql

Updates the *NextVaccineDate* on all rows to be the proper date based on the *Cycle* for the specific *Vaccine* in that row.

This was one of the most difficult for me to write because of the need to make a massive change to the entire table (which will not be required again because of the *AddHorseVaccine* procedure that appears in the section Answering Data Questions page 18.

The WHILE loop provides an update looping through all the vaccines in the ***Vaccine*** table.

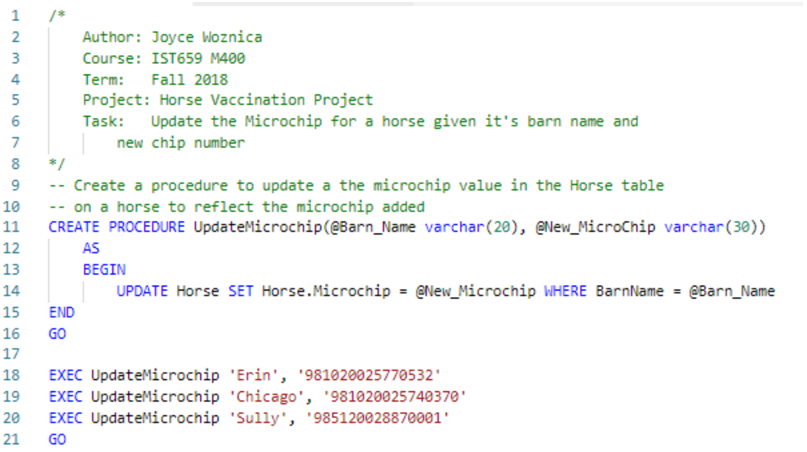


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task:   This sets the NextVaccineDate to be the proper date
            based on the cycle in the Vaccine table
*/
-- Create a procedure to update a vaccine cycle based on the id
-- the first parameter is the ID for the vaccine to change
-- the second is the new vaccine name
CREATE PROCEDURE UpdateNextDate(@Vaccine_ID int)
AS
BEGIN
    -- DECLARE variables we will using
    DECLARE @Add_Months as int

    -- Get the cycle and convert that to number of months to add to date
    -- Use NumMonths function
    SET @Add_Months = (SELECT dbo.NumMonths (@Vaccine_ID))

    -- Update the HorseVaccine Table with the proper NextVaccineDate
    -- (adding proper number of months from previous date)
    UPDATE HorseVaccine SET HorseVaccine.NextVaccineDate = DATEADD(mm, @Add_Months, VaccineDate)
        WHERE VaccineID = @Vaccine_ID
END
GO

-- Need to run this for each vaccine 1 through 10, except 2 which we removed
DECLARE @Vaccine_ID int
SET @Vaccine_ID = 1
WHILE (@Vaccine_ID <= 10)
    BEGIN
        EXEC UpdateNextDate @Vaccine_ID
        SET @Vaccine_ID = @Vaccine_ID + 1
    END
GO
```

Figure 23: UpdateNextVaccineDate Procedure with loop to Update tables

UpdateMicrochip.sql

This file updates based on the current microchip number, the new one reported by the vet when inserted.



```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task:   Update the Microchip for a horse given it's barn name and
        new chip number
*/
-- Create a procedure to update a the microchip value in the Horse table
-- on a horse to reflect the microchip added
CREATE PROCEDURE UpdateMicrochip(@Barn_Name varchar(20), @New_MicroChip varchar(30))
    AS
    BEGIN
        UPDATE Horse SET Horse.Microchip = @New_Microchip WHERE BarnName = @Barn_Name
END
GO

EXEC UpdateMicrochip 'Erin', '981020025770532'
EXEC UpdateMicrochip 'Chicago', '981020025740370'
EXEC UpdateMicrochip 'Sully', '985120028870001'
GO
```

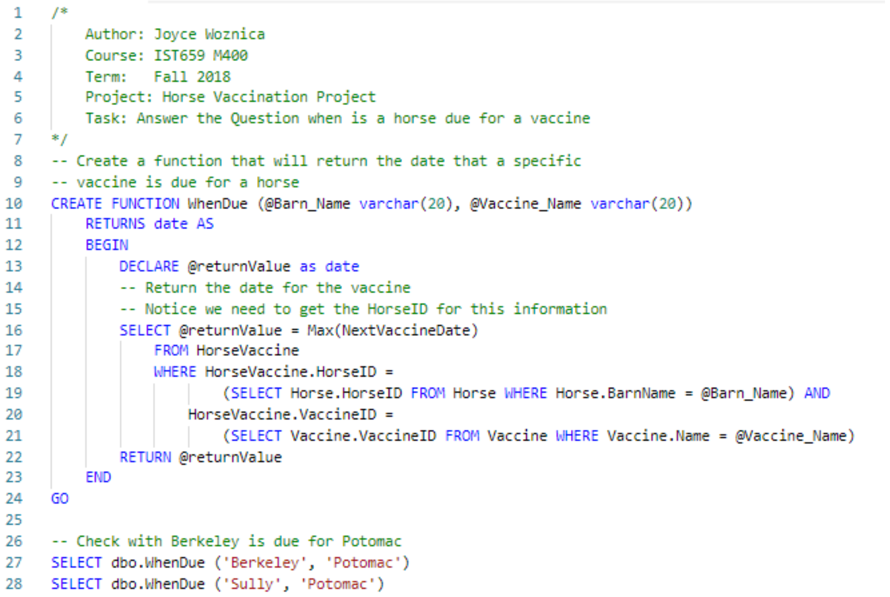
Figure 24: UpdateMicrochip Stored Procedure to add Microchip data to the Horse Table

Answering Data Questions

There are several questions that I would like for this data to answer. Here is some of the queries that I put together to answer some of these questions. Most of my question provide single line answers as that was my goal. These are not reports, but in Implementation section starting on page 23, I have provided some reports on the data as well.

When is a Horse Due for a Specific Vaccine (*WhenDue.sql*)

This function uses others created (like NumMonths for deriving the schedule date for a vaccine) to provide the answer to the question "When is <HORSE> due for <VACCINE>?" The code below also shows how the function is called. As shown in my Normalized Model on page 6, the *BarnName* (in the ***Horse*** table) and *Name* (in the ***Vaccine*** table) were made unique with ALTER TABLE commands provided in MakeBarnNameandVaccineNameUnique.sql section that is shown on page 15.

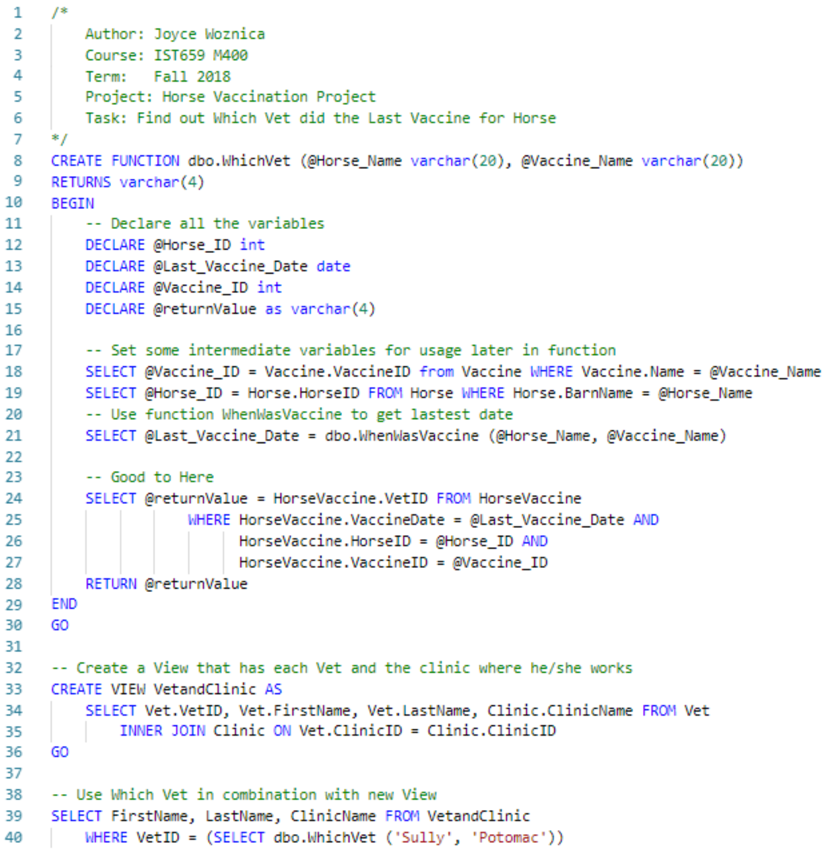


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Answer the Question when is a horse due for a vaccine
*/
-- Create a function that will return the date that a specific
-- vaccine is due for a horse
CREATE FUNCTION WhenDue (@Barn_Name varchar(20), @Vaccine_Name varchar(20))
    RETURNS date AS
    BEGIN
        DECLARE @returnValue as date
        -- Return the date for the vaccine
        -- Notice we need to get the HorseID for this information
        SELECT @returnValue = Max(NextVaccineDate)
            FROM HorseVaccine
            WHERE HorseVaccine.HorseID =
                    (SELECT Horse.HorseID FROM Horse WHERE Horse.BarnName = @Barn_Name) AND
                HorseVaccine.VaccineID =
                    (SELECT Vaccine.VaccineID FROM Vaccine WHERE Vaccine.Name = @Vaccine_Name)
        RETURN @returnValue
    END
GO

-- Check with Berkeley is due for Potomac
SELECT dbo.WhenDue ('Berkeley', 'Potomac')
SELECT dbo.WhenDue ('Sully', 'Potomac')
```

Figure 25: WhenDue Function returns when a horse is due for the noted Vaccine

Which Vet Performed the Last Vaccination of Vaccine (*WhichVet.sql*)

This function uses others created (like *WhenWasVaccine* for deriving the last date for a vaccine) to provide the answer to the question “What vet did <HORSE>’s last <VACCINE>?” This includes a view that joins the ***Vet*** and ***Clinic*** tables to provide a list of the VetID, Vet First Name, Vet Last Name and Clinic where he/she works as a response to this question.

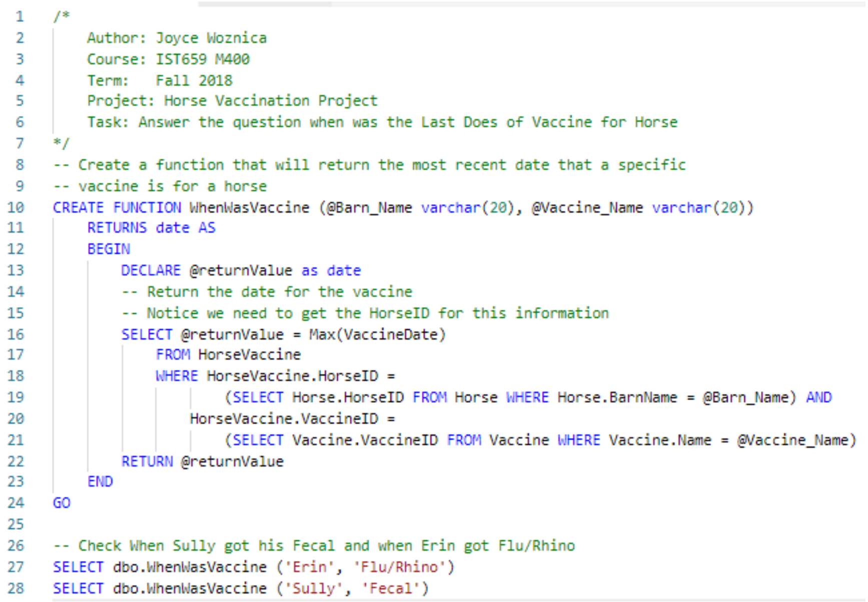


```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Find out Which Vet did the Last Vaccine for Horse
*/
CREATE FUNCTION dbo.WhichVet (@Horse_Name varchar(20), @Vaccine_Name varchar(20))
RETURNS varchar(4)
BEGIN
    -- Declare all the variables
    DECLARE @Horse_ID int
    DECLARE @Last_Vaccine_Date date
    DECLARE @Vaccine_ID int
    DECLARE @returnValue as varchar(4)

    -- Set some intermediate variables for usage later in function
    SELECT @Vaccine_ID = Vaccine.VaccineID from Vaccine WHERE Vaccine.Name = @Vaccine_Name
    SELECT @Horse_ID = Horse.HorseID FROM Horse WHERE Horse.BarnName = @Horse_Name
    -- Use function WhenWasVaccine to get lastest date
    SELECT @Last_Vaccine_Date = dbo.WhenWasVaccine (@Horse_Name, @Vaccine_Name)

    -- Good to Here
    SELECT @returnValue = HorseVaccine.VetID FROM HorseVaccine
                WHERE HorseVaccine.VaccineDate = @Last_Vaccine_Date AND
                      HorseVaccine.HorseID = @Horse_ID AND
                      HorseVaccine.VaccineID = @Vaccine_ID
    RETURN @returnValue
END
GO

-- Create a View that has each Vet and the clinic where he/she works
CREATE VIEW VetandClinic AS
    SELECT Vet.VetID, Vet.FirstName, Vet.LastName, Clinic.ClinicName FROM Vet
        INNER JOIN Clinic ON Vet.ClinicID = Clinic.ClinicID
GO

-- Use Which Vet in combination with new View
SELECT FirstName, LastName, ClinicName FROM VetandClinic
    WHERE VetID = (SELECT dbo.WhichVet ('Sully', 'Potomac'))
```

Figure 26: WhichVet Function returns which Vet did a Horse's last Vaccine

When was a Vaccine Performed (*WhenWasVaccine.sql*)

This function answers the question "When did <HORSE> get the last dose of <VACCINE>?" given a specific horse name and vaccine name.

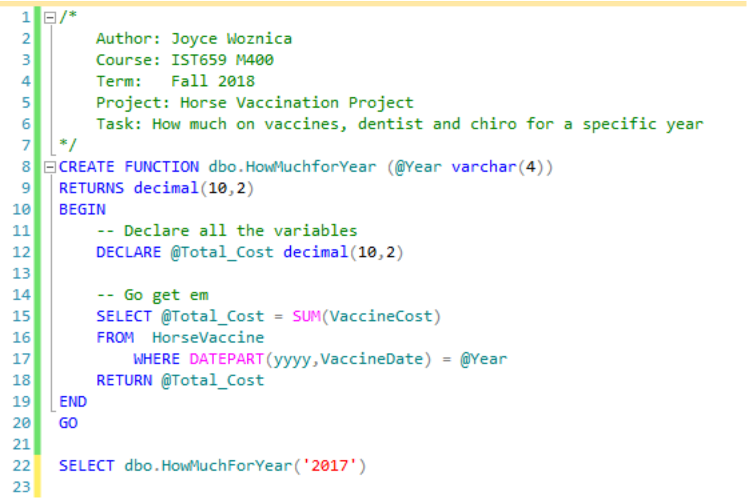


```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Answer the question when was the Last Does of Vaccine for Horse
*/
-- Create a function that will return the most recent date that a specific
-- vaccine is for a horse
CREATE FUNCTION WhenWasVaccine (@Barn_Name varchar(20), @Vaccine_Name varchar(20))
    RETURNS date AS
    BEGIN
        DECLARE @returnValue as date
        -- Return the date for the vaccine
        -- Notice we need to get the HorseID for this information
        SELECT @returnValue = Max(VaccineDate)
            FROM HorseVaccine
            WHERE HorseVaccine.HorseID =
                    (SELECT Horse.HorseID FROM Horse WHERE Horse.BarnName = @Barn_Name) AND
                HorseVaccine.VaccineID =
                    (SELECT Vaccine.VaccineID FROM Vaccine WHERE Vaccine.Name = @Vaccine_Name)
        RETURN @returnValue
    END
GO

-- Check When Sully got his Fecal and when Erin got Flu/Rhino
SELECT dbo.WhenWasVaccine ('Erin', 'Flu/Rhino')
SELECT dbo.WhenWasVaccine ('Sully', 'Fecal')
```

Figure 27: WhenWasVaccine Function returns when a horse is got the last dose of Vaccine

How much money was spent on vaccines in a given year (*HowMuchForYear.sql*)

This function answers the question “How much money was spent in <Year> on vaccines?” given a specific year.

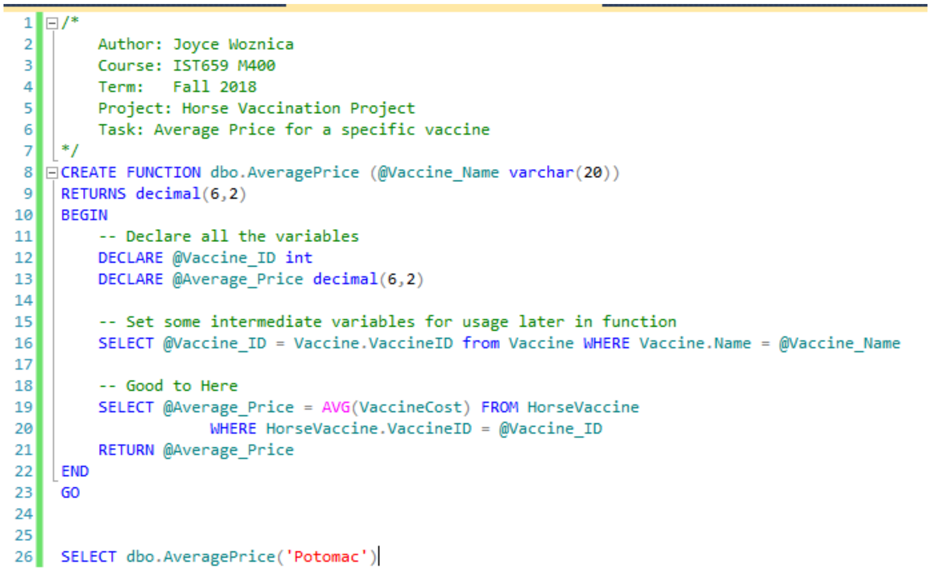


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: How much on vaccines, dentist and chiro for a specific year
*/
CREATE FUNCTION dbo.HowMuchforYear (@Year varchar(4))
RETURNS decimal(10,2)
BEGIN
    -- Declare all the variables
    DECLARE @Total_Cost decimal(10,2)

    -- Go get em
    SELECT @Total_Cost = SUM(VaccineCost)
    FROM  HorseVaccine
        WHERE DATEPART(yyyy,VaccineDate) = @Year
    RETURN @Total_Cost
END
GO

SELECT dbo.HowMuchForYear('2017')
```

Figure 28: HowMuchForYear Function returns a currency value

Average Price for a Specific Vaccine (*AveragePrice.sql*)

This function answers the question “What is the Average Price for <Vaccine>?”

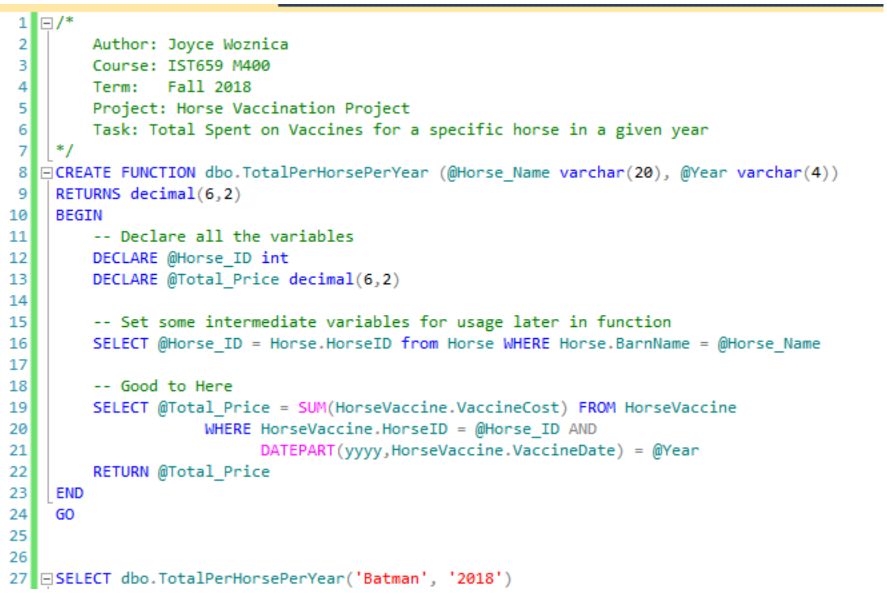


```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Average Price for a specific vaccine
*/
CREATE FUNCTION dbo.AveragePrice (@Vaccine_Name varchar(20))
RETURNS decimal(6,2)
BEGIN
    -- Declare all the variables
    DECLARE @Vaccine_ID int
    DECLARE @Average_Price decimal(6,2)

    -- Set some intermediate variables for usage later in function
    SELECT @Vaccine_ID = Vaccine.VaccineID from Vaccine WHERE Vaccine.Name = @Vaccine_Name

    -- Good to Here
    SELECT @Average_Price = AVG(VaccineCost) FROM HorseVaccine
            WHERE HorseVaccine.VaccineID = @Vaccine_ID
    RETURN @Average_Price
END
GO


SELECT dbo.AveragePrice('Potomac')
```

Figure 29: AveragePrice Function returns a currency value

How much money was spent on vaccines in a given year for a horse (*TotalPerHorsePerYear.sql*)

This function answers the question “How much money was spent in <Year> on vaccines for <Horse>?”



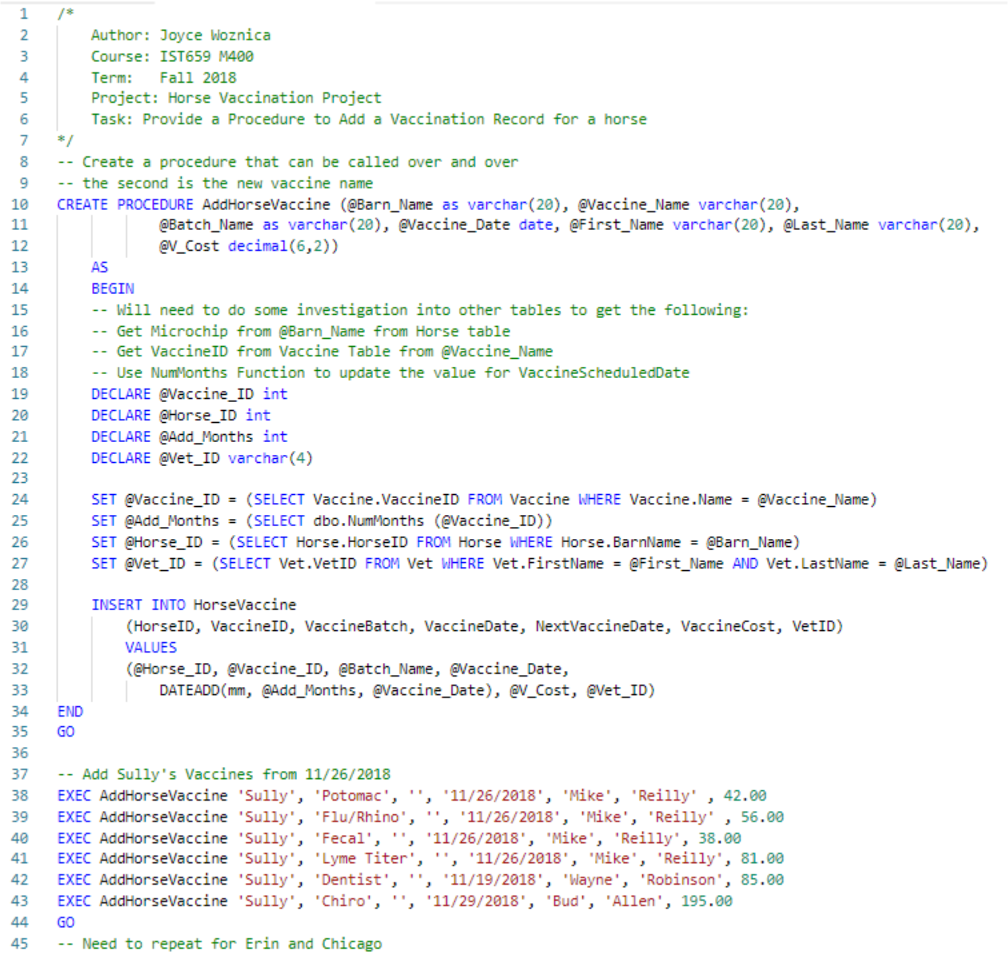
```
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Total Spent on Vaccines for a specific horse in a given year
*/
CREATE FUNCTION dbo.TotalPerHorsePerYear (@Horse_Name varchar(20), @Year varchar(4))
RETURNS decimal(6,2)
BEGIN
    -- Declare all the variables
    DECLARE @Horse_ID int
    DECLARE @Total_Price decimal(6,2)

    -- Set some intermediate variables for usage later in function
    SELECT @Horse_ID = Horse.HorseID from Horse WHERE Horse.BarnName = @Horse_Name

    -- Good to Here
    SELECT @Total_Price = SUM(HorseVaccine.VaccineCost) FROM HorseVaccine
                WHERE HorseVaccine.HorseID = @Horse_ID AND
                      DATEPART(yyyy,HorseVaccine.VaccineDate) = @Year
    RETURN @Total_Price
END
GO


SELECT dbo.TotalPerHorsePerYear('Batman', '2018')
```

Figure 30: TotalPerHorsePerYear Function returns a currency value

General Maintenance

AddHorseVaccine.sql

This file updates the **HorseVaccine** table with a recent vaccine passing it the required parameters to update the row. It includes running the *NumMonths* function to calculate the *NextVaccineDate* using the proper *Cycle*. This is key to the application as we will need to update the database with this function every time a horse gets his/her next vaccine.



```sql
/*
    Author: Joyce Woznica
    Course: IST659 M400
    Term:   Fall 2018
    Project: Horse Vaccination Project
    Task: Provide a Procedure to Add a Vaccination Record for a horse
*/
-- Create a procedure that can be called over and over
-- the second is the new vaccine name
CREATE PROCEDURE AddHorseVaccine (@Barn_Name as varchar(20), @Vaccine_Name varchar(20),
            @Batch_Name as varchar(20), @Vaccine_Date date, @First_Name varchar(20), @Last_Name varchar(20),
            @V_Cost decimal(6,2))
    AS
    BEGIN
    -- Will need to do some investigation into other tables to get the following:
    -- Get Microchip from @Barn_Name from Horse table
    -- Get VaccineID from Vaccine Table from @Vaccine_Name
    -- Use NumMonths Function to update the value for VaccineScheduledDate
    DECLARE @Vaccine_ID int
    DECLARE @Horse_ID int
    DECLARE @Add_Months int
    DECLARE @Vet_ID varchar(4)

    SET @Vaccine_ID = (SELECT Vaccine.VaccineID FROM Vaccine WHERE Vaccine.Name = @Vaccine_Name)
    SET @Add_Months = (SELECT dbo.NumMonths (@Vaccine_ID))
    SET @Horse_ID = (SELECT Horse.HorseID FROM Horse WHERE Horse.BarnName = @Barn_Name)
    SET @Vet_ID = (SELECT Vet.VetID FROM Vet WHERE Vet.FirstName = @First_Name AND Vet.LastName = @Last_Name)

    INSERT INTO HorseVaccine
        (HorseID, VaccineID, VaccineBatch, VaccineDate, NextVaccineDate, VaccineCost, VetID)
        VALUES
        (@Horse_ID, @Vaccine_ID, @Batch_Name, @Vaccine_Date,
            DATEADD(mm, @Add_Months, @Vaccine_Date), @V_Cost, @Vet_ID)
END
GO

-- Add Sully's Vaccines from 11/26/2018
EXEC AddHorseVaccine 'Sully', 'Potomac', '', '11/26/2018', 'Mike', 'Reilly' , 42.00
EXEC AddHorseVaccine 'Sully', 'Flu/Rhino', '', '11/26/2018', 'Mike', 'Reilly' , 56.00
EXEC AddHorseVaccine 'Sully', 'Fecal', '', '11/26/2018', 'Mike', 'Reilly', 38.00
EXEC AddHorseVaccine 'Sully', 'Lyme Titer', '', '11/26/2018', 'Mike', 'Reilly', 81.00
EXEC AddHorseVaccine 'Sully', 'Dentist', '', '11/19/2018', 'Wayne', 'Robinson', 85.00
EXEC AddHorseVaccine 'Sully', 'Chiro', '', '11/29/2018', 'Bud', 'Allen', 195.00
GO
-- Need to repeat for Erin and Chicago
```

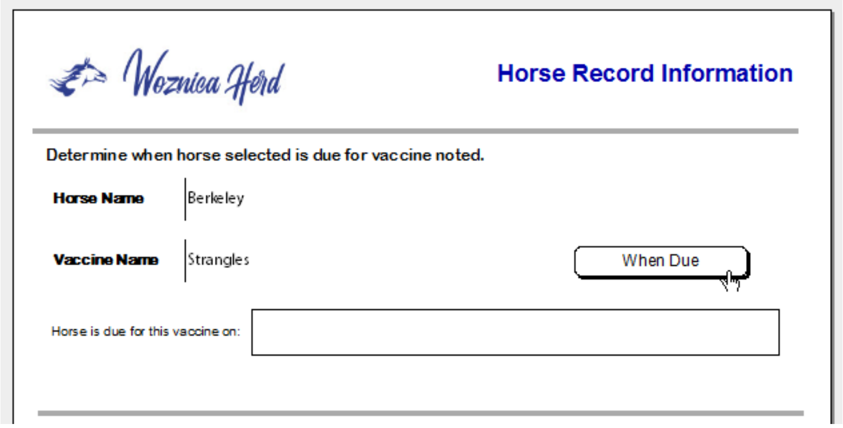
Figure 31: AddHorseVaccine Stored Procedure that Updates the HorseVaccine Table

Implementation

I used two approaches for implementing a user interface to my database. I wanted something easy to use for the end user (including myself) to allow for accessing the data and obtaining the information desired.

Phase 1 – General Screens using eForm Tool

Originally, I started out attempting to use an eForm product to which I am family as it allows database lookups and connection to Microsoft SQL Server through JDBC connections which I have used before; however, this proved to be quite cumbersome. I was able to at least draw what the forms I wanted to create would look like and display the data as I felt it might appear. The following screenshots show some of these ideas.



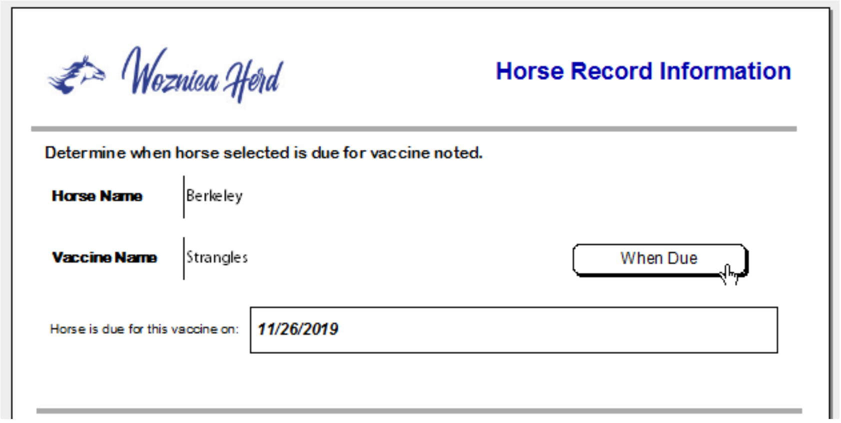
Determine when horse selected is due for vaccine noted.

Horse Name Berkeley

Vaccine Name Strangles

Horse is due for this vaccine on:

Figure 32: Determine when Horse is Due for Vaccine



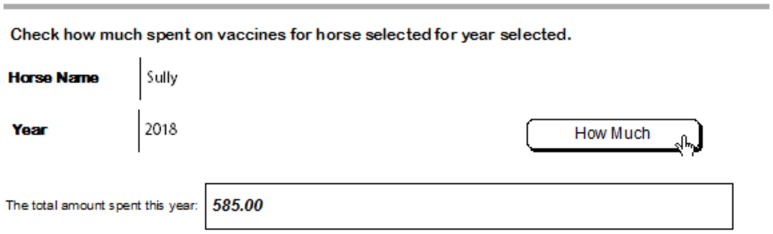
Determine when horse selected is due for vaccine noted.

Horse Name Berkeley

Vaccine Name Strangles

Horse is due for this vaccine on: 11/26/2019

Figure 33: After Clicking "When Due" Response displayed

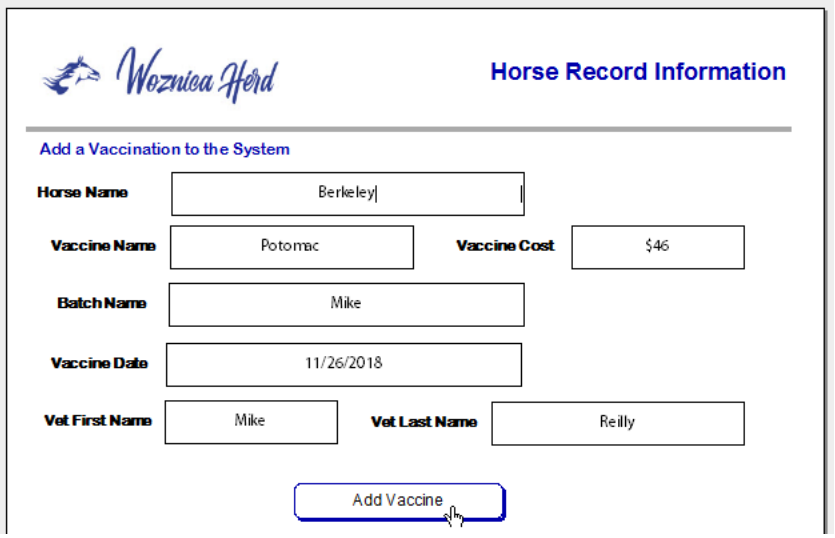

Check how much spent on vaccines for horse selected for year selected.

Horse Name Sully

Year 2018

The total amount spent this year: 585.00

Figure 34: Determining How Much was Spent on Vaccines for a Particular Horse in a Year



Woznica Herd

Horse Record Information

Add a Vaccination to the System

Horse Name: Berkeley

Vaccine Name: Potomac
Vaccine Cost: $46

Batch Name: Mike

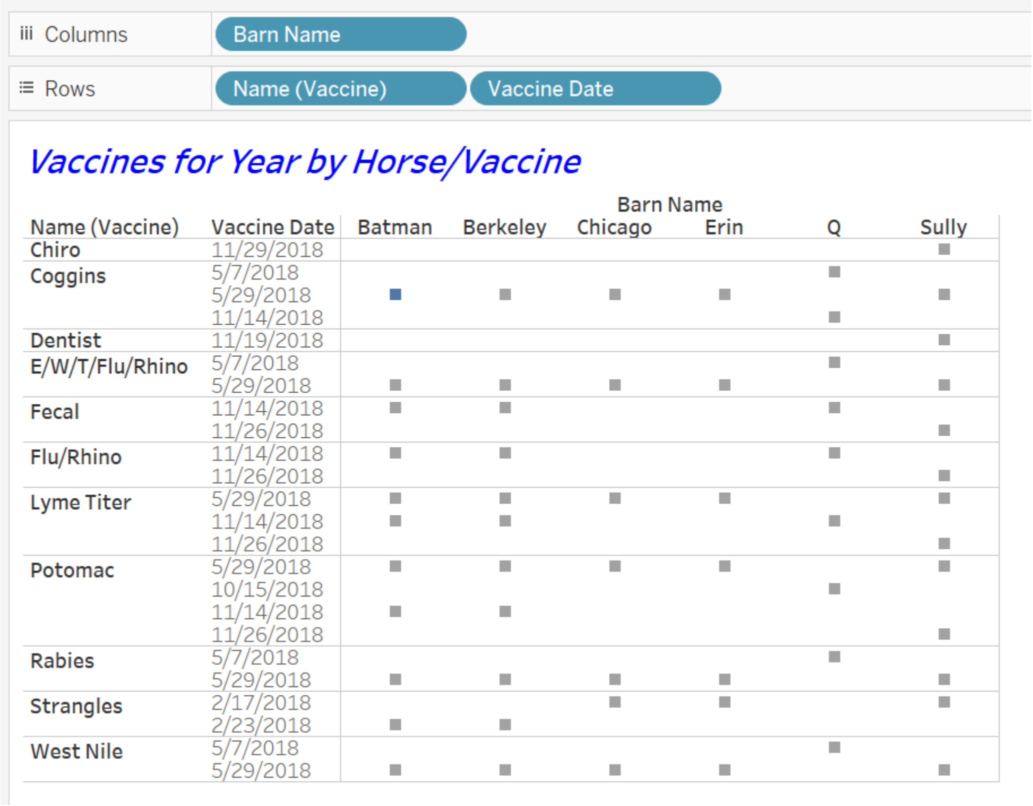
Vaccine Date: 11/26/2018

Vet First Name: Mike
Vet Last Name: Reilly

Figure 35: Form to Add a Vaccine to the Database (Date and Cost are defined as date and float)

Phase 2 – Tableau as a User Design Tool

My second approach was using Tableau to design some ways to display the data in informative ways for the end user. I have not used this tool before, so there was a learning curve associated with this approach.

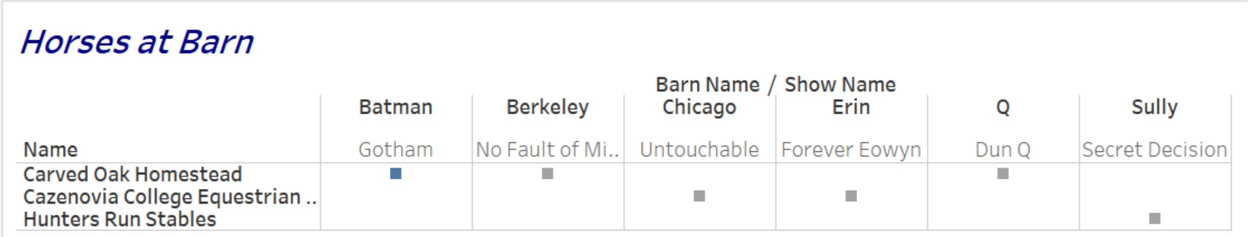


Columns: Barn Name
Rows: Name (Vaccine), Vaccine Date

Vaccines for Year by Horse/Vaccine

		Barn Name					
Name (Vaccine)	Vaccine Date	Batman	Berkeley	Chicago	Erin	Q	Sully
Chiro	11/29/2018						■
Coggins	5/7/2018					■	
	5/29/2018	■	■	■	■		■
	11/14/2018					■	
Dentist	11/19/2018						■
E/W/T/Flu/Rhino	5/7/2018					■	
	5/29/2018	■	■	■	■		■
Fecal	11/14/2018	■	■			■	
	11/26/2018						■
Flu/Rhino	11/14/2018	■	■			■	
	11/26/2018						■
Lyme Titer	5/29/2018	■	■	■	■		■
	11/14/2018	■	■			■	
	11/26/2018						■
Potomac	5/29/2018	■	■	■	■		■
	10/15/2018					■	
	11/14/2018	■	■				
	11/26/2018						■
Rabies	5/7/2018					■	
	5/29/2018	■	■	■	■		■
Strangles	2/17/2018			■	■		■
	2/23/2018	■	■				
West Nile	5/7/2018					■	
	5/29/2018	■	■	■	■		■

Figure 36: Worksheet to Display Vaccines by Horse by Vaccine Date

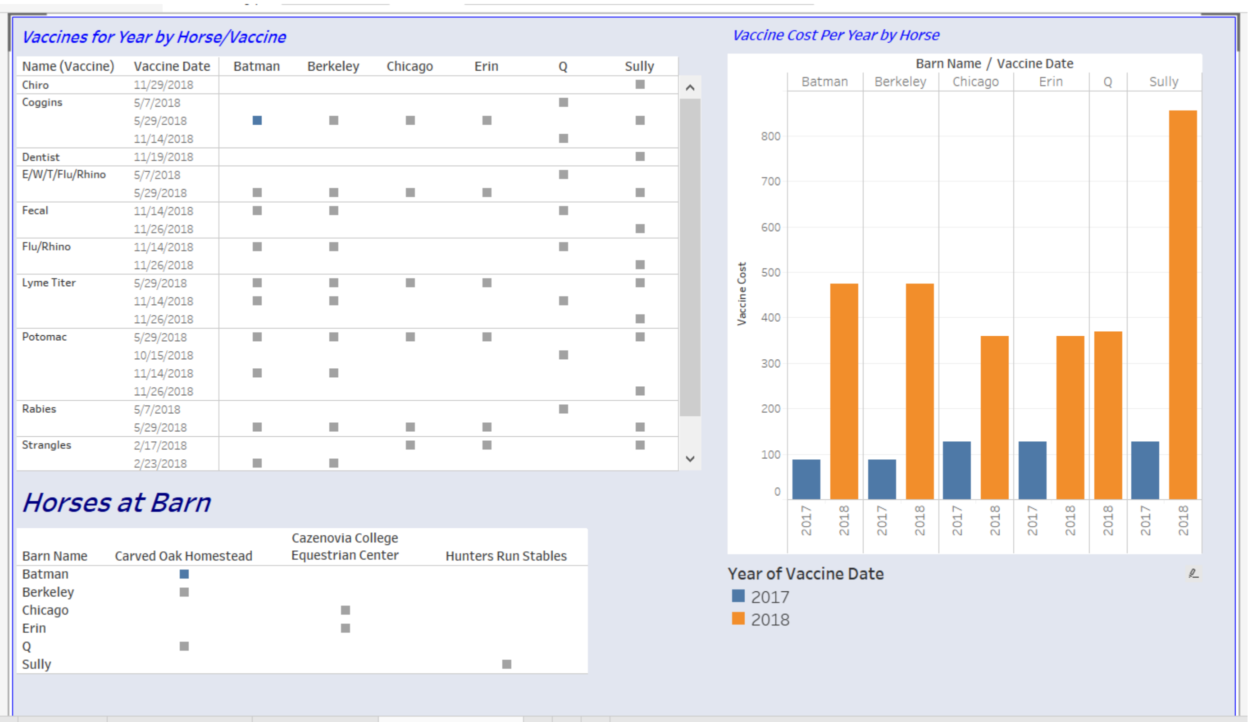


Horses at Barn

Name	Batman Gotham	Berkeley No Fault of Mi..	Chicago Untouchable	Erin Forever Eowyn	Q Dun Q	Sully Secret Decision
	Barn Name / Show Name					
Carved Oak Homestead	■	■			■	
Cazenovia College Equestrian ..			■	■		
Hunters Run Stables						■

Figure 37: Worksheet to Display Horse and their Barn Location by Barn Name

After working with Tableau, I was able to take a few worksheets that I created and put them all together in a dashboard for the end user. The dashboard presented below is a combination of the previously displayed worksheets and an additional one for costs for vaccines by year by horse. This provides the end user a nice view into the information in the database.



Vaccines for Year by Horse/Vaccine

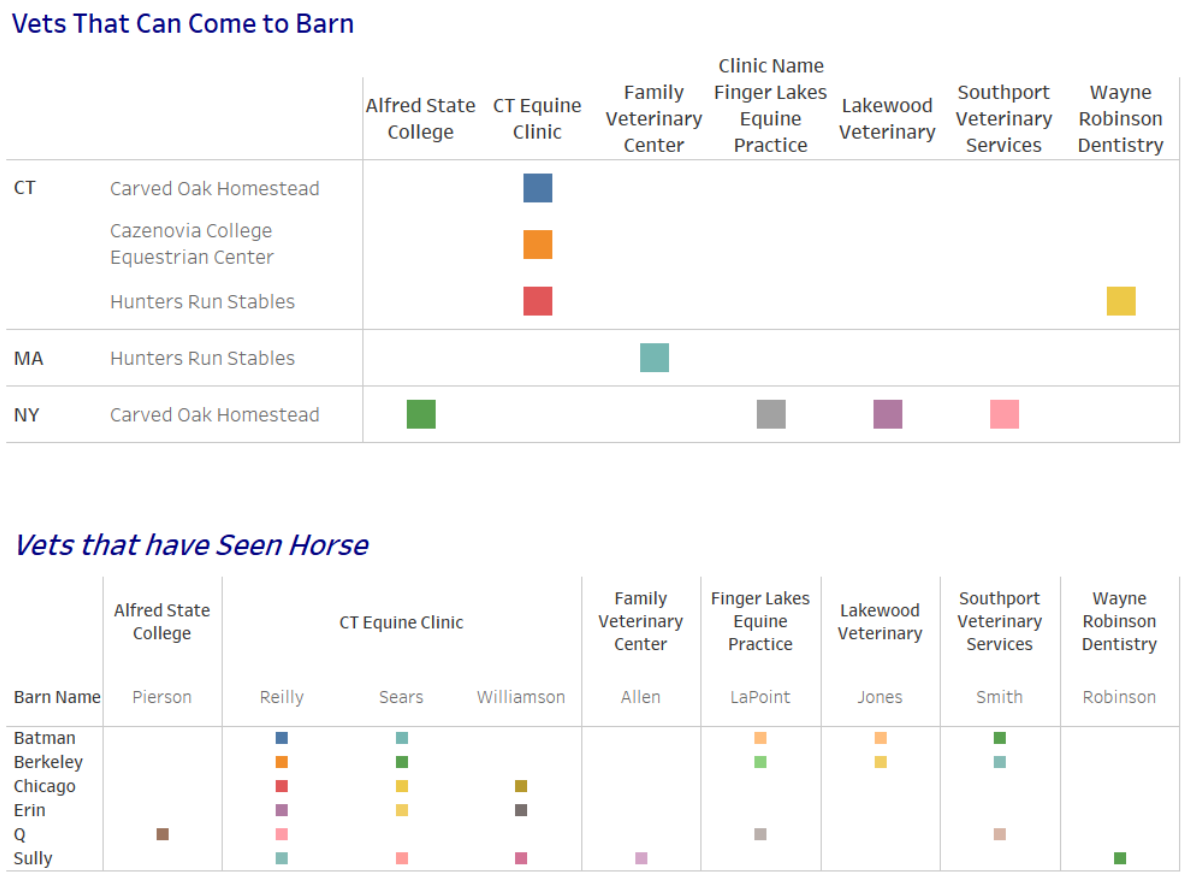
Name (Vaccine)	Vaccine Date	Batman	Berkeley	Chicago	Erin	Q	Sully
Chiro	11/29/2018						■
Coggins	5/7/2018					■	
	5/29/2018	■	■	■	■		■
	11/14/2018					■	
Dentist	11/19/2018						■
E/W/T/Flu/Rhino	5/7/2018					■	
	5/29/2018	■	■	■	■		■
Fecal	11/14/2018	■	■			■	
	11/26/2018						■
Flu/Rhino	11/14/2018	■	■			■	
	11/26/2018						■
Lyme Titer	5/29/2018	■	■	■	■		■
	11/14/2018	■	■			■	
	11/26/2018						■
Potomac	5/29/2018	■	■	■	■		■
	10/15/2018					■	
	11/14/2018	■	■				
	11/26/2018						■
Rabies	5/7/2018					■	
	5/29/2018	■	■	■	■		■
Strangles	2/17/2018			■	■		■
	2/23/2018	■	■				

Horses at Barn

Barn Name	Carved Oak Homestead	Cazenovia College Equestrian Center	Hunters Run Stables
Batman	■		
Berkeley	■		
Chicago		■	
Erin		■	
Q	■		
Sully			■

Figure 38: Dashboard of Various Worksheets

In the following two dashboards, I made more appropriate dashboards around my data questions. The first display more about the Vets, Clinics, Barns and Horses linking data by state to provide further insight – like which vet I can use for a particular horse based on the barn location.



Vets That Can Come to Barn

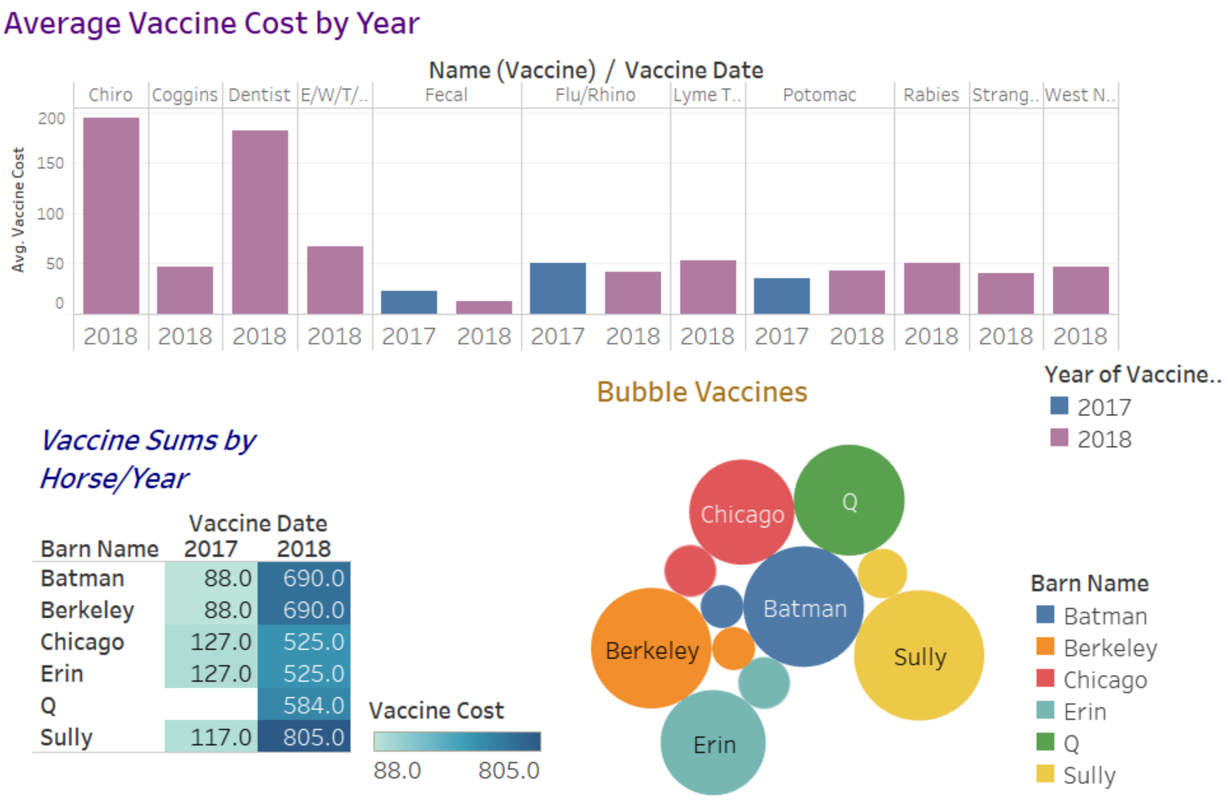
		Clinic Name						
		Alfred State College	CT Equine Clinic	Family Veterinary Center	Finger Lakes Equine Practice	Lakewood Veterinary	Southport Veterinary Services	Wayne Robinson Dentistry
CT	Carved Oak Homestead		■					
	Cazenovia College Equestrian Center		■					
	Hunters Run Stables		■					■
MA	Hunters Run Stables			■				
NY	Carved Oak Homestead	■			■	■	■	

Vets that have Seen Horse

	Alfred State College	CT Equine Clinic			Family Veterinary Center	Finger Lakes Equine Practice	Lakewood Veterinary	Southport Veterinary Services	Wayne Robinson Dentistry
Barn Name	Pierson	Reilly	Sears	Williamson	Allen	LaPoint	Jones	Smith	Robinson
Batman		■	■			■	■	■	
Berkeley		■	■			■	■	■	
Chicago		■	■	■					
Erin		■	■	■					
Q	■	■				■		■	
Sully		■	■	■	■				■

Figure 39: Vet, Clinic, Barn and Horse Dashboard

The next dashboard provides a little more information about average prices per vaccines, costs per horse per year and other vaccine related data that would be of interest.



Vaccine Sums by Horse/Year

	Vaccine Date	
Barn Name	2017	2018
Batman	88.0	690.0
Berkeley	88.0	690.0
Chicago	127.0	525.0
Erin	127.0	525.0
Q		584.0
Sully	117.0	805.0

Figure 40: Vaccine Dashboard

Reflection

I found the overall project very informative and it also provided me with additional insight into this data. I am very familiar with this data and how I need to use the information and found that I had a bit of “scope creep” wanting to add more to the data as it is a real-world problem for me.

At the last minute, I decided to add in a few “procedures,” but to lump them into the *Vaccine* and *HorseVaccine* tables. That works fine, but I think I would have used different table and column names so that my model so that I could delineate the information more clearly. For example, dentistry and chiropractic work skew the cost information as they are high ticket items and not traditional vaccines. not listed as a specific vaccine as it takes away from the information.

In working with data like this, I found that knowing what questions you need to ask help you define how the model must look. It is like working backwards. Understanding how the information and data needs to be used also helps to shape the models in a way (like I mentioned the chiropractor and dentist being lumped into vaccines, which is not completely appropriate – which would have changed my approach in the long run had I thought to include this data up front).