# A. Title Page

Lewis University  
CPSC 50900: Database Systems   
Spring 2023 Term Project

**VETERINARY HOSPITAL MANAGEMENT SYSTEM**

Sai Vamshi Kampasati, [SaiVamsiKampasati@lewisu.edu](mailto:SaiVamsiKampasati@lewisu.edu)

Work products stored in the Github repository <https://github.com/vamshi219/2023-Spring-DBMS>

Table of Contents

[Initial Proposal 2](#_Toc85814569)

[Data Sources 2](#_Toc85814570)

[Data Storage Alternatives 2](#_Toc85814571)

[Relational Database Design Process 3](#_Toc85814572)

[Relational Database Design 3](#_Toc85814573)

[Data Definition Language (DDL) Scripts 3](#_Toc85814574)

[Data Manipulation Language Scripts 4](#_Toc85814575)

[Indexes 5](#_Toc85814576)

[Views 5](#_Toc85814577)

[Triggers 5](#_Toc85814578)

[Transactions 5](#_Toc85814579)

[Database Security 6](#_Toc85814580)

[Locking and Concurrent Access 6](#_Toc85814581)

[Backing Up Your Database 6](#_Toc85814582)

[Python Programming 7](#_Toc85814583)

[PHP Programming 7](#_Toc85814584)

[Suggested Future Work 8](#_Toc85814585)

[Activity Log 8](#_Toc85814586)

# B. Initial Proposal

*Description: You will describe the data you aim to store. What data will be storing? Why are you interested in this data? Why is it important? Where will the data come from? Who will use this data? What kind of application do you plan to build with it?*

*Rubric: Your response to each of these six questions will be graded out of 3 points.*

* *3 points: clear, complete descriptions that convey the importance and meaning of your data*
* *2 points: mostly clear descriptions, although some additional data would have helped in some sections*
* *1 point: necessary details are lacking in many of your responses.*

*You will also earn 2 additional points for coming up with a descriptive title for your project.*

*Total points possible: 20*

Importance of data: The primary aim of this database is to construct a system for managing a veterinary hospital. It is crucial to organize information regarding appointments, animals, doctors, bills, and other pertinent details. The aim of this software is to minimize the manual effort involved in maintaining records for animals, doctors, bills, and other relevant data.

The objective of the system is to develop a veterinarian workflow management tool. A streamlined and easy-to-use application was necessary to monitor patient records, appointment scheduling, treatment procedures, and medications. Every veterinarian requires an efficient application to oversee their clients, patients, medication administration, schedules, and financial transactions.

The data records have been collected from the veterinary hospital located in our locality.

# C. Data Sources

*Description:* *Gather your data in text files. The text files may be csv, tab-delimited, xml, json, or some other custom format. Not all the files need be of the same type. Identify what each file contains by indicating where it came from, explaining in detail how it structured, and describing how you will reorganize the data into a relational database. Post your data files to your GitHub repository, and provide samples of the data in your Word doc.*

*Rubric: Your work will be graded as follows:*

* *5 points: you gathered multiple data files that contain the data that will populate your databases. If you do not use multiple data files, you will not receive credit.*
* *5 points: you described the contents of the data files in detail, including referencing their origin and explaining how they were structured.*
* *3 points: you identify which fields you plan to include in your database, including their data types and any constraints you expect to impose on the data or steps you'll have to take to clean up the data.*
* *2 points: you post the data files to your GitHub account and make it possible for me to see them.*

*Total points possible: 15*

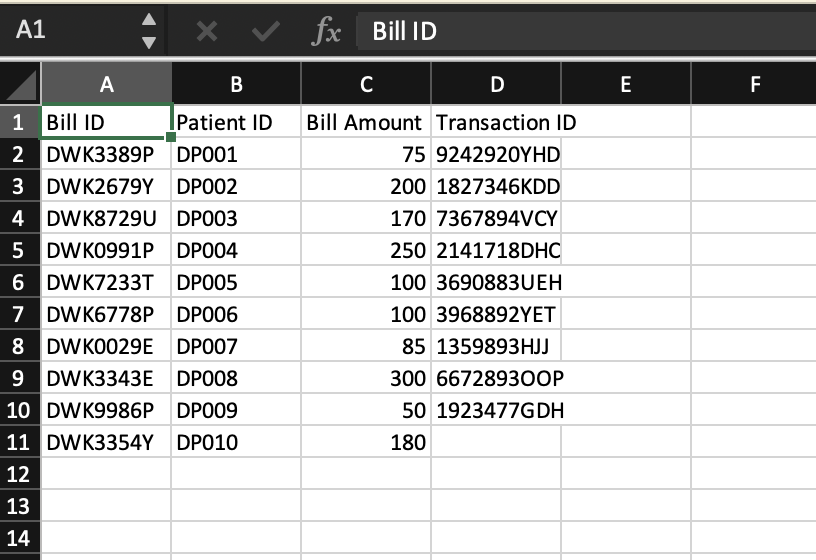


Graphical user interface, application, table, Excel

Description automatically generated

Graphical user interface, application, table, Excel

Description automatically generated



* Details : they include common details about the doctors, hospitals, the animals that come for treatment, bills, etc.
* Source : this data was completely gathered from the local veterinary hospitals. It was written in a single container and later was segregated accordingly.
* Initially, we sorted the data by segregating it into separate tables based on their corresponding categories, such as patients, doctors, and appointments. Subsequently, we converted the information into CSV, XML, and JSON formats, which can be utilized on multiple platforms.
* We performed field-specific data cleansing and incorporated new fields, such as appointments and transactions, which were previously absent. These measures were implemented to ensure that the entire database is structured appropriately.

# D. Alternative Ways to Store the Data

*Description: We will study alternatives to storing data in a relational database. Some of the alternatives come from several decades ago, including the hierarchical and network models. Some are newer options, such as NoSQL databases that use JSON or some other encoding. Describe in detail how to store the data using two alternatives to relational databases. Be sure to describe how you would implement the alternatives and the advantages and disadvantages of each.*

*Rubric: Your work will be graded as follows*

* *5 points for clearly describing how your data could be stored using one alternative to relational databases and what the advantages and disadvantages of that approach would be.*
* *5 points for clearly describing how your data could be stored using another alternative to relational databases and what the advantages and disadvantages of that approach would be.*

*Total points possible: 10*

**1st alternative**

My first alternative would be the NoSQL database, MongoDB. It is a document-oriented database i.e., it stores data in documents. A document-based database is a type of non-relational database that stores data in documents, rather than in traditional rows and columns (tables). These documents are stored in formats such as JSON, BSON, or XML. By using a document-based database, data objects in applications can be stored and retrieved with less translation required, as the data is already in a format close to that used in applications. Index values are assigned to the documents, allowing faster querying of specific elements.

Documents are stored in collections, which are groups of documents that share similar contents. Unlike traditional databases, not all documents in a collection require the same schema. This flexibility is a key feature of document databases.

Other key features include the ability to easily create and maintain documents, as minimal maintenance is required once a document is created. Additionally, there are no foreign keys in document databases, as there is no dynamic relationship between two documents. This allows documents to be independent of one another, and reduces the need for complex relational structures.

**2nd alternative**

For the 2nd alternative, I would like to use a key-value store as it is a type of non-relational database that is often considered the simplest form of a NoSQL database. In this database, each data element is stored in the form of a key-value pair, which can be retrieved using a unique key assigned to it. The values can be simple data types, such as strings and numbers, or complex objects.

In essence, a key-value store can be seen as a relational database with only two columns - the key and the value.

Key features of a key-value store include simplicity, scalability, and speed. Because of their simple structure, key-value stores are relatively easy to use and manage. They are also highly scalable, allowing for large amounts of data to be stored and accessed quickly. Finally, key-value stores are known for their speed and performance, making them a popular choice for high-traffic applications that require fast data retrieval.

# E. Relational Database Design Process

*Description: Consider the list of fields you identified in part c. Identify functional dependencies that exist among them. For each functional dependency, identify the determinants and the fields they determine. This becomes the basis for identifying your entity sets, which then become your tables. Give each entity set or table you identify in this way a unique and clear name, making sure that the names you use are singular nouns. Then list the relationships that exist among the various entity sets. For each relationship, identify its connectivity (one-to-one, one-to-many, many-to-many) and participation (optional or mandatory). Finally, make sure that none of the attributes you've assigned to each entity set are multi-valued. If they are, take the steps needed to break them down.*

*Rubric: Your work will be graded as follows:*

* *8 points for identifying all the functional dependencies, including determinants and the columns whose values they determine.*
* *2 points for naming the entity sets that make up your data with clear, easy-to-understand names.*
* *6 points for identify the relationships among the entity sets and identifying connectivity and participation for each.*
* *2 points for breaking down multi-valued attributes.*

*Total points possible: 18*

ENTER YOUR RELATIONAL DATABASE DESIGN DESCRIPTION HERE. INCLUDE SOURCE CODE AND SCREEN SHOTS.

Functional dependencies are Doctor-id->doctor title, Appointment-id->appointment, animal id->animal name, treatment id-> treatment name, transaction id->bill and the determinants include patient id, treatment id, doctor id, transaction id, Appointment id.

ENTITY SETS

* Animal
* Doctor
* Appointment
* Transaction
* Treatment
* Bill

RELATIONSHIPS

* One-to-many : animal – appointment, doctor – appointment, transactions – bills
* One-to-one : transactions – appointment, treatment – transactions

# F. Relational Database Design

*Description: This is where you will complete your database design. For each of the entity sets you identified in the preceding section, analyze them to make sure they pass 2nd, 3rd, 4th, and Boyce-Codd Normal Form. If they do not, introduce additional entity sets or key changes to make sure that they do. Then, add foreign keys to connect entity sets that are related. For many-to-many relationships, introduce bridge entity sets to convert them into two one-to-many relationships. Also, consider whether you should introduce surrogate keys to create a more efficient primary key for some of your entity sets. Finally, diagram your design in Vertabelo. Make sure your ER diagram correctly shows all entity sets, their primary and foreign keys, the data types for each attribute, and the connectivity and participation characteristics of each entity set. Your final Vertabelo design should be something you could actually implement in a relational database management system.*

*Rubric: Your work will be graded as follows:*

* *4 points for the normalization analysis of your entity sets.*
* *3 points for introducing bridge entity sets.*
* *3 points for choosing foreign keys and perhaps more efficient surrogate keys*
* *10 points for correctly depicting your physical database model in Vertabello*

*Total points possible: 20*

ENTER YOUR RELATIONAL DATABASE DESIGN HERE

Here the bridge set is transactions which includes transaction id, bill amount, animal id, treatment id and doctor id.

Foreign keys are animal id, treatment id and doctor id and transaction id.

Diagram

Description automatically generated

# G. Data Definition Language (DDL) Scripts

*Description: Use Vertabello to generate a script of SQL commands that build the database and its table structures. Write scripts or build Excel spreadsheets that take your data files and generate scripts of SQL insert statements from them. Use the MySQL source command to run the various scripts needed to build and populate the database in MySQL. Include the source code and / or Excel spreadsheets you use to manipulate and populate the data. Make sure all your tables have at least three records in them and that you've linked the tables through their foreign keys.*

*Rubric: Your work will be grades as follows:*

* *Database and table creation statements from Vertabelo saved as an sql script file: 3 points*
* *Scripts you write or Excel spreadsheets you create to generate SQL commands for populating the tables, uploaded to GitHub: 8 points*
* *Descriptions of the scripts and Excel spreadsheets you wrote along with code excerpts included in the Word document: 5 points*
* *Screenshots of your successful attempts to use the MySQL source command to populate each table with at least three records: 4 points*

*Total points possible: 20*

ENTER YOUR DDL WORK HERE

-- Created by Vertabelo (http://vertabelo.com)

-- Last modification date: 2023-04-11 11:27:29.166

create database Veterinary;

use Veterinary;

-- tables

-- Table: Animal

CREATE TABLE Animal (

animal\_id varchar(50) NOT NULL,

animal\_type varchar(50) NOT NULL,

owner\_name varchar(25) NOT NULL ,

phn\_number int NOT NULL ,

age int NOT NULL ,

Address varchar(30) NOT NULL ,

CONSTRAINT patient\_pk PRIMARY KEY (animal\_id)

);

-- Table: Appointment

CREATE TABLE Appointment (

Appointment\_ID int NOT NULL,

Appointment\_Date varchar(20) NOT NULL,

Animal\_animal\_id varchar(50) NOT NULL,

Doctor\_Doctor\_Id varchar(50) NOT NULL,

Transactions\_Transaction\_id varchar(50) NOT NULL,

CONSTRAINT Appointment\_pk PRIMARY KEY (Appointment\_ID)

);

-- Table: Bills

CREATE TABLE Bills (

Bill\_id varchar(50) NOT NULL,

Bill\_amount int NOT NULL,

Transactions\_Transaction\_id varchar(50) NOT NULL,

CONSTRAINT Bills\_pk PRIMARY KEY (Bill\_id)

);

-- Table: Doctor

CREATE TABLE Doctor (

Doctor\_Id varchar(50) NOT NULL,

Doctor\_name varchar(50) NOT NULL,

doctor\_number int NOT NULL,

Specialization varchar(50) NOT NULL,

CONSTRAINT Doctor\_pk PRIMARY KEY (Doctor\_Id)

);

-- Table: Transactions

CREATE TABLE Transactions (

Transaction\_id varchar(50) NOT NULL,

Bill\_id varchar(50) NOT NULL,

Treatement\_treatment\_id varchar(50) NOT NULL,

CONSTRAINT Transactions\_pk PRIMARY KEY (Transaction\_id)

);

-- Table: Treatement

CREATE TABLE Treatement (

treatment\_id varchar(50) NOT NULL,

treatement\_name varchar(255) NOT NULL,

CONSTRAINT Treatement\_pk PRIMARY KEY (treatment\_id)

);

-- foreign keys

-- Reference: Appointment\_Animal (table: Appointment)

ALTER TABLE Appointment ADD CONSTRAINT Appointment\_Animal FOREIGN KEY Appointment\_Animal (Animal\_animal\_id)

REFERENCES Animal (animal\_id);

-- Reference: Appointment\_Doctor (table: Appointment)

ALTER TABLE Appointment ADD CONSTRAINT Appointment\_Doctor FOREIGN KEY Appointment\_Doctor (Doctor\_Doctor\_Id)

REFERENCES Doctor (Doctor\_Id);

-- Reference: Appointment\_Transactions (table: Appointment)

ALTER TABLE Appointment ADD CONSTRAINT Appointment\_Transactions FOREIGN KEY Appointment\_Transactions (Transactions\_Transaction\_id)

REFERENCES Transactions (Transaction\_id);

-- Reference: Bills\_Transactions (table: Bills)

ALTER TABLE Bills ADD CONSTRAINT Bills\_Transactions FOREIGN KEY Bills\_Transactions (Transactions\_Transaction\_id)

REFERENCES Transactions (Transaction\_id);

-- Reference: Transactions\_Treatement (table: Transactions)

ALTER TABLE Transactions ADD CONSTRAINT Transactions\_Treatement FOREIGN KEY Transactions\_Treatement (Treatement\_treatment\_id)

REFERENCES Treatement (treatment\_id);

-- End of file.

use Veterinary;

insert into Doctor values('DD543','Prathusha',9834598,'Anesthesia and analgesia');

insert into Doctor values('DD098','Dexter',682365,'Animal welfare');

insert into Doctor values('DD102','Micheal',234739,'Behavioral medicine');

insert into Appointment values(101, '21/02/2021', 'DP003', 'DD543', 'TS0023');

insert into Appointment values(102, '05/10/2021', 'DP004', 'DD098', 'TS0024');

insert into Appointment values(103, '15/09/2020', 'DP004', 'DD098', 'TS0109');

insert into Appointment values(104, '04/04/2019', 'DP001', 'DD102', 'TS0025');

insert into animal values('DP001','Dog','Vikram',84567643,30,'New jersey');

insert into animal values('DP002','Dog','Stephen',83251890,27,'New jersey');

insert into animal values('DP003','Cow','Ronald',546439335,22,'Newark');

insert into animal values('DP004','Cat','Suzy',437999217,32,'New jersey');

insert into animal values('DP005','buffalo','Rose',549278460,24,'Trenton');

insert into Bills values('DWK3389P',75,'TS0023');

insert into Bills values('DWK2679Y',200,'TS0024');

insert into Bills values('DWK8729U',170,'7367894VCY');

insert into Bills values('DWK0991P',250,'2141718DHC');

insert into Bills values('DWK7233T',100,'TS0025');

insert into Bills values('DWK6778P',100,'3968892YET');

insert into Bills values('DWK0029E',85,'TS0109');

insert into Transactions values('TS0023','DWK3389P', 'TT1001');

insert into Transactions values('TS0024','DWK2679Y', 'TT1002');

insert into Transactions values('TS0025','DWK7233T', 'TT1004');

insert into Transactions values('TS0109','DWK0029E', 'TT1005');

insert into Treatement values('TT1001','Antiparasitics');

insert into Treatement values('TT1002','Antifungals');

insert into Treatement values('TT1003','Pain Relievers');

insert into Treatement values('TT1004','Vaccination');

insert into Treatement values('TT1005','Antibiotics');

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# H. Data Manipulation Language Scripts

*Description: Write the SQL commands for twelve queries. Two queries should be insert statements, two should update statements, one should be a delete statement, one should be a simple select statement that selects a subset of the rows and columns from one table, two should be a select statements that select data from a joining of two tables, two should use summary functions to generate statistics about the data, one should be a multi-table query, and one should be another query of your choice. Show the queries and screenshots of the results in your Word document, and save your queries in a commented sql script to GitHub.*

*Rubric: Your work will be graded as follows:*

* *1 point each for the two insert statements*
* *1 point each for the two update statements*
* *1 point for the delete statement*
* *1 point for the simple select statement*
* *2 points each for the 2 join statements*
* *2 points each for the two that use summary statements*
* *2 points for the multi-table query*
* *2 points for the query of your choice.*
* *6 points for showing the query and a screenshot of the corresponding result set back-to-back for each of these queries in your Word document.*

*Total points possible: 24*

ENTER DML WORK HERE

insert into Doctor values('DD111','Natasha',8745632,'Orthodontics');

insert into animal values('DP010', 'dog','Clair',4782964,33,'los angeles');

Table

Description automatically generated

Table

Description automatically generated

update animal set age=10 where animal\_id='DP010';

update Doctor set doctor\_number=6783434 where doctor\_Id='DD111';

Table

Description automatically generated

Graphical user interface, application, table, Teams

Description automatically generated

delete from Doctor where doctor\_id='DD111';

Table

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, application

Description automatically generated

Graphical user interface, table

Description automatically generated with medium confidence

Graphical user interface, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text

Description automatically generated with medium confidence

# I. Indexes

*Description: Improve the performance of your design by adding indexes to various tables. Show the SQL needed to add the indexes. Explain why you chose the ones you added. Explain how you would demonstrate the impact the indexes had on the performance of various queries.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly defining at least three indexes and explaining why you chose them.*
* *3 points for showing the sql needed to generate the indexes*
* *2 points for explaining how you would demonstrate the performance improvement afforded by the indexes.*

*Total points possible: 8*

ENTER YOUR INDEX WORK HERE

# J. Views

*Description: Add two views to your database to provide easy access to combinations of data from multiple tables.*

*Rubric: Your work will be graded as follows:*

* *2 points for including the SQL for generating the two views in your Word document*
* *2 points for including screenshots for the data contained in each view in your Word document*
* *2 points for explaining why each view is a valuable addition to your database*

*Total points possible: 6*

ENTER YOUR WORK WITH VIEWS HERE

# K. Triggers

*Description: Add a trigger to a table so that data will be updated when a certain event occurs*

*Rubric: Your work will be graded as follows:*

* *2 points for including the SQL for the trigger in your Word document*
* *2 points for clearly explaining the purpose of the trigger*
* *2 points for a screenshot and explanation that shows the trigger in action.*

*Total points possible: 6*

ENTER YOUR WORK WITH TRIGGERS HERE

# L. Transactions

*Description: Demonstrate that you know how to define and use a transaction. Why are transactions important for ensuring ACID behavior?*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly explaining the importance of transactions to ensuring ACID behavior*
* *3 points for including a screenshot and accompanying explanation of a MySQL transaction.*

*Total points possible: 6*

ENTER YOUR WORK WITH TRANSACTIONS HERE

# M. Database Security

*Description: Identify the different kinds of users who will use your database. Write GRANT statements to define the privileges for these different kinds of users.*

*Rubric: Your work will be graded as follows:*

* *6 points for clearly identifying and describing the various kinds of users who will use the databases and identifying and justifying what privileges each should have.*
* *4 points for writing GRANT statements that assign privileges to these different kinds of users.*
* *4 points for demonstrating with screenshots that your GRANT statements do distinguish among different kinds of users in regard to what they can do with the database.*

*Total points possible: 12*

ENTER YOUR WORK WITH DATABASE SECURITY HERE

# N. Locking and Concurrent Access

*Description: Explain the purpose of locking tables and show how to do that to prevent inconsistencies that may arise in your data when concurrent transactions take place.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly explaining an example that shows why you should lock tables to prevent inconsistencies.*
* *3 points for providing a screenshot and accompanying explanation of locking tables.*

*Total points possible: 5*

ENTER YOUR WORK WITH LOCKING AND CONCURRENT ACCESS HERE

# O. Backing Up Your Database

*Description: How you will back up your database. What commands will you issue? How frequently will the commands run? How can they be automated? Where will the backups be stored?*

*Rubric: Your work will be graded as follows:*

* *6 points for clearly explaining and justifying your database backup strategy, including the frequency with which you will back up the database, how you will automate backups, where you will store them, and how you will secure them. You will earn three points for addressing each factor (frequency, location, automation, and security)*
* *3 points for providing a screenshot of the command you would issue to back up the database and for including a portion of the resulting file.*

*Total points possible: 8*

ENTER YOUR WORK ON DATABASE BACKUPS HERE

# P. Programming

*Description: Write a Python, Java, or PHP program that generates a report that contains a subset of the data from your database. Include the code for your Python program in your Word document, and also post the program to your GitHub repository.*

*Rubric: Your work will be graded as follows:*

* *10 points for writing a Python script (and including its code in the Word doc) that will pull data from a database and store it to a text file and present it to the screen. Your code must have comments in it that explain how it works. You will be awarded 3 points for successfully connecting to the database, 3 points for successfully querying it, and 4 points for presenting the data to the screen and to a file. Internal comments count for 2 points.*
* *2 points for posting the code to GitHub*
* *6 points for showing a screenshot of your running the script and showing the results it produces on the screen.*

*Total points possible: 18*

ENTER YOUR PYTHON, PHP, or JAVA DATABASE PROGRAMMING WORK HERE

# R. Suggested Future Work

*Description: Describe the limitations of your current database and explain how you or someone else could improve the design to address these shortcomings. Also describe how you might take advantage of leverage cloud services to increase the performance and availability of your database. Finally, explain the advantages and disadvantages of storing your data in a NoSQL format instead.*

*Rubric: Your work will be graded as follows:*

* *3 points for clearly describing the limitations of your databases*
* *3 points for explaining how you would address these shortcomings*
* *3 points for explaining how you might migrate the database to the cloud and describing what advantages you might gain from doing that.*
* *3 points for explaining the advantages and disadvantages of storing your data in a document-based NoSQL format instead.*

*Total points possible: 12*

ENTER YOUR SUGGESTED FUTURE WORK IDEAS HERE

# S. Activity Log

*Description: As an appendix, the team will keep a daily diary or log of their activity. What did you or your team study in this class each day? What did you learn? What did you accomplish or build or design? You don't have to enter something every day, but there should be at least three entries each week. Since we have eight weeks, that means you should make 3 posts to the Activity Log each week, for a total of at least 24 posts. Each post will be worth 1 point.*

*If you are working as part of a team, make sure you clearly identify which team member worked on which tasks. The Activity Log should help me figure out how each team member contributed to the project. If I cannot discern who worked on what aspects of the project from the activity log, no points will be awarded for it.*

*Total points possible: 24*

Week 1

* Researched various topics, thought of what topic to choose for the project.
* Decided on the topic and started doing the initial proposal. Answered first 2 questions in initial proposal.
* Answered rest of the questions.

Week 2

* Started working on the datasets part. Researched a few datasets online regarding veterinary clinics.
* Went to a local veterinary hospital. Asked a few questions. Gathered some data. Combined both the online and gathered data into csv files.
* Completed doing the data sources part.
* Saw what some alternatives to the traditional RDBMS were. Picked a two, that were convenient for my current project.

Week 3

* Worked on what data to consider and what functional dependencies are there.
* Created my entities and completed the relational database design process.
* Created the design in vertabelo and download the sql file from it and executed it in the mysql workbench.

Week 4

* Started inserting data into the tables.
* Worked on the DML queries and completed it.