Animal Shelter Model

Shum Chang
Alice
Daniel
Gwendolyn
Lily
CIS 3400 EMWA
Group 8

Milestone 1 Proposal

We are an animal shelter start-up that has opened within the last couple of years. At first, we used handwritten logs to keep count of animals in the shelter, materials (food, toys, medicine), and volunteer timesheets. We aim to replace this with a digital database system in order to speed up the work cycle, allowing us to provide better care to these lovely animals.

The information we would need to complete this database are employees, volunteers, animals, location, food, supplies, and adoption information. Each of these sections can be further expressed in attributes.

Entities	Attributes
Personnel	PersonnelID, FirstName, MiddleName, LastName, Birthday StreetName, City State, Zipcode, PhoneNumber, Email, Position, Paytype (Hourly or Volunteer)
Task	TaskID, TaskName, TaskDescription, Priority, Status, StartDate, DueDate, CompletionDate, AnimalsInvolved
Animal Shelter	AnimalShelterID, Name, Email, PhoneNumber, StreetName, City, State, Zipcode
Supplies	SuppliesID, Type, ItemName, Quantity
Pet	PetID, Species, Breed, Name, Birthday, Gender, Color, Height, Weight, Temperament, IntakeDate, IntakeReason, HealthStatus
AdoptionApplication	AdoptionApplicationID, AdoptionStatus (Started, Pending, Completed), AdoptionFileDate, DateAdopted
MedicalRecord	MedicalRecordID, MedicalHistory, MedicalRecordDescription, MedicalRecordDate, Allergies, MedicalInsurance
Patron	PatronID, FirstName, MiddleName, LastName, StreetName, City, State, Zipcode, PhoneNumber, Email, Donation

Distribution of duties

Alice Gonzalez | Project Manager
Shum Chang Wu Chen | Application Developer
Lily Qiu | Application Developer
Gwendolyn Li | Application Developer
Daniel Castillo | Application Developer

Milestone 2 ER Diagram

Relationship Sentences

One animal shelter may(0) be stocked with many(*) supplies.

One supply must(1) be stored at one (1) animal shelter.

One animal shelter may(0) be staffed by many(*) personnel.

One personnel must(1) work for one(1) animal shelter.

One personnel may (0) work on many(*) task.

One task may (0) be done by many(*) personnel

One animal shelter may(0) have many(*) pets.

One pet must(1) be cared for at one(1) animal shelter.

One pet may(0) receive many(*) adoption applications.

One adoption application must(1) be started for one(1) pet.

One adoption application must(1) be assigned to one(1) patron.

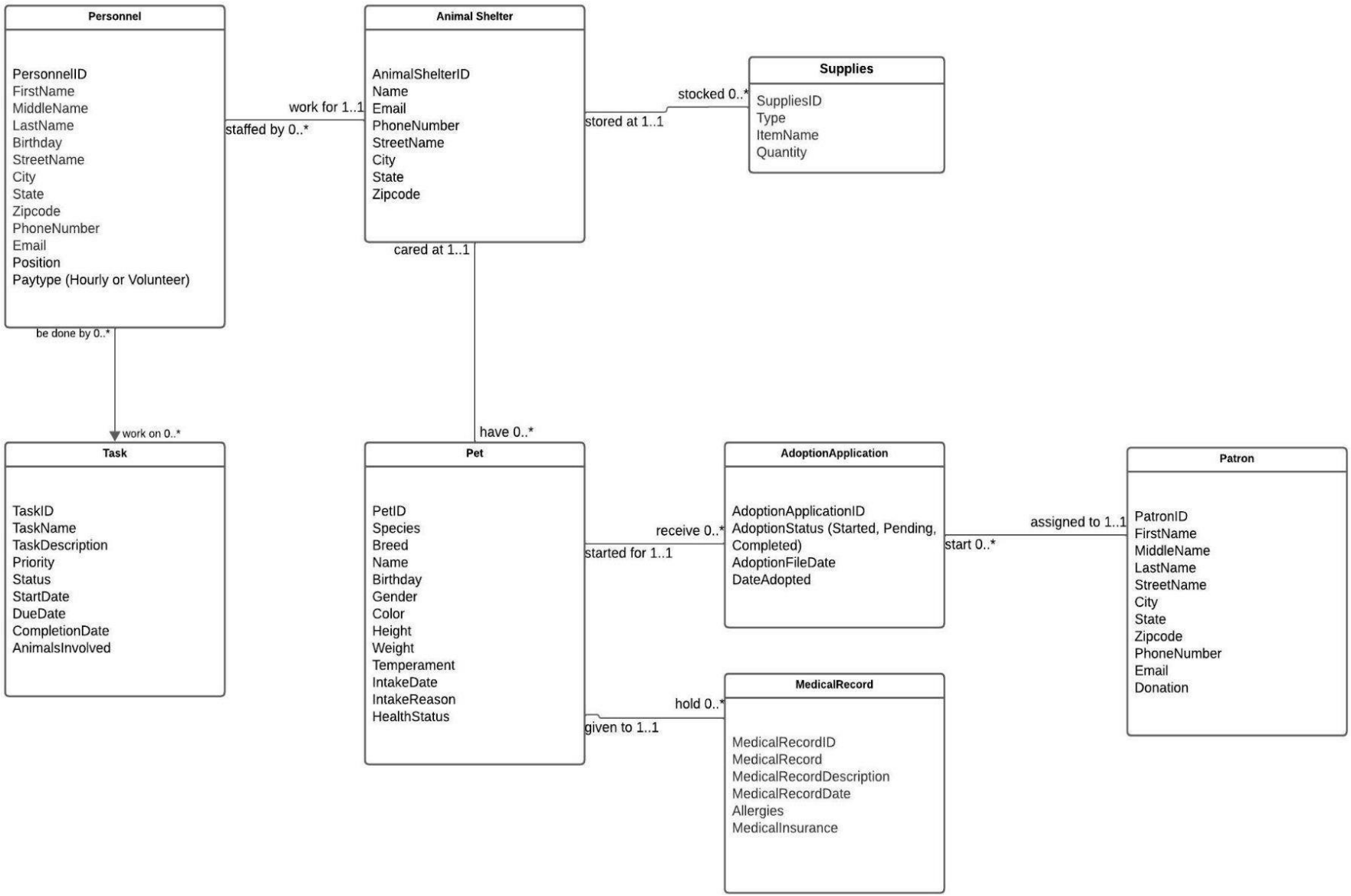
One patron may(0) start many (*) adoption application.

One pet may(0) hold many(*) medical record.

One medical record must(1) be given to one(1) pet.

Model:Group Project Entity Relationship

Created on: 9/18/23 7:26PM
Last Edit: 10/23/23 8:21PM
Edited By: Group 8



Milestone 3 Relation Model and Normalization

Initial Set of Relations

Animal Shelter (AnimalShelterID(KEY), Name, Email, PhoneNumber, StreetName, City, State, Zipcode)

ANIMAL SHELTER							
AnimalShelterID (KEY)	Name	Email	PhoneNumber	StreetName	City	State	Zipcode
A100	Animal Haven	animalhaven@example.com	(543) 777 - 2349	432 Oaks Ave.	NY	NY	11433

Animal Shelter

Key: AnimalShelterID

FD1: AnimalShelterID → Name, Email, Phone Number, StreetName, City, State, Zipcode

FD2: ZipCode->City,State

1NF: Yes Animal Shelter reaches 1NF as it meets the definition of a relation.

2NF: Yes Animal Shelter reaches 2NF because it has no partial key dependencies.

3NF: No Animal Shelter does not reach 3NF because FD2 is a transitive dependency but for this project we are electing to leave this relation in its denormalized form.

Final Set of Relations:

Animal Shelter: (AnimalShelterID, Name, Email, Phone Number, StreetName, City, State, Zipcode)

Key: AnimalShelterID

FD1: AnimalShelterID → Name, Email, Phone Number, StreetName, City, State, Zipcode

FD2: ZipCode->City,State

Personnel (PersonnelID (KEY), FirstName, MiddleName, LastName, Birthday, StreetName, City, State, Zipcode, PhoneNumber, Email, Position, Paytype (Hourly or Volunteer), AnimalShelterID(FK))

Personnel													
PersonnelID (KEY)	FirstName	MiddleName	LastName	Birthday	StreetName	City	State	Zipcode	PhoneNumber	Email	Position	Paytype	AnimalShelterID
E100	Joy	August	White	1/3/99	9391 James St	NY	NY	11419	(555) 555-1234	JoyAugust123@example.com	Manager	Hourly	A100
E102	Mark	Calvin	Joe	3/4/00	7 Foster Ave.	NY	NY	10977	(583) 225-8834	mark.calvin567@example.com	Nurse	Hourly	A100
E103	Sophia	Jade	Jones	4/5/01	9998 Central Drive	NY	NY	10033	(889) 232-1924	sophia.jade@emailgen123.com	Vet	Hourly	A100
E104	Seph	Ryder	Garcia	6/12/02	190 Goldfield Street	NY	NY	11236	(999) 362-8715	seph.ryder@example.com	Graphic Designer	Volunteer	A100
E105	Tyler	Brooklyn	Miller	9/12/03	451 N. Hawthorne Ave	NY	NY	11201	(295) 920-10387	tyler.brooklyn@example.com	Receptionist	Hourly	A100
E106	Troy	Mauve	Davis	7/14/02	7270 Glendale Rd	NY	NY	11228	(382) 183-1030	troy.mauve@example.com	Receptionist	Hourly	A100

Personnel

Key: PersonnelID

FD1: PersonnelID→ FirstName, MiddleName, LastName, Birthday, StreetName, City, State, Zipcode, PhoneNumber, Email, Position, Paytype (Hourly or Volunteer), AnimalShelterID(FK)
FD2: ZipCode->City,State

1NF: Yes Personnel Relation reaches 1NF as it meets the definition of a relation.
2NF: Yes Personnel Relation reaches 2NF because it has no partial key dependencies.
3NF: No Personnel Relation does not reach 3NF because FD2 is a transitive dependency but for this project we are electing to leave this relation in its denormalized form.

Final Set of Relations:
Personnel: (PersonnelID→ FirstName, MiddleName, LastName, Birthday, StreetName, City, State, Zipcode, PhoneNumber, Email, Position, Paytype (Hourly or Volunteer), AnimalShelterID(FK))
Key: PersonnelID
FK: AnimalShelterID
FD1: PersonnelID→ FirstName, MiddleName, LastName, Birthday, StreetName, City, State, Zipcode, PhoneNumber, Email, Position, Paytype (Hourly or Volunteer), AnimalShelterID(FK)
FD2: ZipCode->City,State

Task (TaskID (KEY), TaskName, TaskDescription, Priority, Status, StartDate, DueDate, CompletionDate, AnimalsInvolved)

TASK								
TaskID (KEY)	TaskName	Task Description	Priority	Status	StartDate	DueDate	CompletionDate	AnimalsInvolved
T100	Dogwash	Washing	Low	Working	1/2/23	1/10/23	N/A	Dog
T1002	Catwash	Washing	Low	Working	1/3/23	1/11/23	N/A	Cat
T1003	Checkup	Check	Medium	Working	1/4/23	1/12/23	N/A	Cat
T1004	Walk	Walk	Low	Done	1/5/23	1/13/23	1/6/23	Dog
T1005	Vaccine	Vaccination	High	Done	1/6/23	1/14/23	1/7/23	Rabit
T1006	Neuter	Neuter	High	Done	1/7/23	1/15/23	1/8/23	Cat

Task
Key: TaskID
FD1: TaskID → TaskName, TaskDescription, Priority, Status, StartDate, DueDate, CompletionDate, AnimalsInvolved

1NF: Yes, Task Relation reaches 1NF as it meets the definition of a relation.
2NF: Yes, Task Relation reaches 2NF because it has no partial key dependencies.
3NF: Yes, Task Relation reaches 3NF because it has no transitive dependencies.

Final Set of Relations:

Task (TaskID (KEY), TaskName, TaskDescription, Priority, Status, StartDate, DueDate, CompletionDate, AnimalsInvolved)
Key: TaskID
FD1: TaskID → TaskName, TaskDescription, Priority, Status, StartDate, DueDate, CompletionDate, AnimalsInvolved

---

Supplies (SuppliesID(KEY), Type, ItemName, Quantity, AnimalShelterID (FK))

SUPPLIES				
SuppliesID(KEY)	Type	ItemName	Quantity	AnimalShelterID (FK)
S100	Leash	Dog Leash	10	A100
S102	Bowl	Dog Bowl	20	A100
S103	Food	Cat Food	5	A100
S104	Bed	Cat Bed	3	A100
S105	Sweater	Cat Sweater	5	A100
S106	Toy	Cat toy	30	A100

Supplies
Key: SuppliesID
FD1: SuppliesID → Type, ItemName, Quantity, AnimalSherlterID(FK)

1NF: Yes, Supplies relation reaches 1NF as it meets the definition of a relation.
2NF: Yes, Supplies relation reaches 2NF as there are no partial key dependencies.
3NF: Yes, Supplies relation reaches 3NF as there are no transitive dependencies.

Final Set of Relations:
Supplies (SuppliesID(KEY), Type, ItemName, Quantity, AnimalShelterID (FK))
Key: SuppliesID
FK: AnimalSherlterID
FD1: SuppliesID → Type, ItemName, Quantity, AnimalSherlterID(FK)

Pet (PetID(KEY), Species, Breed, Name, Birthday, Gender, Color, Height, Weight, Temperament, IntakeDate, IntakeReason, HealthStatus, AnimalShelterID (FK))

PET													
PetID(KEY)	Species	Breed	Name	Birthday	Gender	Color	Height	Weight	Temperament	IntakeDate	IntakeReason	HealthStatus	AnimalShelterID(FK)
P100	Cat	Calico	Kitty	1/2/23	Female	Orange	36in	5lb	Friendly	4/6/23	Stranded	Healthy	A100
P1002	Dog	Pug	Doggy	1/3/23	Female	Tan	57in	10lb	Friendly	4/7/23	Stranded	Healthy	A100
P103	Rabit	Short-hair	Bunny	1/4/23	Male	White	2.3in	4lb	Friendly	4/8/23	Stranded	Healthy	A100
P104	Fish	Golden	Fishy	1/5/23	Female	Gold	1in	.5lb	Friendly	4/9/23	Stranded	Healthy	A100
P105	Turtle	Flat back	Rock	1/6/23	Female	Green	1in	.25lb	Friendly	4/10/23	Stranded	Healthy	A100
P106	Bird	Moca	Fly	1/7/23	Male	Red	20in	.5lb	Friendly	4/11/23	Stranded	Healthy	A100

Pet
Key: PetID
FD1: PetID → Species, Breed, Name, Birthday, Gender, Color, Height, Weight, Temperament
IntakeDate, IntakeReason, HealthStatus, AnimalShelterID (FK)

1NF: Yes, Pet relation reaches 1NF as it meets the definition of a relation.
2NF: Yes, Pet relation reaches 2NF as there are no partial key dependencies.
3NF: Yes, Pet relation reaches 3NF as there are no transitive dependencies.

Final Set of Relations:
Pet (PetID(KEY), Species, Breed, Name, Birthday, Gender, Color, Height, Weight, Temperament, IntakeDate, IntakeReason, HealthStatus, AnimalShelterID (FK))
Key: PetID
FK: AnimalShelterID
FD1: PetID → Species, Breed, Name, Birthday, Gender, Color, Height, Weight, Temperament
IntakeDate, IntakeReason, HealthStatus, AnimalShelterID (FK)

AdoptionApplication (AdoptionApplicationID(KEY), AdoptionStatus (Started, Pending, Completed), AdoptionFileDate, DateAdopted,PatronID(FK), PetID(FK))

ADOPTION APPLICATION					
AdoptionApplicationID (KEY)	AdoptionStatus	AdoptionFileDate	DateAdopted	PetID (FK)	PatronID (FK)
AA110	Started	2/23/2023	N/A	P100	P123
AA111	Pending	2/2/2023	N/A	P1002	P124
AA112	Started	2/24/2023	N/A	P103	P125
AA113	Completed	1/16/2023	4/14/2023	P104	P126
AA114	Completed	1/13/2023	4/6/2023	P105	P127
AA115	Pending	2/3/2023	N/A	P106	P128

AdoptionApplication
Key: AdoptionApplicationID
FD1: AdoptionApplicationID → AdoptionStatus, AdoptionFileDate, DateAdopted, PatronID (FK), PetID(FK)
1NF: Yes, AdoptionApplication relation reaches 1NF as it meets the definition of a relation.
2NF: Yes, AdoptionApplication relation reaches 2NF as there are no partial key dependencies.
3NF: Yes, AdoptionApplication relation reaches 3NF as there are no transitive dependencies.

Final Set of Relations:
AdoptionApplication (AdoptionApplicationID(KEY), AdoptionStatus (Started, Pending, Completed), AdoptionFileDate, DateAdopted,PatronID(FK), PetID(FK))
Key: AdoptionApplicationID
FK: PatronID, PetID
FD1: AdoptionApplicationID → AdoptionStatus, AdoptionFileDate, DateAdopted, PatronID (FK), PetID(FK)

MedicalRecord (MedicalRecordID(KEY),MedicalHistory, VaccinationRecord, PetID(FK))

MEDICAL RECORD			
MedicalRecordID(KEY)	MedicalHistory	VaccinationRecord	PetID(FK)
MR110	Neutered	yes	P100
MR111	Cushing disease	yes	P1002
MR112	Not Neutered	yes	P103
MR113	Neutered	no	P104
MR114	Past Surgery	no	P105
MR115	Past Surgery	no	P106

MedicalRecord

Key: MedicalRecordID

FD1: MedicalRecordID→ MedicalHistory, VaccinationRecord,PetID(FK)

1NF:Yes, MedicalRecord relation meets the definition of a relation

2NF: Yes MedicalRecord Relation reaches 2NF because it has no partial key dependencies.

3NF: Yes MedicalRecord Relation reaches 3NF because it has no transitive dependencies.

Final Set of Relations:

MedicalRecord (MedicalRecordID(KEY),MedicalHistory, VaccinationRecord, PetID(FK))

Key: MedicalRecordID

FK: PetID

FD1: MedicalRecordID→ MedicalHistory, VaccinationRecord,PetID(FK)

Patron (PatronID(KEY), FirstName, MiddleName, LastName, StreetName, City, State, Zipcode, PhoneNumber,Email, Donation)

PATRON									
PatronID(KEY)	FirstName	MiddleName	LastName	StreetName	City	Zipcode	PhoneNumber	Email	Donation
P123	Joy	August	White	9391 James St	NY	11419	(555) 555-1234	JoyAugust123@example.com	20
P124	Mark	Calvin	Joe	7 Foster Ave.	NY	10977	(583) 225-8834	mark.calvin567@example.com	25
P125	Sophia	Jade	Jones	9998 Central Drive	NY	10033	(889) 232-1924	sophia.jade@emailgen123.com	30
P126	Seph	Ryder	Garcia	190 Goldfield Street	NY	11236	(999) 362-8715	seph.ryder@example.com	10
P127	Tyler	Brooklyn	Miller	451 N. Hawthorne Ave	NY	11201	(295) 920-1038	tyler.brooklyn@example.com	5
P128	Troy	Mauve	Davis	7270 Glendale Rd	NY	11228	(382) 183-1030	troy.mauve@example.com	1

Patron

Key: PatronID

FD1: PatronID → FirstName, MiddleName, LastName, StreetName, City, State, Zipcode, PhoneNumber,Email, Donation

1NF: Yes, Patron relation meets the definition of a relation

2NF: Yes Patron Relation reaches 2NF because it has no partial key dependencies.
3NF: Yes Patron Relation reaches 3NF because it has no transitive dependencies.

Final Set of Relations:
Patron (PatronID(KEY), FirstName, MiddleName, LastName, StreetName, City, State, Zipcode, PhoneNumber,Email, Donation)
Key: PatronID
FD1: PatronID → FirstName, MiddleName, LastName, StreetName, City, State, Zipcode, PhoneNumber,Email, Donation

---

Personnel_Task (PersonnelID (FK) (KEY), (TaskID(FK)(KEY))

Personnel_Task	
PersonnelID (KEY)	TaskID (KEY)
E100	T100
E102	T1002
E103	T1003
E104	T1004
E105	T1005
E106	T1006

Personnel_Task
Key: PersonnelID, TaskID

1NF: Yes, Personnel_Task relation meets the definition of a relation
2NF: Yes Personnel_Task reaches 2NF because it has no partial key dependencies.
3NF: Yes Personnel_Task reaches 3NF because it has no transitive dependencies.

Final Set of Relations:
Personnel_Task (PersonnelID (FK) (KEY), (TaskID(FK)(KEY))
Key: PersonnelID, TaskID

Milestone 4: Database Schema Physical Modeling

CREATING TABLES

```
1)
CREATE TABLE AnimalShelter
(
  AnimalShelterID     CHAR(5) NOT NULL PRIMARY KEY,
  Name                     CHAR(75) NOT NULL,
  Email                      CHAR(50),
  PhoneNumber         CHAR(12),
  StreetName             CHAR(30)
  City                          CHAR(20),
  State                        CHAR(2)
  ZipCode]                  CHAR(10)
);

2)
CREATE TABLE Personnel
(
  PersonnelID       VARCHAR(10) NOT NULL,
  FirstName         VARCHAR(35) NOT NULL,
  MiddleName   VARCHAR(35) NOT NULL,
  LastName          VARCHAR(35),
  Birthday               DATE,
  Street       VARCHAR(20),
  City              VARCHAR(36),
  State             VARCHAR(4),
  Zipcode           VARCHAR(10),
  PhoneNumber       VARCHAR(12),
  Email                  VARCHAR(20),
  Position               VARCHAR(20),
  Paytype           VARCHAR(10),
  AnimalShelterID   VARCHAR(10) NOT NULL,
  CONSTRAINT pk_Personnel
     PRIMARY KEY (PersonnelID)
);

3)
CREATE TABLE Task
(
TaskID              VARCHAR(225) NOT NULL,
TaskName            VARCHAR(225),
TaskDescription               VARCHAR(225),
Priority            VARCHAR(225),
Status                 VARCHAR(225),
StartDate           DATE,
EndDate             DATE,
CompletionDate      DATE,
```

```
AnimalsInvolved          VARCHAR(225)
CONSTRAINT PK_Task
PRIMARY KEY (TaskID)
);

4)
CREATE TABLE Supplies
(
SuppliesID          VARCHAR(5) NOT NULL,
Type           VARCHAR(75) NOT NULL,
ItemName       VARCHAR(50),
Quantity            INTEGER,
AnimalShelterID       VARCHAR(5),
CONSTRAINT PK_SupplieID
PRIMARY KEY(SuppliesID)
);

5)
CREATE TABLE Pet
(
PetID          VARCHAR(5) NOT NULL PRIMARY KEY,
Species             VARCHAR(20) NOT NULL,
Breed               VARCHAR(20),
Name                VARCHAR(20),
Birthday                 DATE,
Gender         VARCHAR(1),
Color               VARCHAR(20),
Height              FLOAT,
Weight         FLOAT,
Temperament    VARCHAR(25),
IntakeDate          DATE,
IntakeReason   VARCHAR(200),
HealthStatus   VARCHAR(100),
AnimalShelterID     VARCHAR (10)
CONSTRAINT pk_pet
PRIMARY KEY (PetID)
);

6)
CREATE TABLE AdoptionApplication (
AdoptionApplicationID    VARCHAR(10) NOT NULL,
AdoptionStatus           VARCHAR(225),
AdoptionFileDate         DATE,
DateAdopted              DATE,
CONSTRAINT PK_AdoptionApplication,
PRIMARY KEY (AdoptionApplicationID),
FOREIGN KEY (PatronID) REFERENCE Patron(PatronID),
```

```
FOREIGN KEY (PetID) REFERENCE Pet(PetID)
);

7)
CREATE TABLE MedicalRecord
(
MedicalRecordID          VARCHAR(5) NOT NULL PRIMARY KEY,
MedicalHistory           VARCHAR(75) NOT NULL,
VaccinationRecord        VARCHAR(50),
PetID                     VARCHAR(20),
CONSTRAINT PK_MedicalRecordID PRIMARY KEY(MedicalRecordID));

8)
CREATE TABLE Patron
(
PatronID            VARCHAR(5) NOT NULL PRIMARY KEY,
FirstName      VARCHAR(75) NOT NULL,
LastName            VARCHAR(75) NOT NULL,
StreetName          VARCHAR(75),
City                VARCHAR(20),
State               VARCHAR(2),
ZipCode             VARCHAR(10),
PhoneNumber    VARCHAR(21),
Email               VARCHAR(50),
Donation       INTEGER
);

9)
CREATE TABLE Personnel_task
(
PersonnelID    VARCHAR(10),
TaskID         VARCHAR(225),
FOREIGN KEY (PersonnelID) REFERENCES Personnel(PersonnelID)
FOREIGN KEY (TaskID) REFERENCES Task(TaskID)
);
```

Adding Foreign Keys

```
ALTER TABLE Personnel
ADD CONSTRAINT fk_personnel_AnimalShelterID
FOREIGN KEY (AnimalShelterID)
REFERENCES AnimalShelter(AnimalShelterID);

ALTER TABLE Supplies
ADD CONSTRAINT fk_supplies_AnimalShelterID
FOREIGN KEY (AnimalShelterID)
REFERENCES AnimalShelter(AnimalShelterID);
```

```
ALTER TABLE Pet
ADD CONSTRAINT fk_pet_AnimalShelterID
FOREIGN KEY (AnimalShelterID)
REFERENCES AnimalShelter(AnimalShelterID);

ALTER TABLE AdoptionApplication
ADD CONSTRAINT fk_AdoptionApplication_PatronID
FOREIGN KEY (PatronID)
REFERENCES Patron(PatronID);

ALTER TABLE AdoptionApplication
ADD CONSTRAINT fk_AdoptionApplication_PetID
FOREIGN KEY (PetID)
REFERENCES Pet(PetID);

ALTER TABLE MedicalRecord
ADD CONSTRAINT fk_MedicalRecord_PETID
FOREIGN KEY (PetID) REFERENCES Pet(PetID)

ALTER TABLE Personnel_task
ADD CONSTRAINT fk_personnel_task_personnel
FOREIGN KEY (PersonnelID)
REFERENCES Personnel(PersonnelID);

ALTER TABLE Personnel_task
ADD CONSTRAINT fk_personnel_task_task
FOREIGN KEY (TaskID)
REFERENCES Task(TaskID);
```

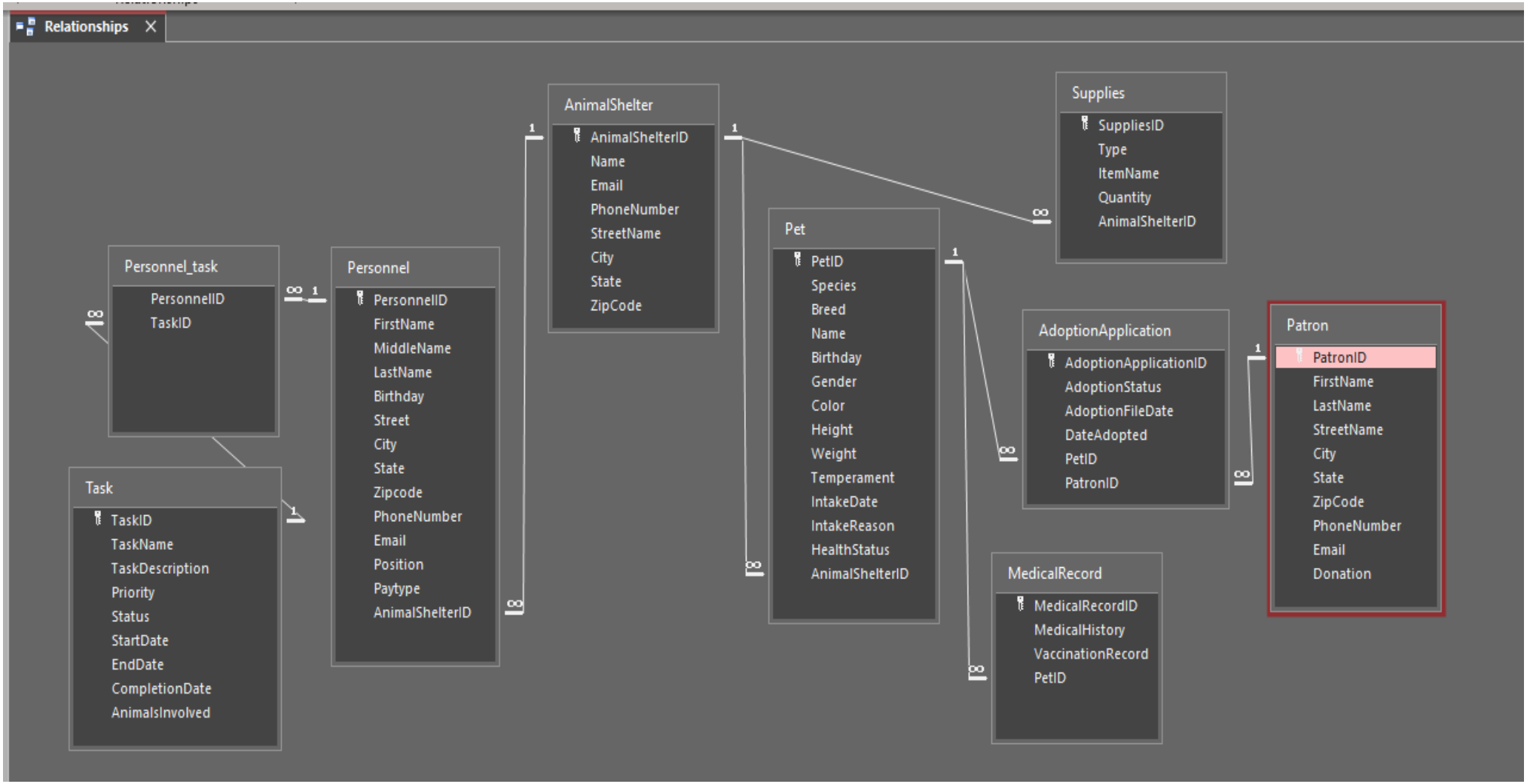
Adding Data to the Tables using SQL INSERT Statements

```
INSERT INTO AnimalShelter VALUES ('A100', 'Animal Haven',
'animalhaven@example.com', '543-777-2349', '432 Oaks Ave.',
'NY', 'NY', '11433');

INSERT INTO Personnel VALUES ('E100', 'Joy', 'August', 'White',
'1/3/1999', '9391 James St', 'NY', 'NY', '11419', '(555) 555-
1234', 'JoyAugust123@example.com', 'Manager', 'Hourly', 'A100');

INSERT INTO Personnel VALUES ('E102', 'Mark', 'Calvin', 'Joe',
'3/4/2000', '7 Foster Ave.', 'NY', 'NY', '10977', '(583) 225-
8834', 'mark.calvin567@example.com', 'Nurse', 'Hourly', 'A100');
```

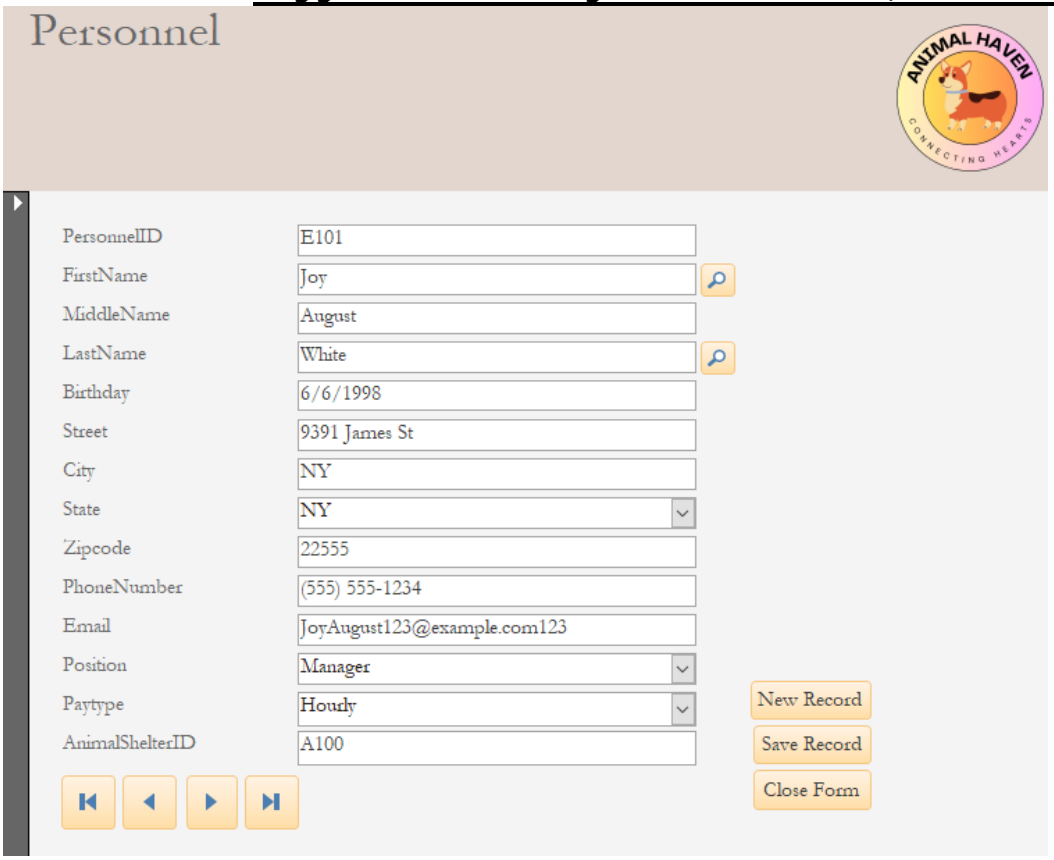
```
INSERT INTO Personnel VALUES ('E103', 'Sophia', 'Jade', 'Jones',
'4/5/2001', '9998 Central Drive', 'NY', 'NY', '10033', '(889)
232-1924', 'sophia.jade@emailgen123.com', 'Vet', 'Hourly',
'A100');
INSERT INTO Personnel VALUES ('E104', 'Seph', 'Ryder', 'Garcia',
'6/12/2002', '190 Goldfield Street', 'NY', 'NY', '11236', '(999)
362-8715', 'seph.ryder@example.com', 'Graphic Designer',
'Volunteer', 'A100');

INSERT INTO Personnel VALUES ('E105', 'Tyler', 'Brooklyn',
'Miller', '9/12/2003', '451 N. Hawthorne Ave', 'NY', 'NY',
'11201', '(295) 920-1038', 'tyler.brooklyn@example.com',
'Receptionist', 'Hourly', 'A100');

INSERT INTO Personnel VALUES ('E106', 'Troy', 'Mauve', 'Davis',
'7/14/2002', '7270 Glendale Rd', 'NY', 'NY', '11228', '(382)
183-1030', 'troy.mauve@example.com', 'Receptionist', 'Hourly',
'A100');

INSERT INTO Task VALUES (T001, 'Dogwash', 'Washing', 'Low',WIP
'1/2/23', '1/10/23', NULL, ‘Dog');
INSERT INTO Task VALUES (T002, 'Catwash', 'Washing', 'Low',
’WIP, '1/3/23', '1/11/23', NULL, ‘Cat');
INSERT INTO Task VALUES (T003, 'Checkup', “Bring to Vets”,
'Medium', ’WIP '1/12/23', '1/10/23', NULL, ‘Cat');
INSERT INTO Task VALUES (T004, 'Walk', “Walk the dog”,
'Low',’Done’ '1/5/23', '1/13/23', '1/6/23’, ‘Dog');
INSERT INTO Task VALUES (T005, 'Vaccine’', “Bring to Vets to
perform shot”, 'High',’Done’ '1/6/23', '1/14/23', '1/7/23',
‘Rabbit');
INSERT INTO Task VALUES (T006, 'Neuter', “Bring to Vets to
perform operations”, 'High',’Done’ '1/7/23', '1/15/23',
'1/8/23', ‘Cat');

INSERT INTO Supplies VALUES ('S100', 'Leash', 'Dog Leash', 10,
'A100');
INSERT INTO Supplies VALUES ('S102', 'Bowl', 'Dog Bowl', 20,
'A100');
INSERT INTO Supplies VALUES ('S103', 'Food', 'Cat Food', 5,
'A100');
INSERT INTO Supplies VALUES ('S104', 'Bed', 'Cat Bed', 3,
'A100');
INSERT INTO Supplies VALUES ('S105', 'Sweater', 'Cat Sweater',
5, 'A100');
INSERT INTO Supplies VALUES ('S106', 'Toy', 'Cat toy', 30,
'A100');
```

```
INSERT INTO Pet VALUES ('P100', 'Cat', 'Calico', 'Kitty',
'1/2/2023', 'Female', 'Orange', '36', '5', 'Friendly',
'4/6/2023', 'Stranded', 'Healthy', 'A100');
INSERT INTO Pet VALUES ('P102', 'Dog', 'Pug', 'Doggy',
'1/3/2023', 'Female', 'Tan', '57', '10', 'Friendly', '4/7/2023',
'Stranded', 'Healthy', 'A100');
INSERT INTO Pet VALUES ('P103', 'Rabbit', 'Short-hair', 'Bunny',
'1/4/2023', 'Male', 'White', '2.3', '4', 'Friendly', '4/8/2023',
'Stranded', 'Healthy', 'A100');
INSERT INTO Pet VALUES ('P104', 'Fish', 'Golden', 'Fishy',
'1/5/2023', 'Female', 'Gold', '1', '0.5', 'Friendly',
'4/9/2023', 'Stranded', 'Healthy', 'A100');
INSERT INTO Pet VALUES ('P105', 'Turtle', 'Flat back', 'Rock',
'1/6/2023', 'Female', 'Green', '1', '0.25', 'Friendly',
'4/10/2023', 'Stranded', 'Healthy', 'A100');
INSERT INTO Pet VALUES ('P106', 'Bird', 'Moca', 'Fly',
'1/7/2023', 'Male', 'Red', '20', '0.5', 'Friendly', '4/11/2023',
'Stranded', 'Healthy', 'A100');

INSERT INTO AdoptionApplication VALUES ('AA110', “Started”,
'2/23/2023', NULL, 'P100', 'PA123');
INSERT INTO AdoptionApplication VALUES ('AA111', “Pending”,
'2/2/2023', NULL, 'P102', 'PA124');
INSERT INTO AdoptionApplication VALUES ('AA112', “Started”,
'2/24/2023', NULL , 'P103', 'PA125');
INSERT INTO AdoptionApplication VALUES ('AA113', “Completed”,
'1/16/2023', '4/14/2023', 'P104', 'PA126');
INSERT INTO AdoptionApplication VALUES ('AA114', “Completed”,
'1/13/2023', '4/6/2023', 'P105', 'PA127');
INSERT INTO AdoptionApplication VALUES ('AA115', “Pending”,
'2/3/2023',NULL , 'P106', 'PA128');

INSERT INTO MedicalRecord VALUES ('MR110', 'Neutered', 'Yes',
'P100');
INSERT INTO MedicalRecord VALUES ('MR111', 'Cushing disease',
'Yes', 'P102');
INSERT INTO MedicalRecord VALUES ('MR112', 'Not Neutered',
'Yes', 'P103');
INSERT INTO MedicalRecord VALUES ('MR113', 'Neutered', 'No',
'P104');
INSERT INTO MedicalRecord VALUES ('MR114', 'Past Surgery', 'No',
'P105');
INSERT INTO MedicalRecord VALUES ('MR115', 'Past Surgery', 'N
     o', 'P106');
```

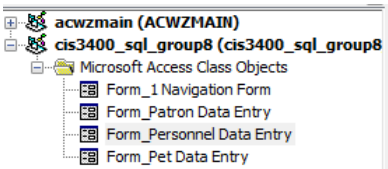
```
INSERT INTO Patron VALUES ('PA123', 'Joy', 'August', 'White',
'9391 James St', 'NY', '11419', '(555) 555-1234',
'JoyAugust123@example.com', 20);
INSERT INTO Patron VALUES ('PA124', 'Mark', 'Calvin', 'Joe', '7
Foster Ave.', 'NY', '10977', '(583) 225-8834',
'mark.calvin567@example.com', 25);
INSERT INTO Patron VALUES ('PA125', 'Sophia', 'Jade', 'Jones',
'9998 Central Drive', 'NY', '10033', '(889) 232-1924',
'sophia.jade@emailgen123.com', 30);
INSERT INTO Patron VALUES ('PA126', 'Seph', 'Ryder', 'Garcia',
'190 Goldfield Street', 'NY', '11236', '(999) 362-8715',
'seph.ryder@example.com', 10);
INSERT INTO Patron VALUES ('PA127', 'Tyler', 'Brooklyn',
'Miller', '451 N. Hawthorne Ave', 'NY', '11201', '(295) 920-
10387', 'tyler.brooklyn@example.com', 5);
INSERT INTO Patron VALUES ('PA128', 'Troy', 'Mauve', 'Davis',
'7270 Glendale Rd', 'NY', '11228', '(382) 183-1030',
'troy.mauve@example.com', 1);

INSERT INTO Personnel_task VALUES ('E100', 'T001');
INSERT INTO Personnel_task VALUES ('E102', 'T002');
INSERT INTO Personnel_task VALUES ('E103', 'T003');
INSERT INTO Personnel_task VALUES ('E104', 'T004');
INSERT INTO Personnel_task VALUES ('E105', 'T005');
INSERT INTO Personnel_task VALUES ('E106', 'T006');
```


Relationships
AnimalShelter
AnimalShelterID
Name
Email
PhoneNumber
StreetName
City
State
ZipCode
Supplies
SuppliesID
Type
ItemName
Quantity
AnimalShelterID
Personnel_task
PersonnelID
TaskID
Personnel
PersonnelID
FirstName
MiddleName
LastName
Birthday
Street
City
State
Zipcode
PhoneNumber
Email
Position
Paytype
AnimalShelterID
Pet
PetID
Species
Breed
Name
Birthday
Gender
Color
Height
Weight
Temperament
IntakeDate
IntakeReason
HealthStatus
AnimalShelterID
AdoptionApplication
AdoptionApplicationID
AdoptionStatus
AdoptionFileDate
DateAdopted
PetID
PatronID
Patron
PatronID
FirstName
LastName
StreetName
City
State
ZipCode
PhoneNumber
Email
Donation
Task
TaskID
TaskName
TaskDescription
Priority
Status
StartDate
EndDate
CompletionDate
AnimalsInvolved
MedicalRecord
MedicalRecordID
MedicalHistory
VaccinationRecord
PetID

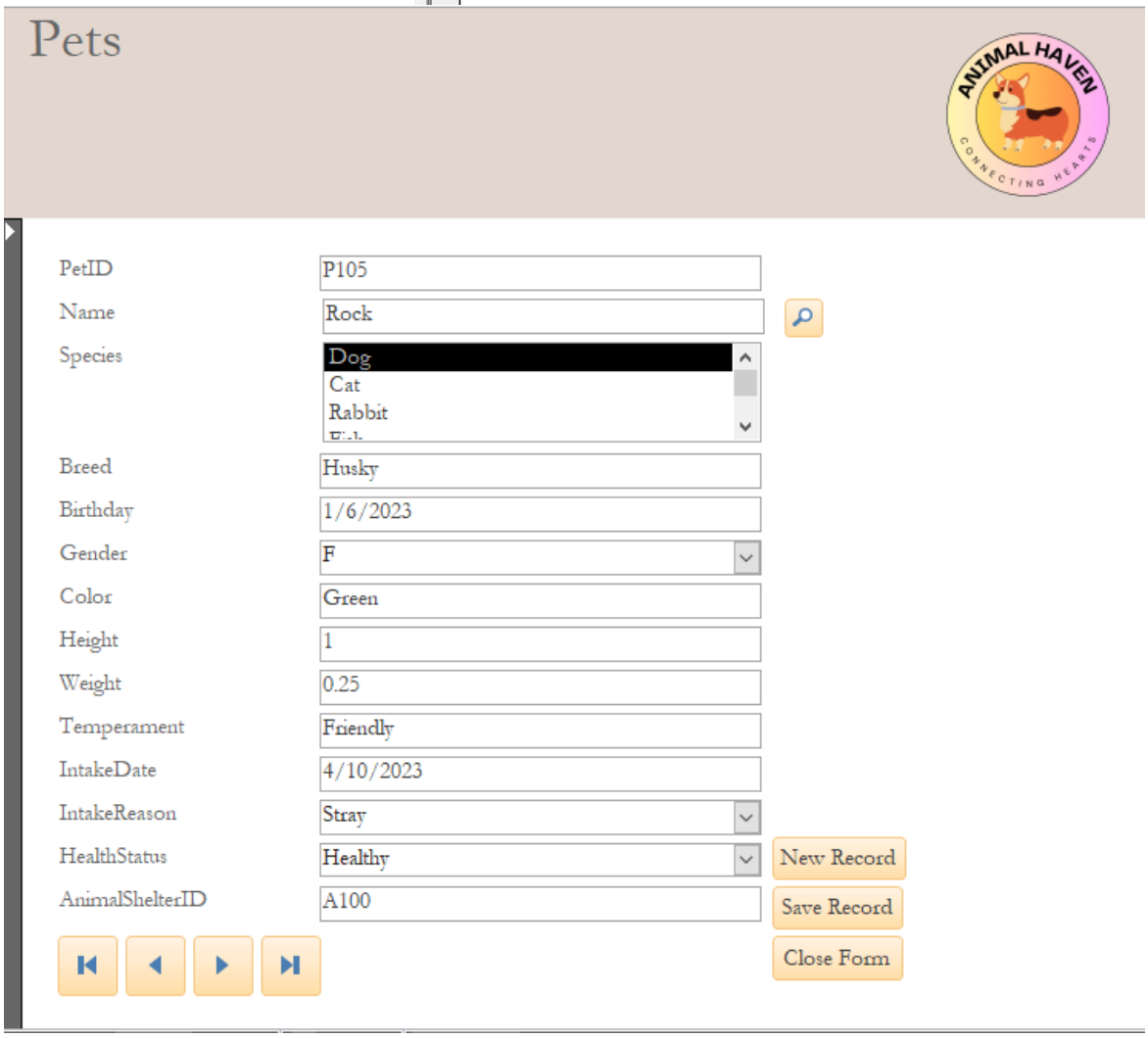
Milestone 5
Application Implementation | Final Report



The first form is a Personnel data entry, the magnifying glass next to First and Last name allows for a filter search for similar results like what was typed. Below is an example of the VBA code used to build the filter button. As for the other buttons, they were built using the button wizard, they are republicated throughout all the forms for ease of use.



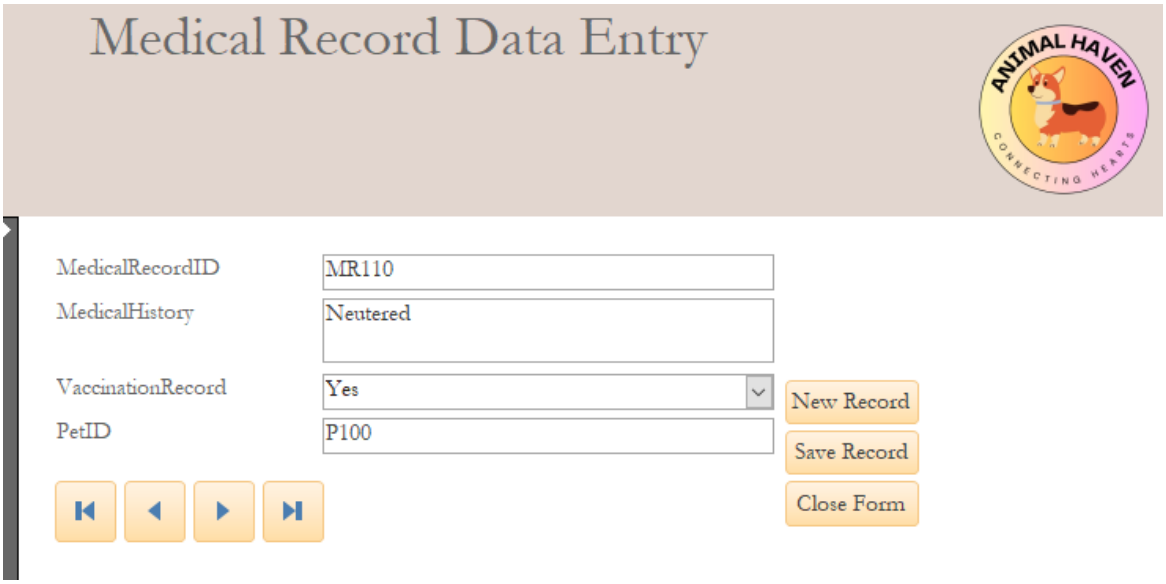

```
Private Sub FirstName_AfterUpdate()
    FirstName = StrConv(FirstName, cvProperCase)

End Sub

Private Sub LastName_AfterUpdate()
    LastName = StrConv(LastName, cvProperCase)
End Sub

Private Sub MiddleName_AfterUpdate()
    MiddleName = StrConv(MiddleName, cvProperCase)
End Sub

Private Sub Search_Last_Name_Button_Click()
 Dim S As String
    S = InputBox("Please enter last name", "Last Name")
    If S = "" Then Exit Sub

    Me.Filter = "LastName Like '*" & S & "*'"
    Me.FilterOn = True
End Sub

Private Sub SearchFirstNameButton_Click()

    Dim S As String
    S = InputBox("Please enter first name", "First Name")
    If S = "" Then Exit Sub

    Me.Filter = "FirstName Like '*" & S & "*'"
    Me.FilterOn = True
```



This form allows the data entry of pets. There are drop downs for species(dog, cat, etc), gender (m or f), intake reasons (stray or surrender), and health status(healthy or not)

```
Option Compare Database

Private Sub Name_AfterUpdate()
    Name = StrConv(Name, cvProperCase)
End Sub
```

```
Private Sub Search_Pet_Name_Button_Click()
    Dim S As String
    S = InputBox("Please enter pet name", "Pet Name")
    If S = "" Then Exit Sub

    Me.Filter = "PetName Like '*" & S & "*'"
    Me.FilterOn = True
End Sub
```



Medical Record Data Entry form to record whether the animal has medical history for record keeping.

Adoption Application Data Entry

AdoptionApplicationID: 111

AdoptionStatus: Pending

AdoptionFileDate: 2/2/2023

DateAdopted:

PetID: P102

PatronID: PA124

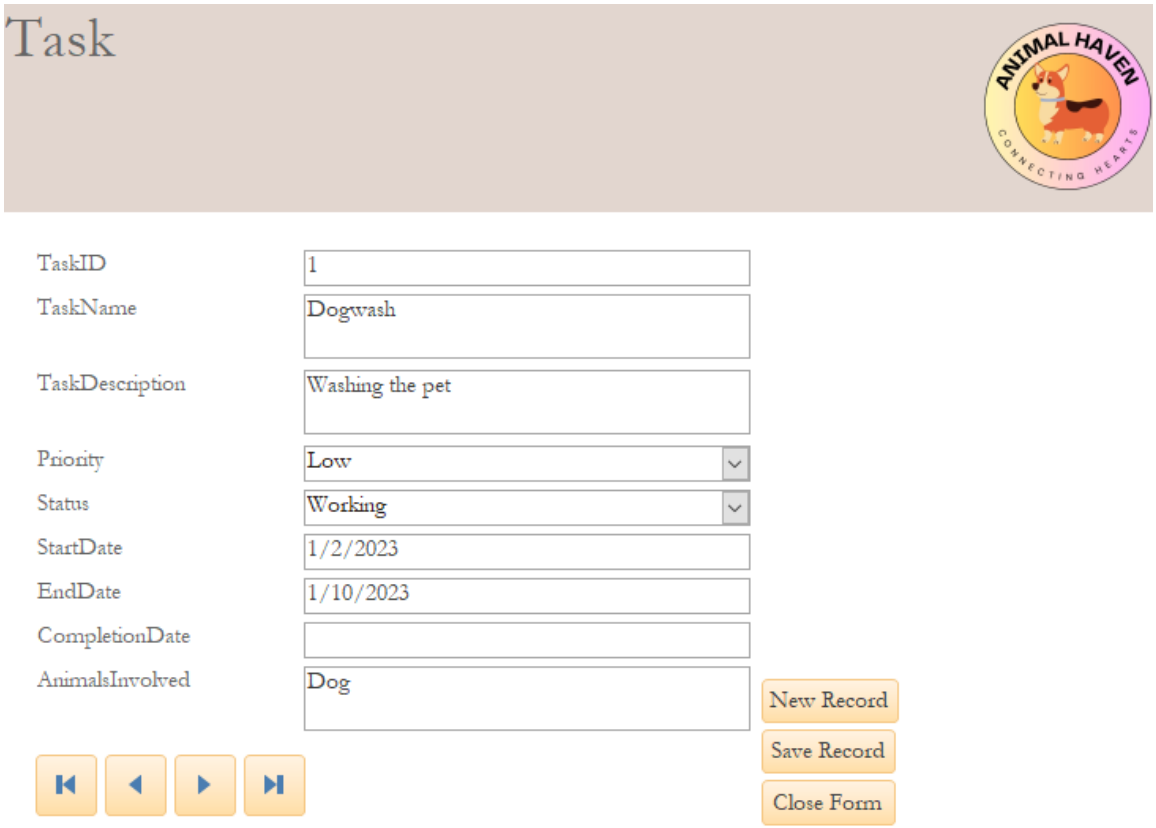
New Record

Save Record

Close Form

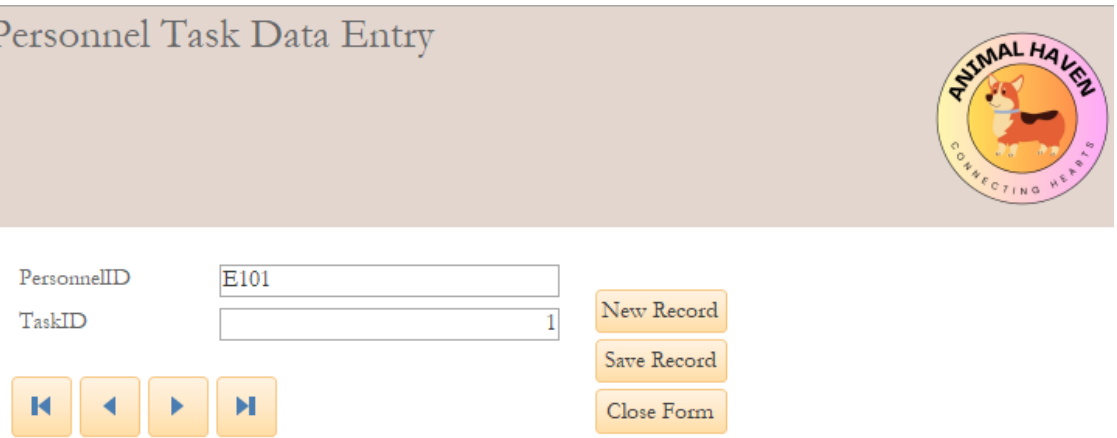
This form allows the recording of the pet adoption process/applications. PETID I have set to only display available pets which was found using a SQL Query (“Available Pets for Adoptions”) to reduce double entry.

Task

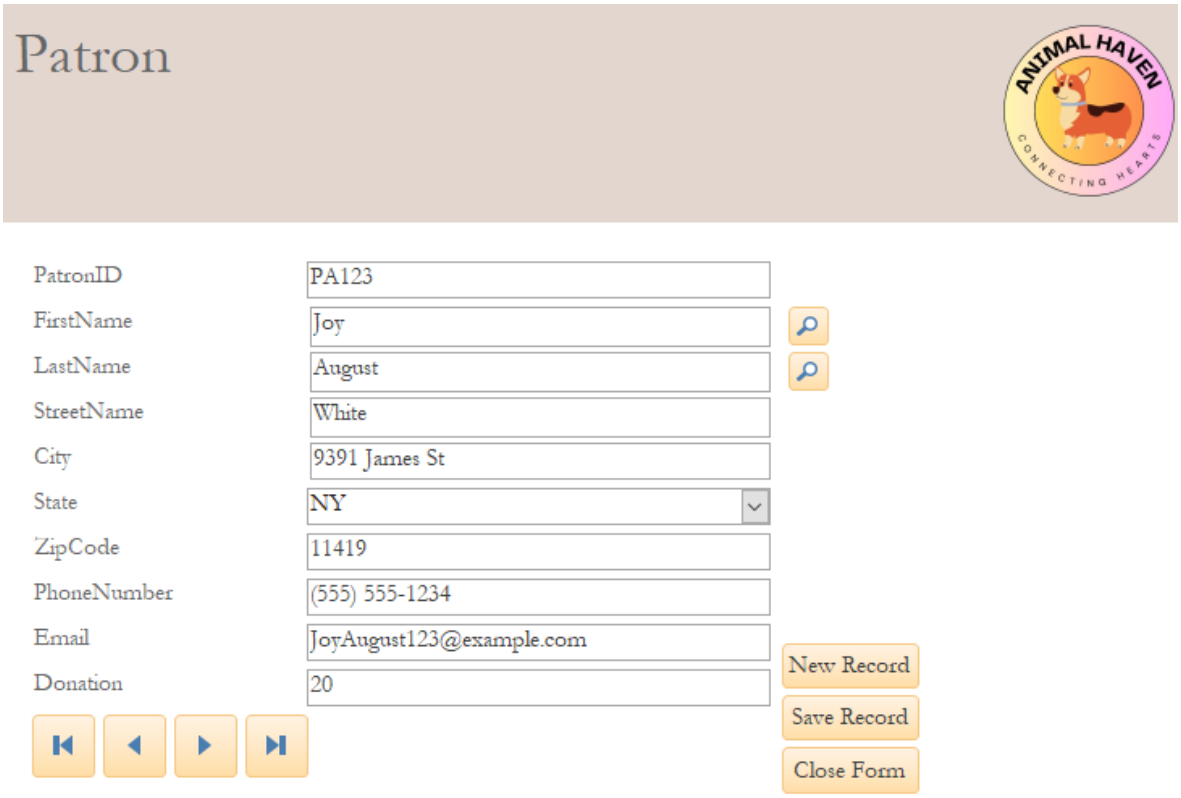


This form records tasks, whether it's daily or weekly, and records its urgency. It allows for better operation management, it is important because staff count is low.

Personnel Task Data Entry



This form records who is going to work on what task. For better operation management.



This form allows us to keep track of how to adopt animals from us. The filters from before were replicated here to help find information easier. In the future, the donation should be a whole separate table, but for the project, this is enough.

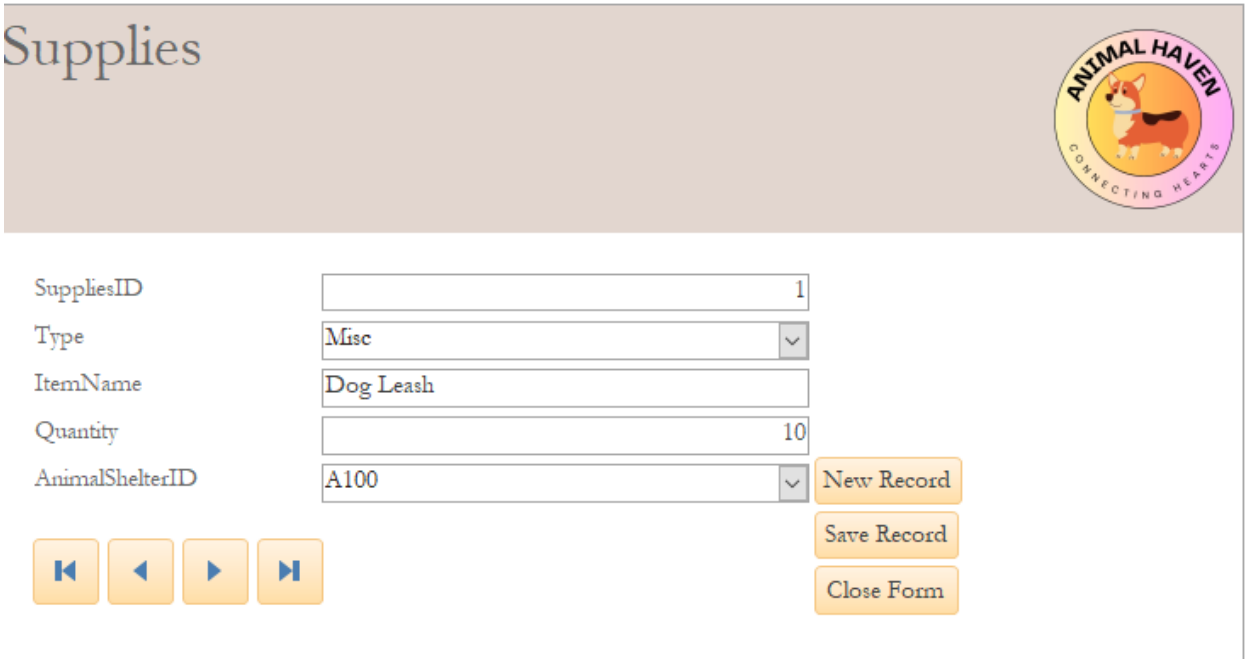
```
Option Compare Database
```

```
Private Sub FirstName_AfterUpdate()
    FirstName = StrConv(FirstName, vbProperCase)

End Sub
```

```
Private Sub LastName_AfterUpdate()
    LastName = StrConv(LastName, vbProperCase)
End Sub
```

```
Private Sub Search_Last_Name_Button_Click()
    Dim S As String
    S = InputBox("Please enter last name", "Last Name")
    If S = "" Then Exit Sub

    Me.Filter = "LastName Like '*" & S & "*'"
    Me.FilterOn = True

End Sub
```

```
Private Sub SearchFirstNameButton_Click()
    Dim S As String
    S = InputBox("Please enter first name", "First Name")
    If S = "" Then Exit Sub

    Me.Filter = "FirstName Like '*" & S & "*'"
    Me.FilterOn = True
End Sub
```



This form helps us track the quantity of supplies we have in hand, and help us figure out what we will need in the future.

Reporting Queries:

Report: Supplies

Wednesday, December 13, 2023
9:49:29 PM

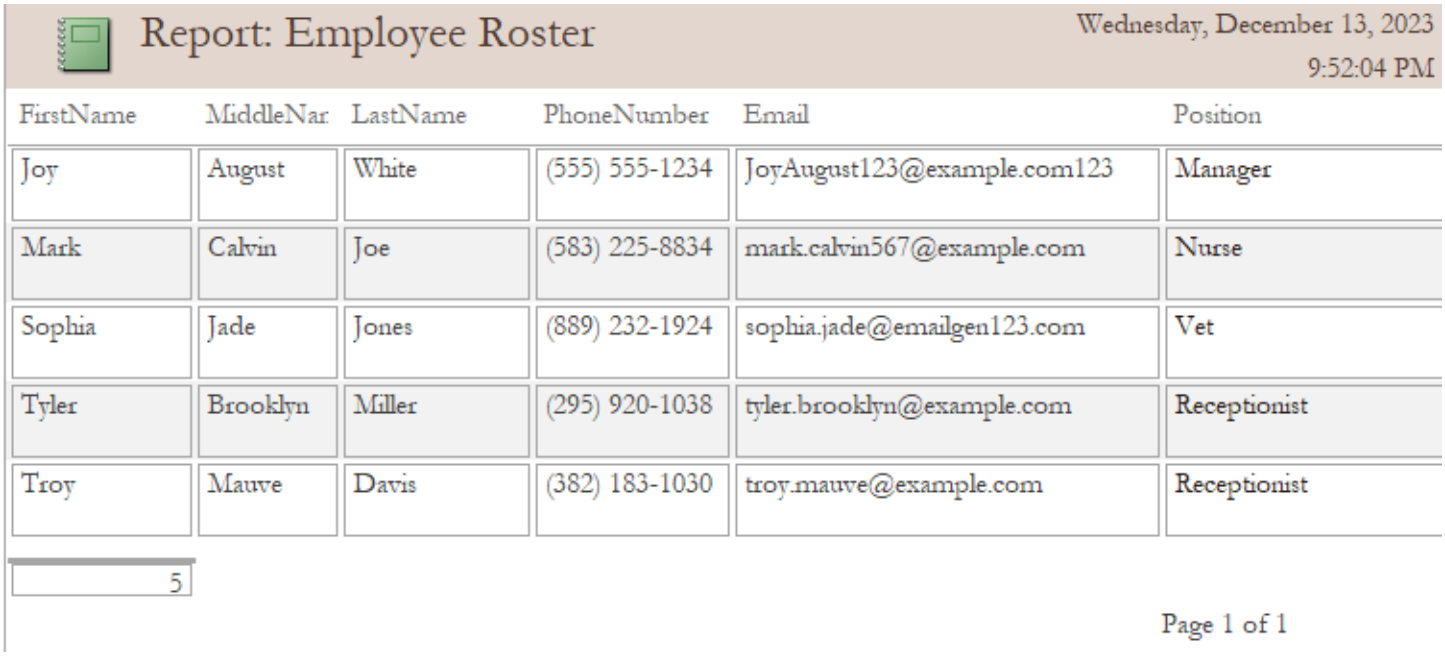
SuppliesID	Type	ItemName	Quantity	AnimalShe
1	Misc	Dog Leash	10	A100
2	Misc	Dog Bowl	20	A100
3	Food	Cat Food	5	A100
4	Beds	Cat Bed	3	A100
5	Misc	Cat Sweater	5	A100
6	Toys	Cat toy	30	A100
7	Food	Wet Cat Food 4Oz Tins	200	A100
8	Food	Kibbles for Dogs	20	A100

8

Page 1 of 1

Supplies report help us keep track of what we have on hand.

SQL CODE:
```
SELECT *
FROM Supplies;
```



Report: Employee Roster

Wednesday, December 13, 2023
9:52:04 PM

FirstName	MiddleNar	LastName	PhoneNumber	Email	Position
Joy	August	White	(555) 555-1234	JoyAugust123@example.com123	Manager
Mark	Calvin	Joe	(583) 225-8834	mark.calvin567@example.com	Nurse
Sophia	Jade	Jones	(889) 232-1924	sophia.jade@emailgen123.com	Vet
Tyler	Brooklyn	Miller	(295) 920-1038	tyler.brooklyn@example.com	Receptionist
Troy	Mauve	Davis	(382) 183-1030	troy.mauve@example.com	Receptionist

5

Page 1 of 1

1. Employee Roster keeps track of who our current employees are and their point of contact.

SQL CODE:

```
SELECT Personnel.FirstName, Personnel.MiddleName, Personnel.LastName,
Personnel.PhoneNumber, Personnel.Email, Personnel.Position
FROM Personnel
WHERE (((Personnel.Paytype)<>'Volunteer'))
ORDER BY Personnel.LastName DESC;
```

Report: Volunteers Roster

Wednesday, December 13, 2023
9:53:30 PM

FirstName	MiddleName	LastName	PhoneNumber	Email
Seph	Ryder	Garcia	(999) 362-8715	seph.ryder@example.com
Troy	A	Bolton	9142448978	troy.bolton@gmail.com
Gabriella	A	Montez	9175667887	gabriella.montez@gmail.com
Chuck	A	Norris	9144444444	chuck.norris@gmail.com

4

Page 1 of 1

2. Volunteer report allows us to keep track of our volunteers. For times we need more help, this report allows us to send emails out to them and encourage them to come help.

SQL CODE:

```
SELECT FirstName, MiddleName, LastName, PhoneNumber, Email
FROM Personnel
WHERE Paytype = 'Volunteer'
ORDER BY LastName DESC;
```

Report: Donations

Wednesday, December 13, 2023
9:55:17 PM

FirstName	LastName	Email	Donation
Joy	August	JoyAugust123@example.com	$20.00
Mark	Calvin	mark.calvin567@example.com	$25.00
Sophia	Jade	sophia.jade@emailgen123.com	$30.00
Seph	Ryder	seph.ryder@example.com	$10.00
Tyler	Brooklyn	tyler.brooklyn@example.com	$5.00
Troy	Mauve	troy.mauve@example.com	$1.00

6

Page 1 of 1

3. Donation reports allow us to see what was donated to us and will allow us to send thank you emails. In the future, Donations should be a separate table that allows to track each donation transaction. With fields like (Campaign/title, Donation Amount, Donation time/date, etc), this will allow us to further track how many times they donated.

```
SQL CODE:
SELECT Patron.FirstName, Patron.LastName, Patron.Email, Patron.Donation
FROM Patron
ORDER BY Patron.Donation DESC;
```

Adoption Application History

Wednesday, December 13, 2023
9:57:52 PM

Adoption Application ID	AdoptionStatus	AdoptionFileDate	DateAdopted	PetID	Name	Species	Breed	PatronID	FirstName	LastName	PhoneNumber	Email
111	Pending	2/2/2023		P102	Doggy	Dog	Pug	PA124	Mark	Calvin	(583) 225-8834	mark.calvin567@example.com
112	Started	2/24/2023		P103	Bunny	Rabbit	Short-hair	PA125	Sophia	Jade	(889) 232-1924	sophia.jade@emailgen123.com
113	Completed	1/16/2023	4/14/2023	P104	Fishy	Fish	Golden	PA126	Seph	Ryder	(999) 362-8715	seph.ryder@example.com
114	Completed	1/13/2023	4/6/2023	P105	Rock	Dog	Husky	PA127	Tyler	Brooklyn	(295) 920-10387	tyler.brooklyn@example.com
115	Pending	2/3/2023		P106	Fly	Bird	Moca	PA128	Troy	Mauve	(382) 183-1030	troy.mauve@example.com
116	Started	2/23/2023		P100	Kitty	Dog	Calico	PA123	Joy	August	(555) 555-1234	JoyAugust123@example.com

6

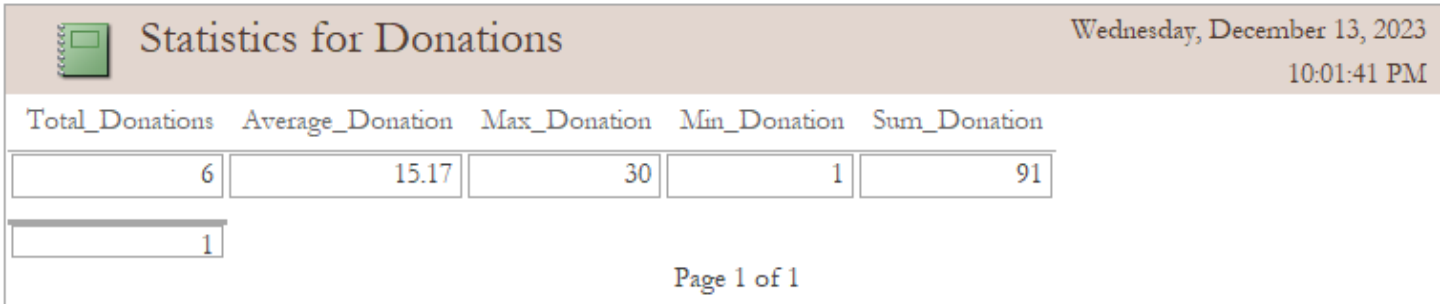
Page 1 of 1

4. The Adoption Application History report allows us to see the history. It takes the Application table as a base and left joins from the Pet and Patron table for basic information, to let us see the status of the adoption, what pet, and who is adopting.

SQL CODE:

```
SELECT AdoptionApplication.AdoptionApplicationID,
AdoptionApplication.AdoptionStatus, AdoptionApplication.AdoptionFileDate,
AdoptionApplication.DateAdopted, Pet.PetID, Pet.Name, Pet.Species,
Pet.Breed, Patron.PatronID, Patron.FirstName, Patron.LastName,
Patron.PhoneNumber, Patron.Email
FROM Pet INNER JOIN (Patron INNER JOIN AdoptionApplication ON
Patron.PatronID = AdoptionApplication.PatronID) ON Pet.PetID =
AdoptionApplication.PetID
ORDER BY AdoptionApplication.AdoptionStatus DESC;
```

Business Question Queries:



Statistics for Donations

Wednesday, December 13, 2023
10:01:41 PM

Total_Donations	Average_Donation	Max_Donation	Min_Donation	Sum_Donation
6	15.17	30	1	91

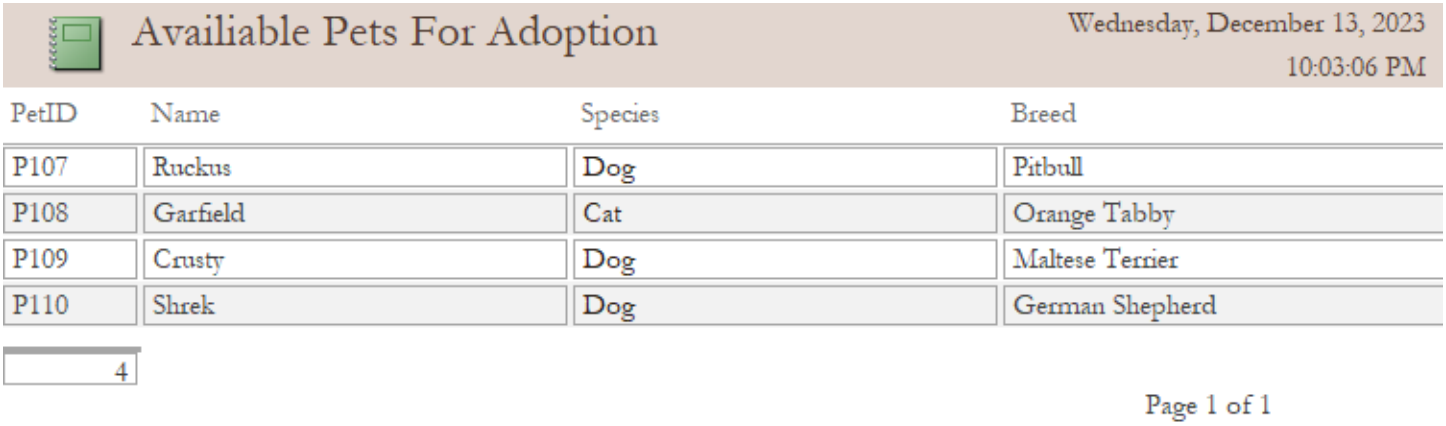
1

Page 1 of 1

1. This query is used to determine the average amount that our patrons are donating to our animal shelter. It helps us plan ahead for our yearly budgeting finances to determine what we can afford for the upcoming future.

SQL CODE:

```
SELECT Count(*) AS Total_Donations, Round(Avg(Donation),2) AS
Average_Donation, Max(Donation) AS Max_Donation, Min(Donation) AS
Min_Donation, Sum(Donation) AS Sum_Donation
FROM Patron;
```



Availiable Pets For Adoption

Wednesday, December 13, 2023
10:03:06 PM

PetID	Name	Species	Breed
P107	Ruckus	Dog	Pitbull
P108	Garfield	Cat	Orange Tabby
P109	Crusty	Dog	Maltese Terrier
P110	Shrek	Dog	German Shepherd

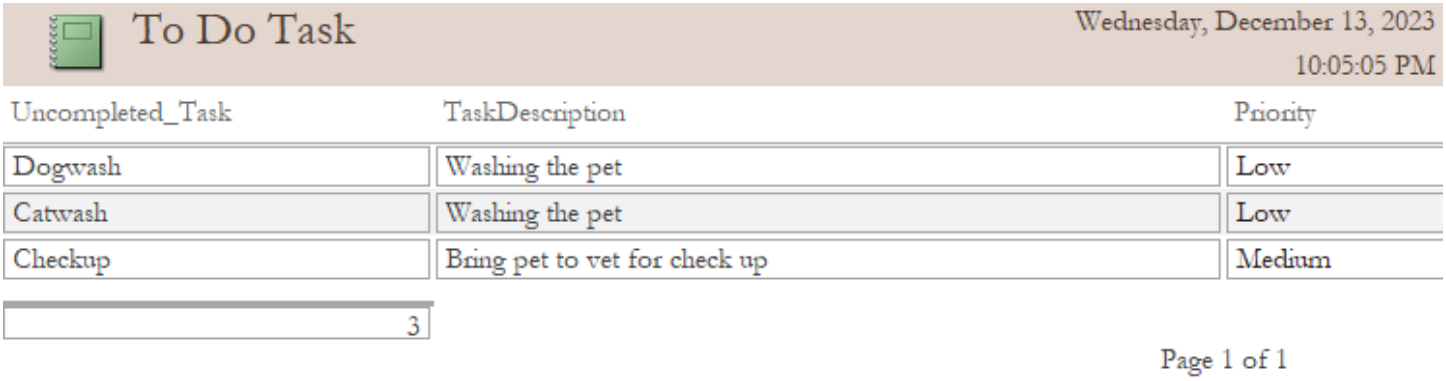
4

Page 1 of 1

2. This report displays the pets that are available for adoption.

SQL CODE:

```
SELECT P.PetID, P.Name, P.Species, P.Breed
FROM Pet AS P LEFT JOIN AdoptionApplication AS AA ON P.PetID = AA.PetID
WHERE AA.PetID IS NULL;
```



To Do Task

Wednesday, December 13, 2023
10:05:05 PM

Uncompleted_Task	TaskDescription	Priority
Dogwash	Washing the pet	Low
Catwash	Washing the pet	Low
Checkup	Bring pet to vet for check up	Medium

3

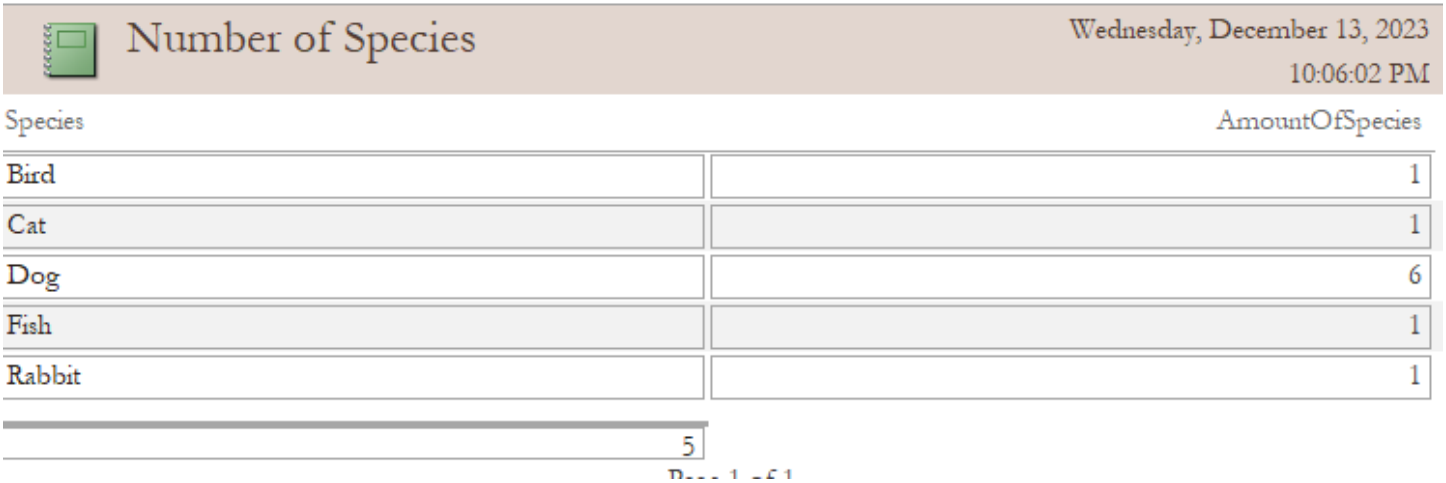
Page 1 of 1

3. This reports the incomplete tasks and their corresponding priority levels, ordered by the highest priority. Which allows our employees to easily determine which task they should complete next.

SQL CODE:

```
SELECT Task.TaskName AS Uncompleted_Task, Task.TaskDescription,
Task.Priority
FROM Task
```

```
WHERE (((Task.CompletionDate) Is Null))
ORDER BY Task.Priority DESC;
```



Number of Species

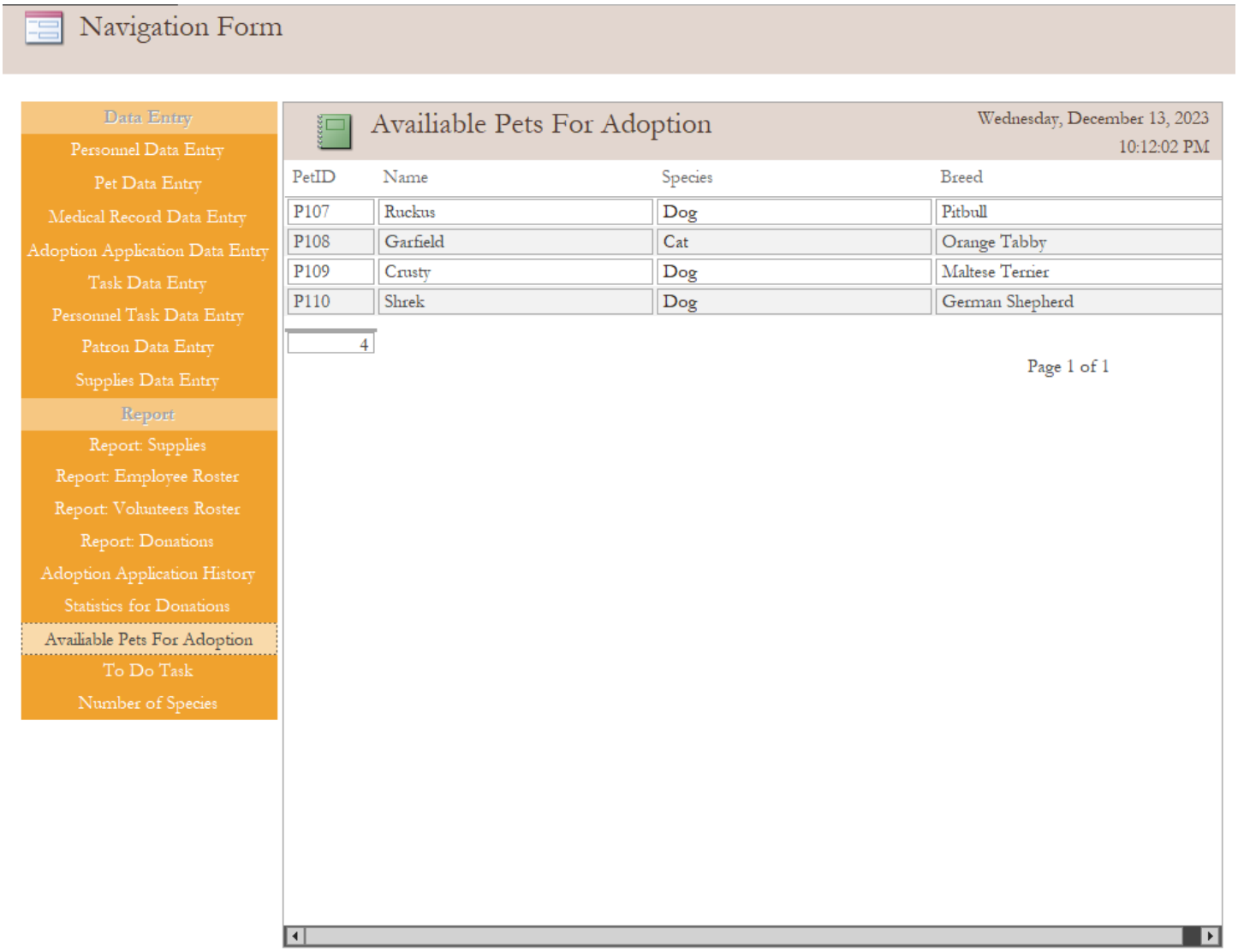
Wednesday, December 13, 2023
10:06:02 PM

Species	AmountOfSpecies
Bird	1
Cat	1
Dog	6
Fish	1
Rabbit	1
5	

4. This query is used to report the current number of each species in our shelter, in order to allow us to inform patrons whether the specific animal species they are interested in is currently available.

SQL CODE:

```
SELECT Species, COUNT(*) AS AmountOfSpecies
FROM Pet
GROUP BY Species;
```



Conclusion

Group 8 primarily used WhatsApp as a messaging platform to communicate in between meetings. For our meetings, we used Zoom as we are all accustomed to it. When it comes to the experience with the project, it was generally a good experience to how database systems work. The easier portion of the project was brainstorming for the proposal, it allowed us to express our creativity. The most difficult process was writing the SQL code to create the tables and inserting the information. This portion of the project took up to 3 hours during one of the meetings. From this group project, we learned alot about SQL and unexpectedly a good amount about MS Access. The MS Access was fun as we were able to explore the possibilities within the MS Access database software. Moving forwards, if I were to redo this project again, then I would have added a separate donation table, to separate it from the patron table, because not every patron would donate. I would also incorporate more lookup functions inside MS access, to reduce clicking back and forth to get ID numbers.