University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: Feb 19th, 2025

Group Number: 37

Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Jibek Dzhunusova	38422853	l4n8p	dzhunusova260503@gmail.com
Cathy Lei	12532537	i9q8h	cathy.lei2004@gmail.com
Leo Qu	17720962	x0a7w	leoqu2004@hotmail.com

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

In addition, we indicate that we are fully aware of the rules and consequences of plagiarism, as set forth by the Department of Computer Science and the University of British Columbia

2. Our project focuses on pet adoption and shelter management, providing an efficient system for tracking adoptions, fostering, donations, and medical care for rescued animals. Shelters can register pets, schedule vet check-ups, manage adoption applications, and link donations to funding needs.

3. See the last page for the revised ER diagram.
To improve readability, we have changed the name of common attributes to more specific terms that reflect the entities they belong to. For instance, instead of having just ‘name’, we now have ‘dName’ for the name of donors, ‘aName’ for the name of adopters, etc.

4.

```
pet(
        pid: char(20) PRIMARY KEY,
        species: char(20) not NULL,
        breed: char(20) not NULL,
        pName: char(20)
)

shelterPet(
        pid: char(20),
        PRIMARY KEY (pid),
        FOREIGN KEY (pid)
)

adoptedPet(
        pid: char(20)
        PRIMARY KEY (pid)
        FOREIGN KEY (pid)
)

fosterParent(
        fid: char(20) PRIMARY KEY,
        fName: char(20) not NULL,
        fAddress: char(20) not NULL,
        fPhoneNum: char(20) not NULL,
        experienceLevel: char(20)
)

fosterAssignment(
        faid: char(20) PRIMARY KEY,
        startDate: char[20] not NULL,
        endDate: char[20] not NULL
```

```
)

fosters(
        pid: char(20),
        fid: char(20),
        faid: char(20),
        PRIMARY KEY (faid),
        FOREIGN KEY (pid),
        FOREIGN KEY (fid),
        FOREIGN KEY (faid))

adopt(
        pid: char(20),
        aaid: char(20),
        aid: char(20),
        PRIMARY KEY (aaid),
        FOREIGN KEY (pid),
        UNIQUE (pid),
        FOREIGN KEY (aaid),
        FOREIGN KEY (aid))

shelter(
        shAddress: char[40] PRIMARY KEY
        sName: char[20])

staff(
        eid: char(20)  PRIMARY KEY,
         hourlyWage: integer,
         position: char[20])

adoptionApplication(
        aaid: char(20)  PRIMARY KEY
        adoptionFee: integer,
        adoptionDate: char[20]
)

donor(
        did: char(20)  PRIMARY KEY,
        dName: char[20],
        donorType: char(40))

adopter(
        aid: char(20)  PRIMARY KEY,
        aName: char[20],
        aAddress: char[40],
        aPhoneNum: integer,
        numofChildren: integer)

hasMedicalRecord(
        pid: char(20) FOREIGN KEY,
        medicalConditions: char(40),
        vaccineHistory: char(40),
        recordNum: char(20) FOREIGN KEY,
        PRIMARY KEY (pid, recordNum))

performedMedicalTest(
        vid: char(20) FOREIGN KEY,
        recordNum: char(20) FOREIGN KEY,
        PRIMARY KEY (vid, recordNum),
        testType: char(20),
        testResult: char(20)
)

worksAt(
        eid: char(20) FOREIGN KEY,
        shAddress: char(40) FOREIGN KEY,
        PRIMARY KEY (eid, shAddress))

vet(vid: char[20] PRIMARY KEY,
        vName: char[20],
        licenseExpiryDate: char[20])

donates(
        shAddress: char(40) FOREIGN KEY,
        did: char(20) FOREIGN KEY,
        PRIMARY KEY (shAddress, did),
        donationAmount: integer)

belongsTo(
        pid: char(20),
        shAddress: char(40),
        PRIMARY KEY (pid),
        FOREIGN KEY (pid),
        FOREIGN KEY (shAddress)
)
```

5.

FDs:
vid -> vName, licenceExpiryDate
pid -> pName, breed, species
breed -> species
pid, recordNum -> medicalConditions, vaccineHistory
recordNum, vid -> testType, testResult

eid -> hourlyWage, position
position -> hourlyWage

shAddress -> shName
shAddress, did -> donationAmount

did -> donorType, dName

fid -> fAddress, fPhoneNum, fName, experienceLevel

aaid -> adoptionFee, adoptionDate
aaid -> aid, pid

faid -> startDate, endDate
faid -> pid, fid

aid -> numOfChildren, aAddress, aPhoneNum, aName

6.

BCNF Normalization for Pet:
R(pid, species, breed, pname)
pid -> pName, breed, species
breed -> species

Closures:
pid+ = {pid, pName, breed, species}
breed+ = {breed, species}

Decompose on breed -> species
R1(breed, species), R2(pid, pname, breed)

```
R1(
        breed: char(20) PRIMARY KEY,
        species: char(20)
)

R2(
        pid: char(20) PRIMARY KEY,
        pName: char(20),
        breed: char(20)
)
```

BCNF Normalization for shelterPet and adoptedPet is the same as Pet

BCNF Normalization for fosterParent:
R(fid, fname, address, phoneNum, experienceLevel)
fid -> address, phoneNum, fname, experienceLevel

Closure:
fid+ = {fid, fname, address, phoneNum, experienceLevel} in BCNF
The keys and table didn’t change after normalization.

BCNF Normalization for fosterAssignment:
R(faid, startDate, endDate)
faid -> startDate, endDate

Closure:
faid+ = {faid, startDate, endDate} in BCNF
The keys and table didn’t change after normalization.

BCNF Normalization for fosters:
R(pid, fid, faid)
faid -> pid, fid

Closure: faid+ = {faid, pid, fid} in BCNF
The keys and table didn’t change after normalization.

BCNF Normalization for adopt:
R(pid, aaid, aid)
aaid -> aid, pid

Closure: aaid+ = {aaid, aid, pid} in BCNF
The keys and table didn’t change after normalization.

BCNF Normalization for Shelter:
R(shAddress, sname)
shAddress -> sname

Closure: address+ = {shAddress, sname} in BCNF

The keys and table didn't change after normalization.

BCNF Normalization for Staff:
R(eid, hourlyWage, position)
eid -> hourlyWage, position
position -> hourlyWage

Closures:
eid+ = {eid, hourlyWage, position} in BCNF
position+ = {position, hourlyWage} not in BCNF

Decompose on position -> hourlyWage
R1(position, hourlyWage), R2(position, eid)

```
R1(
        position: char(20) PRIMARY KEY,
        hourlyWage: integer
)

R2(
        eid: char(20) PRIMARY KEY,
        position: char(20)
)
```

BCNF Normalization for adoptionApplication:
R(aaid, adoptionFee, adoptionDate)
aaid -> adoptionFee, adoptionDate

Closures:
aaid+ = {aaid, adoptionFee, adoptionDate} in BCNF

The keys and table didn't change after normalization.
BCNF Normalization for Donor:
R(did, name, donorType)
did -> donorType, dName

closure:

did+ = {did, name, donorType}   in BCNF
The keys and table didn't change after normalization.

<u>BCNF Normalization for Adopter:</u>

R(aid, aName, aAddress, phoneNum, numOfChildren)
aid -> numOfChildren, aAddress, aPhoneNum, aName

Closure:
aid+ = {aid, aName, aAddress, aPhoneNum, numOfChildren} in BCNF
The keys and table didn't change after normalization.

<u>BCNF Normalization for hasMedicalRecord:</u>

```
hasMedicalRecord(
        pid: char(20) FOREIGN KEY,
        medicalConditions: char(40),
        vaccineHistory: char(40),
        recordNum: char(20) FOREIGN KEY,
        PRIMARY KEY (pid, recordNum)
)
```

R(pid, recordNum, medicalConditions, vaccineHistory)
pid, recordNum -> medicalConditions, vaccineHistory

closure:
{pid, recordNum}+ = {pid, recordNum, medicalConditions, vaccineHistory} in BCNF
The keys and table didn't change after normalization

<u>BCNF Normalization for performedMedicalTest:</u>
R(vid, recordNum, testType, testResult)
recordNum, vid -> testType, testResult

closure:
{recordNum, vid}+ = {recordNum, vid, testType, testResult} in BCNF
The keys and table didn't change after normalization.

<u>BCNF Normalization for worksAt:</u>
R(eid, shAddress)

Closure:
(eid, shAdress)+ = {eid, shAdress} in BCNF

The keys and table didn’t change after normalization.

BCNF Normalization for Vet:

R(vid, vName, licenseExpiryDate)
vid -> vName, licenseExpiryDate
closure:
vid+ = {vid, vName, licenseExpiryDate}

BCNF Normalization for donates:
R(shAddress, did, donationAmount)
shAddress, did -> donationAmount

closure:
{shAddress, did}+ = {shAddress, did, donationAmount} in BCNF

The keys and table didn’t change after normalization.

BCNF Normalization for belongsTo:
R(pid, shAddress)
No functional dependencies for this relation.

7.

```
CREATE TABLE Pet (
        pid CHAR(20) PRIMARY KEY,
        pName  CHAR(20) NOT NULL,
        breed CHAR(20),
        species CHAR(20),
        recordNum INTEGER,
        FOREIGN KEY (recordNum) REFERENCES Medical_Record(recordNum)
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE Shelter_Pet (
        pid CHAR(20),
        PRIMARY KEY (pid),
        FOREIGN KEY (pid) REFERENCES Pet(pid)
                ON DELETE CASCADE
                ON UPDATE CASCADE
)
```

```
CREATE TABLE Adopted_Pet (
        pid CHAR(20) PRIMARY KEY,
        aid CHAR(20),
        FOREIGN KEY (pid) REFERENCES Pet(pid)
                ON DELETE CASCADE
                ON UPDATE CASCADE,
        FOREIGN KEY (aid) REFERENCES Adopter(aid)
                ON DELETE CASCADE
                ON UPDATE CASCADE
)

CREATE TABLE fosterParent (
        fid CHAR(20) PRIMARY KEY,
        fName CHAR(20) NOT NULL,
        fAddress CHAR(20) NOT NULL,
        phoneNum CHAR(20) UNIQUE NOT NULL,
        experienceLevel CHAR(20)
)



CREATE TABLE fosterAssignment(
        faid CHAR(20) PRIMARY KEY,
        startDate CHAR(20) not NULL,
        endDate CHAR(20) not NULL
)

CREATE TABLE fosters(
        faid CHAR(20) PRIMARY KEY,
        pid CHAR(20),
        fid CHAR(20),
        FOREIGN KEY (faid) REFERENCES fosterAssignment(faid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (pid) REFERENCES ShelterPet(pid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (fid) REFERENCES fosterParent(fid)
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
)
```

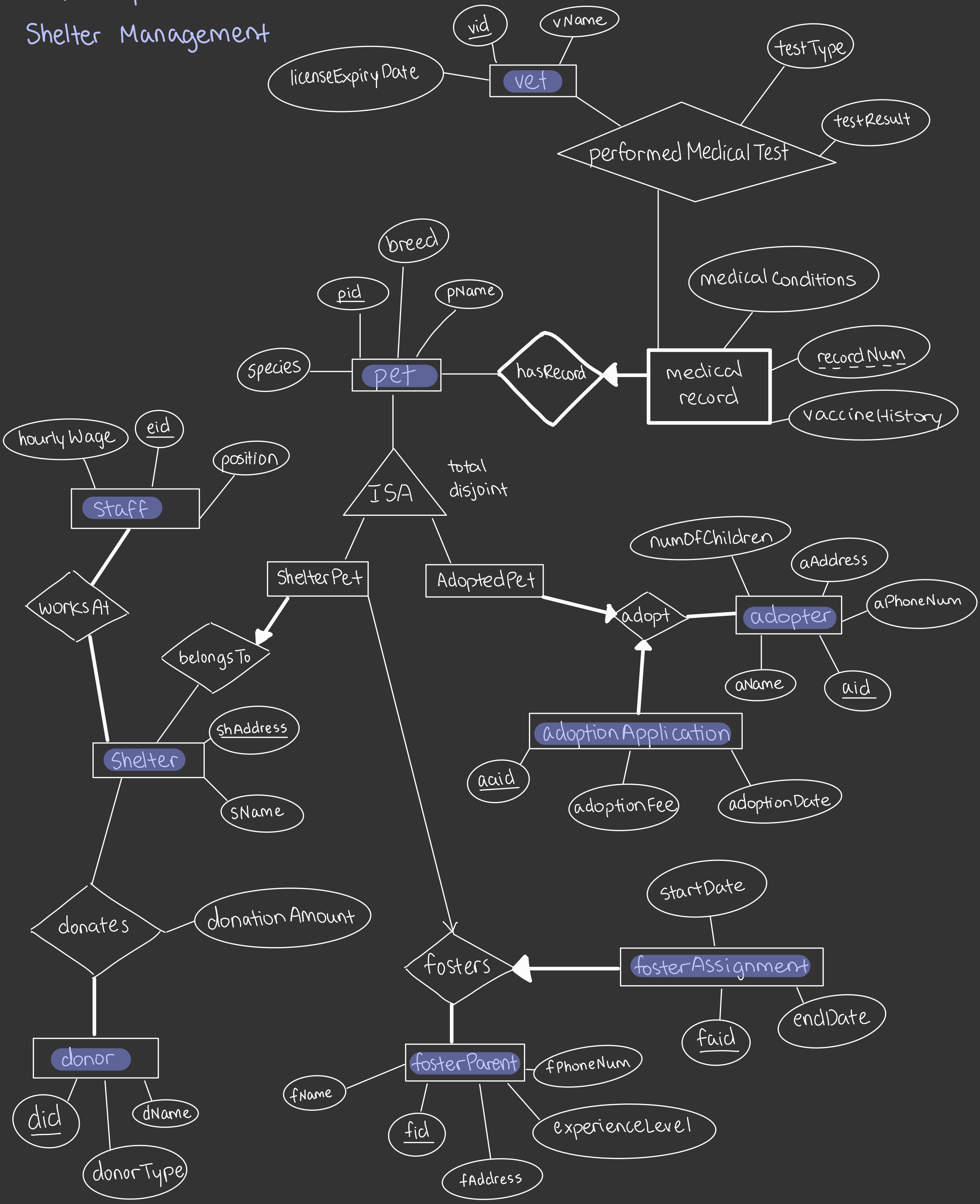
```
CREATE TABLE adopt(
        aaid CHAR(20) PRIMARY KEY,
        pid CHAR(20),
        aid CHAR(20),
        FOREIGN KEY (aaid) REFERENCES AdoptionApplication(aaid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (pid) REFERENCES AdoptedPet(pid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        UNIQUE (pid),
        FOREIGN KEY (aid) REFERENCES Adopter(aid)
                ON DELETE CASCADE,
                ON UPDATE CASCADE
)

CREATE TABLE Shelter (
        address CHAR(20) PRIMARY KEY,
        shName CHAR(20) not NULL
)


CREATE TABLE Staff (
        eid CHAR(20) PRIMARY KEY,
        hourlyWage INTEGER,
        position CHAR(20)
)

CREATE TABLE AdoptionApplication (
        aaid CHAR(20) PRIMARY KEY,
        adoptionFee INTEGER,
        adoptionDate CHAR(20)
)

CREATE TABLE Donor (
        did CHAR(20) PRIMARY KEY,
        dName CHAR(20),
        donationType CHAR(20)
)

CREATE TABLE Adopter (
        aid CHAR(20) PRIMARY KEY,
        aName CHAR(20) not NULL,
        address CHAR(20),
        phoneNum INTEGER not NULL,
        numOfChildren INTEGER,
        UNIQUE(phoneNum)
)

CREATE TABLE hasMedicalRecord (
        recordNum CHAR(20) PRIMARY KEY,
        medicalConditions CHAR(40),
        vaccineHistory CHAR(40)
)

CREATE TABLE performedMedicalTest(
        vid CHAR(20),
        recordNum CHAR(20),
        testType CHAR(20),
        testResult CHAR(20),
        PRIMARY KEY (vid, recordNum),
        FOREIGN KEY (vid) REFERENCES Vet(vid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (recordNum) REFERENCES Medical_Record(recordNum),
                ON DELETE CASCADE,
                ON UPDATE CASCADE
)


CREATE TABLE WorksAt (
        eid CHAR(20),
        shAddress CHAR(20),
        PRIMARY KEY (eid, shAddress)
        FOREIGN KEY (eid) REFERENCES Staff(eid),
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (shAdress) REFERENCES Shelter(sid)
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
)

CREATE TABLE Vet (
        vid CHAR(20) PRIMARY KEY,
        vName CHAR(20) not NULL,
        licenseExpiryDate CHAR(20) not NULL
)

CREATE TABLE Donates (
        shAddress CHAR(20),
        did CHAR(20),
        donationAmount INTEGER,
        PRIMARY KEY (shAddress, did),
        FOREIGN KEY (shAddress) REFERENCES Shelter(shAddress)
                ON DELETE CASCADE,
                ON UPDATE CASCADE,
        FOREIGN KEY (did) REFERENCES Donor(did)
                ON DELETE CASCADE
                ON UPDATE CASCADE
)


CREATE TABLE belongsTo (
        pid CHAR(20),
        shAddress CHAR(20)
)
```

8.

```
INSERT INTO Pet (pid, pName, breed, species, recordNum)
VALUES (‘SP1000’, ‘Buddy’, ‘Golden Retriever’, ‘Dog’, ‘1’),
         (‘SP1001’, ‘Cleo’, ‘Maine Coon’, ‘Cat’, ‘2’),
        (‘SP1002’, ‘Princess’, ‘Poodle’, ‘Dog’, ‘3’),
        (‘SP1003’, ‘Feathers’, ‘Parrot’, ‘Bird’, ‘4’),
        (‘SP1004’, ‘Alice’, ‘Husky’, ‘Dog’, ‘5’),
        (‘AP1000’, ‘Max’, ‘British Shorthair’, ‘Cat’, ‘6’),
        (‘AP1001’, ‘Mia’, ‘Corgi’, ‘Dog’, ‘7’),
        (‘AP1002’, ‘Angel’, ‘Dwarf Campbell Hamster’, ‘Hamster’, ‘8’),
        (‘AP1003’, ‘Teddy’, ‘Teddy Guinea Pig’, ‘Guinea Pig’, ‘9’)
        (‘AP1004’, ‘Misty’, ‘Painted Turtle’, ‘Turtle’, ‘10’);



INSERT INTO Shelter_Pet(pid)
VALUES (‘SP1000’),
         (‘SP1001’),
         (‘SP1002’),
         (‘SP1003’),
         (‘SP1004’);

INSERT INTO Adopted_Pet(pid)
VALUES (‘AP1000’),
         (‘AP1001’),
         (‘AP1002’),
         (‘AP1003’),
         (‘AP1004’);

INSERT INTO fosterParent(fid, fName, fAddress, phoneNum, experienceLevel)
VALUES (‘FP1’, ‘Ash’, ‘1234 Pallet Town’, ‘778 109 1480’, ‘Beginner’),
         (‘FP2’, ‘Dawn’, ‘5801 Twinleaf Town’, ‘604 103 5910’, ‘Beginner’),
         (‘FP3’, ‘Brock’, ‘6803 Pewter City’, ‘802 684 0023’, ‘Expert’),
         (‘FP4’, ‘Serena’, ‘5123 Vaniville Town’, ‘556 402 4950’, ‘Intermediate’),
         (‘FP5’, ‘Clement’, ‘145 0395 Lumiose City’, ‘019 589 1289’, ‘Expert’);

INSERT INTO fosterAssignment(faid, startDate, endDate)
VALUES (‘FA1’, ‘2021-12-22’, ‘2022-01-01’),
(‘FA2’, ‘2021-12-12’, ‘2022-03-01’),
(‘FA3’, ‘2023-01-31’, ‘2023-05-01’),
(‘FA4’, ‘2024-02-14’, ‘2024-02-28’),
(‘FA5’, ‘2025-10-01’, ‘2023-12-29’);


INSERT INTO fosters(faid, pid, fid)
VALUES (‘FA1’, ‘SP1000’, ‘FP1’),
(‘FA2’, ‘SP1001’, ‘FP2’),
(‘FA3’, ‘SP1002’, ‘FP3’),
(‘FA4’, ‘SP1003’, ‘FP4’),
(‘FA5’, ‘SP1004’, ‘FP5’);

INSERT INTO adopt(aaid, pid, aid)
VALUES (‘AA1’, ‘AP1000’, ‘12345’),
        (‘AA2’, ‘AP1001’, ‘32594’),
        (‘AA3’, ‘AP1002’, ‘65894’),
        (‘AA4’, ‘AP1003’, ‘64892’),
        (‘AA5’, ‘AP1004’, ‘54681’);


INSERT INTO Shelter(address, shName)
VALUES (‘5903 Vermillion City’, ‘Happy Furry Friends’),
         (‘9054 Canalave City’, ‘Angry Spiky Enemies’),
          (‘6965 Shalour City’, ‘Mild Hairless Animals’),
          (‘2252 Snowpoint City’, ‘Icy Soft Beasts’),
          (‘0059 Oreburgh City’, ‘Fiery Rough Cuties ’);

INSERT INTO Staff(eid, hourlyWage, position)
VALUES (‘S1’, ‘25’, ‘Receptionist’),
          (‘S2’, ‘32’, ‘Animal Shelter Welfare Coordinator’),
          (‘S3’, ‘32’, ‘Animal Care Attendant’),
          (‘S4’, ‘34’, ‘Animal Shelter Welfare Coordinator’),
          (‘S5’, ‘32’, ‘Shelter Attendant’);

INSERT INTO AdoptionApplication(aaid, adoptionFee, adoptionDate)
VALUES (‘AA1’, 300, ‘2021-10-01’),
          (‘AA2’, 250, ‘2021-04-22’),
          (‘AA3’, 100, ‘2019-03-13’),
          (‘AA4’, 125, ‘2020-12-01’),
          (‘AA5’, 200, ‘2025-01-12’);

INSERT INTO Donor(did, dName, donationType)
VALUES (‘12345’, ‘Raymond’, ‘individual’),
          (‘45678’, ‘Jibek’, ‘individual’),
          (‘98745’, ‘Cathy’, ‘individual’),
          (‘65412’, ‘Leo’, ‘individual’),
          (‘12396’, ‘UBC’, ‘organization’);

INSERT INTO Adopter(aid, aName, address, phoneNum, numOfChildren)
VALUES (‘12354’, ‘Seva’, ‘1236 University Blvd’, 123684569, 1),
          (‘32594’, ‘Gregor’, ‘3853 Granville St’, 186492345, 3),
          (‘65894’, ‘Jordon’, ‘1234 West Mall’, 15569878, 1),
          (‘64892’, ‘Paul’, ‘6543 Lower Mall’, 12369656532),
          (‘54681’, ‘Andy’, ‘4738 East Mall’, 12368846325);

INSERT INTO hasMedicalRecord(recordNum, medicalConditions, vaccineHistory)
VALUES (‘12345’, ‘no medical conditions’, ‘vaccinated’),
          (‘65241, ‘influenza’, ‘vaccinated’),
          (‘46585’, ‘Arthritis’, ‘not vaccinated’),
          (‘88439’, ‘Hypothyroidism’, ‘not vaccinated’),
          (‘49832’, ‘Cancer’, ‘vaccinated’);

INSERT INTO performedMedicalTest(vid, recordNum, testType, testResult)
VALUES (‘42369’, ‘12345’, ‘blood test’, ‘within normal range’),
          (‘13954’, ‘65894’, ‘blood test’, ‘within normal range’),
          (‘15934’, ‘54681’, ‘X-ray’, ‘tumour’),
          (‘36945’, ‘46585’, ‘X-ray’, ‘swallowed foreign object’)
          (‘89543’, ‘45785’, ‘CT scan’, ‘further testing required’);

INSERT INTO WorksAt(eid, shAddress)
VALUES (‘12345’, ‘3695 West Broadway’),
          (‘69845’, ‘6245 Sasamat St’),
          (‘59436’, ‘2364 Burrard St’),
          (‘68423’, ‘4869 West Georgia’),
          (‘36492’, ‘3648 W 16th Avenue’);

INSERT INTO Vet(vid, vName, licenseExpiryDate)
VALUES (‘V1345’, ‘Abby’, ‘2025-03-12’),
          (‘V1345’, ‘Michael’, ‘2027-12-31’),
          (‘V1928’, ‘Anna’, ‘2026-04-12’),
          (‘V2837’, ‘Sam’, ‘2025-06-23’),
          (‘V8273’, ‘Jim’, ‘2027-01-02’);

INSERT INTO Donates(shAddress, did, donationAmount)
VALUES (‘3695 West Broadway’, ‘56328’, 1000),
          (‘6245 Sasamat S’, ‘56328’, 1000),
          (‘2364 Burrard St’, ‘65831’, 100),
          (‘4869 West Georgia’, ‘59628’, 20000),
          (‘3648 W 16th Avenue’, ‘36458’, 9050);


INSERT INTO belongsTo(pid, shAddress)
VALUES (‘SP1000’, ‘5903 Vermillion City’),
           (‘SP1001’, ‘2252 Snowpoint City’),
           (‘SP1002’, ‘2252 Snowpoint City’),
           (‘AP1000’, ‘0059 Oreburgh City’),
           (‘AP1001’, ‘6965 Shalour City’);
```

9. We confirm we did not use AI tools in this assignment.

Pet Adoption & Shelter Management


vid
vName
vet
licenseExpiryDate
testType
testResult
performed Medical Test
breed
pid
pName
species
pet
hasRecord
medical conditions
medical record
record Num
vaccineHistory
hourly Wage
eid
position
Staff
ISA
total
disjoint
works At
ShelterPet
AdoptedPet
belongsTo
numOfChildren
aAddress
aPhoneNum
adopt
adopter
aName
aid
shAddress
Shelter
sName
adoptionApplication
acid
adoptionFee
adoptionDate
donates
donationAmount
startDate
fosters
fosterAssignment
faid
endDate
donor
did
dName
donorType
fosterParent
fPhoneNum
fName
fid
experienceLevel
fAddress