University of British Columbia, Vancouver
Department of Computer Science

CPSC 304 Project Cover Page

Milestone #: 2

Date: October 9, 2024

Group Number: 72

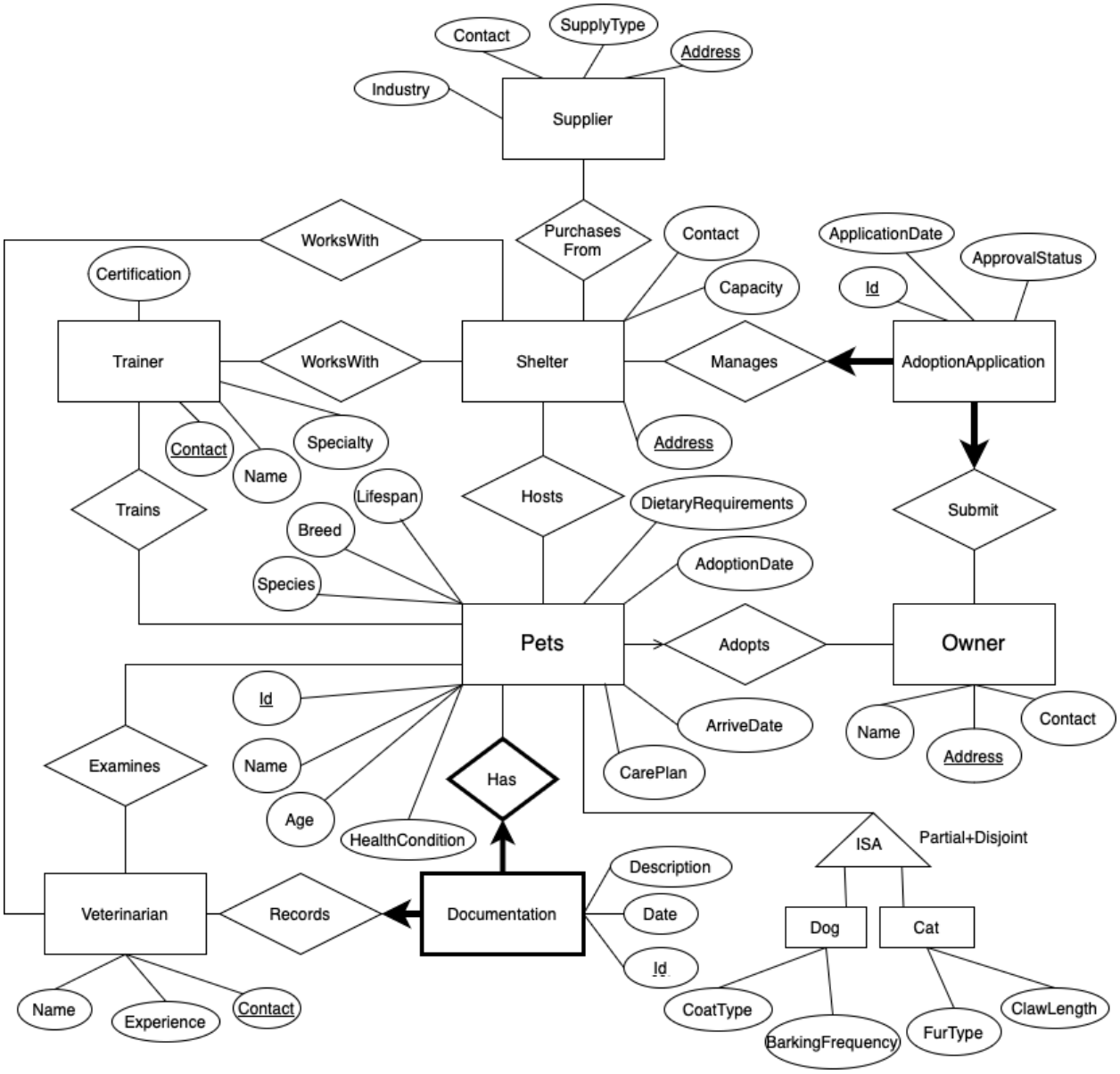
Name	Student Number	CS Alias (Userid)	Preferred E-mail Address
Edward Liu	55997308	e9s4m	edwardtliu8@gmail.com
Kobe Shen	13079694	b0j3y	Shenkobe.111@gmail.com
Yang Yu	45834330	n5p8x	yuyang2003m@163.com

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

Project Summary:

Our project aims to model the whole pet adoption and post-adoption animal care ecosystem, allowing users to browse and adopt pets hosted at the shelter, allowing animal specialists like veterinarians to view, store and update a pet's medical documentation and for trainers to work with shelters and train their respective pets.



List of Changes from Milestone #1:

- Incorporating the TA's suggestion, the **Pet-Owner relationship** was changed from many-to-many to **many-to-one**, as an owner can adopt multiple pets but a pet can only have one adoption date.

- Incorporating the TA's suggestion, the **primary keys** for Shelter, Owner, and Supplier were changed from Id to **address**, and for Trainer and Veterinarian to **contact**, using already existing keys instead of adding new Id attributes.
- The following additional attributes were added to provide more detailed relevant information to each entity:
  Trainer: certification, contact
  Supplier: industry
  Pet: dietaryRequirements, carePlan, lifeSpan, breed, species

Relational Schema

Underlined: Primary Key
Italicized: Candidate Key
Bolded: Foreign Key

Trainer(Contact: varchar, Name: varchar, Specialty: varchar, Certification: varchar)
Shelter(Address: varchar, Capacity: Integer, *Contact*: varchar)
Owner(Address: varchar, Name: varchar, *Contact*: varchar)
Veterinarian(Contact: varchar, Name: varchar, Experience: varchar)
Supplier(Address: varchar, SupplyType: varchar, *Contact*: varchar, Industry: varchar)
Pets(**OwnerAddress**: varchar, Id: Integer, Name: varchar, Age: Integer, HealthCondition: varchar, AdoptionDate: date, ArriveDate: date, Species: varchar, Breed: varchar, Lifespan: Integer, DietaryRequirments: varchar, CarePlan: varchar)

Many-to-One Participation
AdoptionApplication(**ShelterAddress**: varchar**, OwnerAddress**: varchar**,** Id: Integer, ApplicationDate: date, ApprovalStatus: Bool) (ShelterAddress and OwnerAddress not null)

Weak Entity
Documentation(**PetsId**: Integer, **VeterinarianContact**: varchar, Id: Integer, Description: varchar, Date: date) (VeterinanrianContact not null)

Isa
Dog(**PetsId**: Integer, CoatType: varchar, BarkingFrequency: Integer)
Cat(**PetsId**: Integer, FurType: varchar, ClawLength: Integer)

Many-to-Many Relationship
PurchasesFrom(**SupplierAddress**: varchar, **ShelterAddress**: varchar)
Trains(**TrainerContact**: varchar, **PetsId**: Integer)
Examines(**VeterinarianContact**: varchar, **PetsId**: Integer)
VetWorksWithShel(**VeterinarianContact**: varchar, **ShelterAddress**: varchar)
TrainWorksWithShel(**TrainerContact**: varchar, **ShelterAddress**: varchar)
Hosts(**PetId**: Integer, **ShelterAddress**: varchar)

PK relations

Trainer:
Contact -> Name, Specialty, Certification

Shelter:
Address -> Capacity, Contact

AdoptionApplication:
Id -> ShelterAddress, Owner Address, ApplicationDate, Approval Status

Owner:
Address -> Name, Contact

Veterinarian:
Contact -> Name, Experience

Supplier:
Address -> SupplyType, Contact, Industry

Documentation:
Id -> PetsId, VeterinarianContact, Description, Date

Pet:
Id -> OwnerAddress, Name, Age, HealthCondition, AdoptionDate, ArriveDate, Species, Breed, Lifespan, DietaryRequirements, CarePlan

Dog:
PetsId -> CoatType, BarkingFrequency

Cat:
PetsId -> FurType, ClawLength

CK relations

Shelter:
Contact -> Capacity, Address

Owner:
Contact -> Name, Address

Supplier:
Contact -> SupplyType, Address, Industry

Non-PK/CK relations:

Trainer:
Certification -> Specialty

Supplier:
Industry -> SupplyType

Pet:
Species, Age -> DietaryRequirements
Species, Breed -> LifeSpan
Species, DietaryRequirements, HealthCondition -> CarePlan

Normalization

Underlined: Primary Key
Italicized: Candidate Key
Bolded: Foreign Key

Trainer:

Trainer(Contact: varchar, Name: varchar, Specialty: varchar, Certification: varchar)
Contact -> Name, Specialty, Certification
Certification -> Specialty

Certification -> Specialty fails 3NF

Find minimal cover:

Step 1 - Putting FDs in standard form:
Contact -> Name, Specialty, Certification
Certification -> Specialty
=
Contact -> Name
Contact -> Specialty
Contact -> Certification
Certification -> Specialty

Step 2 - Minimize LHS of each FD:
Each FD only has 1 attribute on LHS, thus already minimized

Step 3 - Delete Redundant FDs:
Contact -> Specialty is redundant
=
Contact -> Name

Contact -> Certification
Certification -> Specialty

Using synthesis
Trainer1(Contact: varchar, Name: varchar),
Trainer2(**Contact:** varchar, **Certification**: varchar),
Trainer3(Certification: varchar, Specialty: varchar)

Shelter:

Shelter(Address: varchar, Capacity: Integer, *Contact*: varchar)
Address -> Capacity, Contact
Contact -> Capacity, Address

Already in normalized form, as both FDs pass 3NF

AdoptionApplication:

AdoptionApplication(**ShelterAddress**: varchar**, OwnerAddress**: varchar**,** Id: Integer,
ApplicationDate: date, ApprovalStatus: Bool)
Id -> ShelterAddress, Owner Address, ApplicationDate, Approval Status

Already in normalized form, as both FDs pass 3NF

Owner:

Owner(Address: varchar, Name: varchar, *Contact*: varchar)
Address -> Name, Contact
Contact -> Name, Address

Already in normalized form, as FDs passes 3NF

Veterinarian:

Veterinarian(Contact: varchar, Name: varchar, Experience: varchar)
Contact -> Name, Experience

Already in normalized form, as FD passes 3NF

Supplier:

Supplier(Address: varchar, SupplyType: varchar, *Contact*: varchar, Industry: varchar)
Address -> SupplyType, SupplierContact, Industry
Contact -> SupplyType, Address, Industry

Industry -> SupplyType

Industry -> SupplyType fails 3NF

Find minimal cover:

Step 1 - Putting FDs in standard form:
Address -> SupplyType, Contact, Industry
Contact -> SupplyType, Address, Industry
Industry -> SupplyType
=
Address -> SupplyType
Address -> Contact
Address -> Industry
Contact -> SupplyType
Contact -> Address
Contact -> Industry
Industry -> SupplyType

Step 2 - Minimize LHS of each FD:
Each FD only has 1 attribute on LHS, thus already minimized

Step 3 - Delete Redundant FDs:
Address -> SupplyType is redundant
=
Address -> Contact
Address -> Industry
Contact -> SupplyType
Contact -> Address
Contact -> Industry
Industry -> SupplyType

Contact -> SupplyType is redundant
=
Address -> Contact
Address -> Industry
Contact -> Address
Contact -> Industry
Industry -> SupplyType

Contact -> Industry is redundant
=
Address -> Contact
Address -> Industry

Contact -> Address
Industry -> SupplyType

Using synthesis
Supplier1(Address, *Contact*),
Supplier2(**Address**, **Industry**),
Supplier3(Industry, SupplyType)

Documentation:

Documentation(**PetsId**: Integer, **VeterinarianContact**: varchar, Id: Integer, Description: varchar, Date: date)
Id -> PetsId, VeterinarianContact, Description, Date

Already in normalized form, as FD passes 3NF

Pets:

Pets(**OwnerAddress**: varchar, Id: Integer, Name: varchar, Age: Integer, HealthCondition: varchar, AdoptionDate: date, ArriveDate: date, Species: varchar, Breed: varchar, Lifespan: Integer, DietaryRequirments: varchar, CarePlan: varchar)

Id -> Name, Age, HealthCondition, AdoptionDate, ArriveDate, Species, Breed, Lifespan, DietaryRequirements, CarePlan, OwnerAddress
Species, Age -> DietaryRequirements
Species, Breed -> LifeSpan
Species, DietaryRequirements, HealthCondition -> CarePlan

Species, Age -> DietaryRequirements, Species, Breed -> LifeSpan, Species, DietaryRequirements, HealthCondition -> CarePlan fails 3NF

Find minimal cover:

Step 1 - Putting FDs in standard form:
Id -> Name, Age, HealthCondition, AdoptionDate, ArriveDate, Species, Breed, Lifespan, DietaryRequirements, CarePlan
Species, Age -> DietaryRequirements
Species, Breed -> LifeSpan
Species, DietaryRequirements, HealthCondition -> CarePlan
=
Id -> Name
Id -> Age
Id -> HealthCondition
Id -> AdoptionDate

Id -> ArriveDate
Id -> Species
Id -> Breed
Id -> Lifespan
Id -> DietaryRequirements
Id -> CarePlan
Id -> OwnerAddress
Species, Age -> DietaryRequirements
Species, Breed -> LifeSpan
Species, DietaryRequirements, HealthCondition -> CarePlan

Step 2 - Minimize LHS of each FD:
Each FD only has 1 attribute on LHS, thus already minimized

Step 3 - Delete Redundant FDs:
Id -> DietaryRequirements, Id -> Lifespan, Id-> CarePlan are redundant
=
Id -> PetsName
Id -> Age
Id -> HealthCondition
Id -> AdoptionDate
Id -> ArriveDate
Id -> Species
Id -> Breed
Id -> OwnerAddress
Species, Age -> DietaryRequirements
Species, Breed -> LifeSpan
Species, DietaryRequirements, HealthCondition -> CarePlan

Using synthesis:
Pet1(Id, Name),
Pet2(**Id**, **Age**),
Pet3(**Id**, HealthCondition),
Pet4(**Id**, AdoptionDate),
Pet5(**Id**, ArriveDate),
Pet6(**Id**, **Species**),
Pet7(**Id**, **Breed**),
Pet8(**Id, OwnerAddress**),
Pet9(Species, Age, **DietaryRequirements**),
Pet10(**Species**, Breed, LifeSpan),
Pet11(**Species**, DietaryRequirements, HealthCondition, CarePlan)

Dog:

Dog(**PetsId**: Integer, CoatType: varchar, BarkingFrequency: Integer)
PetsId -> CoatType, BarkingFrequency

Already in normalized form, as FD passes 3NF

Cat:

Cat(**PetsId**: Integer, FurType: varchar, ClawLength: Integer)
PetsId -> FurType, ClawLength

Already in normalized form, as FD passes 3NF

The following tables do not have any FDs, thus are already normalized:

PurchasesFrom(**SupplierAddress**: varchar, **ShelterAddress**: varchar)
Trains(**TrainerContact**: varchar, **PetsId**: Integer)
Examines(**VeterinarianContact**: varchar, **PetsId**: Integer)
VecWorksWithShel(**VeterinarianContact**: varchar, **ShelterAddress**: varchar)
TrainWorksWithShel(**TrainerContact**: varchar, **ShelterAddress**: varchar)
Hosts(**PetId**: Integer, **ShelterAddress**: varchar)

SQL DDL

Trainer:

```
CREATE TABLE Trainer1 (
        Contact         VARCHAR       PRIMARY KEY,
        Name            VARCHAR
);

CREATE TABLE Trainer2 (
        Contact         VARCHAR       PRIMARY KEY,
        Certification   VARCHAR,
        FOREIGN KEY (Contact) REFERENCES Trainer1(Contact),
        FOREIGN KEY (Certification) REFERENCES Trainer3(Certification)
);

CREATE TABLE Trainer3 (
        Certification   VARCHAR       PRIMARY KEY,
        Specialty       VARCHAR
);
```

Shelter:

```
CREATE TABLE Shelter (
        Address         VARCHAR         PRIMARY KEY,
        Capacity        INTEGER,
        Contact         VARCHAR,
        UNIQUE (Contact)
);
```

Owner:

```
CREATE TABLE Owner(
        Address         VARCHAR         PRIMARY KEY,
        Name            VARCHAR,
        Contact         VARCHAR,
        UNIQUE (Contact)
);
```

Veterinarian:

```
CREATE TABLE Veterinarian(
        Contact         VARCHAR         PRIMARY KEY,
        Name            VARCHAR,
        Experience      VARCHAR
);
```

Supplier:

```
CREATE TABLE Supplier1 (
        Address         VARCHAR         PRIMARY KEY,
        Contact         VARCHAR,
        UNIQUE (Contact)
);

CREATE TABLE Supplier2 (
        Address         VARCHAR         PRIMARY KEY,
        Industry        VARCHAR,
        FOREIGN KEY (Address) REFERENCES Supplier1(Address),
        FOREIGN KEY (Industry) REFERENCES Supplier3(Industry)
);

CREATE TABLE Supplier3 (
        Industry        VARCHAR         PRIMARY KEY,
        SupplyType      VARCHAR
);
```

Pet:

```
CREATE TABLE Pet1 (
        Id              INTEGER         PRIMARY KEY,
        Name            VARCHAR
);

CREATE TABLE Pet2 (
        Id      INTEGER         PRIMARY KEY,
        Age     INTEGER
        FOREIGN KEY (Id) REFERENCES Pet1(Id),
        FOREIGN KEY (Age) REFERENCES Pet9(Age)
);

CREATE TABLE Pet3 (
        Id                      INTEGER         PRIMARY KEY,
        HealthCondition         VARCHAR,
        FOREIGN KEY (Id) REFERENCES Pet1(Id)
);

CREATE TABLE Pet4 (
        Id                      INTEGER         PRIMARY KEY,
        AdoptionDate            DATE,
        FOREIGN KEY (Id) REFERENCES Pet1(Id)
);

CREATE TABLE Pet5 (
        Id                      INTEGER         PRIMARY KEY,
        ArriveDate              DATE,
        FOREIGN KEY (Id) REFERENCES Pet1(Id)
);

CREATE TABLE Pet6 (
        Id                      INTEGER         PRIMARY KEY,
        Species                 VARCHAR,
        FOREIGN KEY (Id) REFERENCES Pet1(Id),
        FOREIGN KEY (Species) REFERENCES Pet9(Species)
);

CREATE TABLE Pet7 (
        Id                      INTEGER         PRIMARY KEY,
        Breed                   VARCHAR,
        FOREIGN KEY (Id) REFERENCES Pet1(Id),
        FOREIGN KEY (age) REFERENCES Pet10(Breed)
```

```
);

CREATE TABLE Pet8 (
        Id                      INTEGER         PRIMARY KEY,
        OwnerAddress            VARCHAR,
        FOREIGN KEY (Id) REFERENCES Pet1(Id),
        FOREIGN KEY (OwnerAddress) References Owner(Address)
);

CREATE TABLE Pet9 (
        Species                         VARCHAR,
        Age                             INTEGER,
        DietaryRequirements             VARCHAR,
        PRIMARY KEY (Species, Age),
        FOREIGN KEY (DietaryRequirements) REFERENCES Pet11(DietaryRequirements)
);

CREATE TABLE Pet10 (
        Species                         VARCHAR,
        Breed                           VARCHAR,
        Lifespan                        INTEGER,
        PRIMARY KEY (Species, Breed),
        FOREIGN KEY (Species) REFERENCES Pet9(Species)
);

CREATE TABLE Pet11 (
        Species                         VARCHAR,
        DietaryRequirements             VARCHAR,
        HealthCondition                 VARCHAR,
        CarePlan                        VARCHAR,
        PRIMARY KEY (Species, DietaryRequirements, HealthCondition),
        FOREIGN KEY (Species) REFERENCES Pet9(Species)
);
```

AdoptionApplication:

```
CREATE TABLE AdoptionApplication (
        ShelterAddress      VARCHAR     NOT NULL,
        OwnerAddress        VARCHAR     NOT NULL,
        Id                  INTEGER     PRIMARY KEY,
        ApplicationDate     DATE,
        ApprovalStatus      BOOLEAN,
```

```
    FOREIGN KEY (ShelterAddress) REFERENCES Shelter(Address),
    FOREIGN KEY (OwnerAddress) REFERENCES Owner(Address),
);
```

Documentation:

```
CREATE TABLE Documentation (
    PetsId              INTEGER,
    VeterinarianContact VARCHAR,
    Id                  INTEGER,
    Description         VARCHAR,
    Date                DATE,
    PRIMARY KEY (PetsId, Id),
    FOREIGN KEY (PetsId) REFERENCES Pet1(Id),
    FOREIGN KEY (VeterinarianContact) REFERENCES Veterinarian(Contact)
);
```

We are aware that the foreign key PetsId should be “on update cascade”, due to the weak entity relationship, however, we will be using Oracle in our project, which does not support it.

Dog:

```
CREATE TABLE Dog (
    PetsId              INTEGER,
    CoatType            VARCHAR,
    BarkingFrequency    INTEGER,
    PRIMARY KEY (PetsId),
    FOREIGN KEY (PetsId) REFERENCES Pet1(Id)
);
```

Cat:

```
CREATE TABLE Cat (
    PetsId              INTEGER,
    FurType             VARCHAR,
    ClawLength          INTEGER,
    PRIMARY KEY (PetsId),
    FOREIGN KEY (PetsId) REFERENCES Pet1(Id)
)
```

Many-to-Many Relationships:

```
CREATE TABLE PurchasesFrom (
```

```
        SupplierAddress     VARCHAR,
        ShelterAddress      VARCHAR,
        PRIMARY KEY (SupplierAddress, ShelterAddress),
        FOREIGN KEY (SupplierAddress) REFERENCES Supplier1(Address),
        FOREIGN KEY (ShelterAddress) REFERENCES Shelter(Address)
);

CREATE TABLE Trains (
        TrainerContact      VARCHAR,
        PetsId              INTEGER,
        PRIMARY KEY (TrainerContact, PetsId),
        FOREIGN KEY (TrainerContact) REFERENCES Trainer1(Contact),
        FOREIGN KEY (PetsId) REFERENCES Pet1(Id)
);

CREATE TABLE Examines (
        VeterinarianContact VARCHAR,
        PetsId              INTEGER,
        PRIMARY KEY (VeterinarianContact, PetsId),
        FOREIGN KEY (VeterinarianContact) REFERENCES Veterinarian(Contact),
        FOREIGN KEY (PetsId) REFERENCES Pet1(Id)
);

CREATE TABLE VetWorksWithShel (
        VeterinarianContact VARCHAR,
        ShelterAddress      VARCHAR,
        PRIMARY KEY (VeterinarianContact, ShelterAddress),
        FOREIGN KEY (VeterinarianContact) REFERENCES Veterinarian(Contact),
        FOREIGN KEY (ShelterAddress) REFERENCES Shelter(Address)
);

CREATE TABLE TrainWorksWithShel (
        TrainerContact      VARCHAR,
        ShelterAddress      VARCHAR,
        PRIMARY KEY (TrainerContact, ShelterAddress),
        FOREIGN KEY (TrainerContact) REFERENCES Trainer1(Contact),
        FOREIGN KEY (ShelterAddress) REFERENCES Shelter(Address)
);
CREATE TABLE Hosts (
        PetsId              INTEGER,
        ShelterAddress      VARCHAR,
        PRIMARY KEY (PetsId, ShelterAddress),
        FOREIGN KEY (PetsId) REFERENCES Pet1(Id)
        FOREIGN KEY (ShelterAddress) REFERENCES Shelter(Address)
```

);

Populate tables

```
INSERT INTO Trainer1        VALUES ('778-111-1111', 'Bob'),
                                   ('778-111-1112', 'Rob'),
                                   ('778-111-1113', 'Steve'),
                                   ('778-111-1114', 'Matt'),
                                   ('778-111-1115', 'Julie');

INSERT INTO Trainer2        VALUES ('778-111-1111', 'Certified Professional Dog Trainer'),
                                   ('778-111-1112', 'Certified Cat Behavior Consultant'),
                                   ('778-111-1113', 'Certified Dog Behavior Consultant'),
                                   ('778-111-1114', 'PetSmart Dog Trainer Certification'),
                                   ('778-111-1115', 'Feline Training Certification');

INSERT INTO Trainer3        VALUES ('Certified Professional Dog Trainer', 'Dogs'),
                                   ('Certified Cat Behavior Consultant', 'Cats'),
                                   ('Certified Dog Behavior Consultant', 'Dogs'),
                                   ('PetSmart Dog Trainer Certification', 'Cats'),
                                   ('Feline Training Certification', 'Cats');

INSERT INTO Shelter         VALUES ('100 Fraser St', 20, '778-395-3495')
                                   ('54 Robson St', 34, '604-333-2322'),
                                   ('74 Granville St', 50, '245-345-6432'),
                                   ('76 Nanaimo St', 45, '246-345-2424'),
                                   ('16 Main St', 80, '778-242-2422');

INSERT INTO Owner           VALUES('1234 Main St', 'John', '778-155-6186'),
                                  ('5678 Fraser St', 'Chris', '778-156-6289'),
                                   ('2198 Dunbar St', 'Sam', '778-769-1875'),
                                   ('1570 Cambie St', 'Mary', '604-178-1785'),
                                   ('3916 Knight St', 'Sarah', '604-333-9617');

INSERT INTO Veterinarian    VALUES('111-111-1323', 'Callum', 'Certified Veterinarian
Practitioner'),
                                   ('234-245-2311', 'Dean', 'Certified Veterinarian
Practitioner'),
                                   ('604-323-1212', 'Ken', 'Certified Veterinarian
Practitioner'),
                                   ('999-233-3232', 'Alison', 'Certified Veterinarian
Practitioner'),
                                   ('456-234-1242', 'Hailey', 'Certified Veterinarian
Practitioner');
```

```
INSERT INTO Supplier1     VALUES (‘13 Watford St’, ‘604-253-6346’),
                                 (‘164 Alma St’, ‘778-354-5743’),
                                 (‘99 Jump St’, ‘604-888-8888’),
                                 (‘1443 Commercial St’, ‘778-231-1111’),
                                 (‘1212 Orlando St’, ‘236-777-7777’);

INSERT INTO Supplier2     VALUES (‘13 Watford St’, ‘Entertainment’),
                                 (‘164 Alma St’, ‘Food’),
                                 (‘99 Jump St’, ‘Sanitation’),
                                 ('1443 Commercial St', 'Grooming'),
                                 ('1212 Orlando St', 'Healthcare');

INSERT INTO Supplier3     VALUES (‘Entertainment’, ‘Pet Toys’),
                                 (‘Food’, ‘Pet Food’),
                                 (‘Sanitation’, ‘Waste Management Product’),
                                 ('Grooming', 'Pet Grooming Products'),
                                 ('Healthcare', 'Pet Vitamins');

INSERT INTO Pet1          VALUES(1, 'Chad'),
                                (2, 'Snoopy'),
                                (3, 'Fido'),
                                (4, 'Stripes'),
                                (5, 'Beerus'),
                                (6, 'Buddy'),
                                (7, 'Max'),
                                (8, 'Charlie'),
                                (9, 'Lucy'),
                                (10, ‘Betty’),
                                (11, 'Tom’');

INSERT INTO Pet2          VALUES(1, 3),
                                (2, 5),
                                (3, 1),
                                (4, 7),
                                (5, 10),
                                (6, 8),
                                (7, 9),
                                (8, 10),
                                (9, 11),
                                (10,7),
                                (11, 6);

INSERT INTO Pet3          VALUES(1, ‘Healthy’),
```

(2, ‘Diabetes’),
(3, ‘Healthy’),
(4, ‘Vision Impairment’),
(5, ‘Arthritis’),
(6, 'Healthy'),
(7, 'Healthy'),
(8, 'Healthy'),
(9, 'Healthy'),
(10, 'Healthy'),
(11, 'Healthy');

INSERT INTO Pet4 VALUES(1, NULL),
(2, '2023-03-28'),
(3, NULL),
(4, '2020-09-14'),
(5, NULL),
(6, NULL),
(7, NULL),
(8, NULL),
(9, NULL),
(10, '2024-10-01'),
(11, NULL);

INSERT INTO Pet5 VALUES(1, '2022-01-15'),
(2, '2020-02-05'),
(3, '2023-11-14'),
(4, '2019-05-18'),
(5, '2024-07-21'),
(6, '2021-09-12'),
(7, '2023-05-20'),
(8, '2022-12-10'),
(9, '2023-01-01'),
(10, '2023-02-15'),
(11, '2023-03-10');

INSERT INTO Pet6 VALUES(1, ‘Turtle’),
(2, ‘Dog’),
(3, ‘Dog’),
(4, ‘Cat’),
(5, ‘Cat’),
(6, 'Dog'),
(7, 'Dog'),
(8, 'Dog'),
(9, 'Cat'),

(10, 'Cat'),
(11, 'Cat');

INSERT INTO Pet7 VALUES(1, ‘Snapping Turtle’),
(2, ‘Bulldog’),
(3, ‘Husky’),
(4, ‘Siamese’),
(5, ‘Sphynx’),
(6, 'Beagle'),
(7, 'Poodle'),
(8, 'Golden Retriever'),
(9, 'Bengal'),
(10, 'Tabby'),
(11, 'Persian');

INSERT INTO Pet8 VALUES(1, NULL),
(2, ‘1234 Main St’),
(3, NULL),
(4, ‘5678 Fraser St’),
(5, NULL),
(6, NULL),
(7, NULL),
(8, NULL),
(9, NULL),
(10, ‘2198 Dunbar St’),
(11, NULL);

INSERT INTO Pet9 VALUES(‘Turtle’, 3, ‘Calcium’),
(‘Dog’, 5, ‘Protein’),
(‘Dog’, 1, ‘Fat’),
(‘Cat’, 7, ‘Taurine’),
(‘Cat’, 10, ‘Taurine’);
(‘Dog’, 6, 'Fat'),
(‘Dog’, 7, 'Protein'),
(‘Dog’, 8, ‘Fiber’),
(‘Cat’, 9, 'Fat'),
(‘Cat’, 10,'Protein'),
(‘Cat’, 11, ‘Calcium’);

INSERT INTO Pet10 VALUES(‘Turtle’, ‘Snapping Turtle’, 40),
(‘Dog’, ‘Bulldog’, 8),
(‘Dog’, ‘Husky’, 12),
(‘Cat’, ‘Siamese’, 15),
(‘Cat’, ‘Sphynx’, 15),

```
                                ('Dog', 'Beagle', 10),
                                ('Dog', 'Poodle', 5),
                                ('Dog', 'Golden Retriever', 12),
                                ('Cat', 'Bengal', 10),
                                ('Cat', 'Tabby', 8),
                                ('Cat', 'Persian', 10);

INSERT INTO Pet11               VALUES(‘Snapping Turtle’, ‘Calcium’, ‘Healthy’, ‘Regular Check-up’),
                                (‘Bulldog’, ‘Protein’, ‘Diabetes’, ‘Daily Walks’),
                                (‘Husky’, ‘Fat’, ‘Healthy’, ‘Exercise Once a Week’),
                                (‘Siamese’, ‘Taurine’, ‘Vision Impairment’, ‘Stay Indoors’),
                                (‘Sphynx’, ‘Taurine’, ‘Arthritis’, ‘Let Roam’),
                                ('Beagle', 'Fat', 'Healthy', 'Daily Walks'),
                                ('Poodle', 'Protein', 'Healthy', 'Regular Check-up'),
                                ('Golden Retriever', ‘Fiber’, 'Healthy', 'Play Daily'),
                                ('Bengal', 'Fat', 'Healthy', 'Groom Regularly'),
                                ('Tabby', 'Protein', 'Healthy', 'Feed Twice a Day'),
                                ('Persian', ‘Calcium’, 'Healthy', 'Brush Regularly');

INSERT INTO Documentation       VALUES(1, '111-111-1323', 345, 'Vaccination Records',
'2004-01-04'),
                                (2, '234-245-2311', 89, 'Allergies', '2001-12-04'),
                                (3, '604-323-1212', 982, 'Adoption History', '2012-02-10'),
                                (4, '999-233-3232', 243, 'Allergies', '2020-03-29'),
                                (5, '456-234-1242', 789, 'Vaccination Records',
                        '2001-01-01');

INSERT INTO AdoptionApplication VALUES('100 Fraser St', '1234 Main St', 801, '2024-03-16',
TRUE),
                                        ('54 Robson St', '5678 Fraser St', 450, '2023-01-28',
                                TRUE),
                                        ('74 Granville St', '2198 Dunbar St', 244,
                                '2012-12-12', FALSE),
                                        ('76 Nanaimo St', '1570 Cambie St', 143,
                                '2022-05-12', TRUE),
                                        ('16 Main St', '3916 Knight St', 93, '2020-07-28',
                                FALSE);

INSERT INTO Dog                 VALUES(2, ‘Medium’, 9),
                                (3, ‘Long’, 10),
                                (6, 'Short-Coated', 5),
                                (7, 'Curly-Coated', 6),
                                (8, 'Fluffy', 8);
```

```
INSERT INTO Cat              VALUES(4, ‘Bristly’, 20),
                                   (5, ‘Hairless’, 18),
                                   (9, 'Short-Haired', 10),
                                   (10, 'Long-Haired', 12),
                                   (11, 'Fluffy', 14);

INSERT INTO PurchasesFrom    VALUES ('13 Watford St', '100 Fraser St'),
                                    ('164 Alma St', '54 Robson St'),
                                    ('99 Jump St', '74 Granville St'),
                                    ('1443 Commercial St', '76 Nanaimo St'),
                                    ('1212 Orlando St', '16 Main St'),
                                    ('1212 Orlando St', '100 Fraser St');

INSERT INTO Trains           VALUES(‘778-111-1111’, 1),
                                   (‘778-111-1112’, 2),
                                   (‘778-111-1113’, 3),
                                   (‘778-111-1114’, 4),
                                   (‘778-111-1115’, 5);

INSERT INTO Examines         VALUES ('111-111-1323', 1),
                                    ('234-245-2311', 1),
                                    ('234-245-2311', 2),
                                    ('604-323-1212', 2),
                                    ('999-233-3232', 3),
                                    ('456-234-1242', 4),
                                    ('111-111-1323', 5);

INSERT INTO VetWorksWithShel VALUES(‘111-111-1323’, ‘100 Fraser St’),
                                   (‘234-245-2311’, ‘54 Robson St’),
                                   (‘604-323-1212’, ‘74 Granville St’),
                                   (‘999-233-3232’, ‘76 Nanaimo St’),
                                   (‘456-234-1242’, ‘16 Main St’);


INSERT INTO TrainWorksWithShel       VALUES(‘778-111-1111’, ‘100 Fraser St’),
                                           (‘778-111-1112’, ‘54 Robson St’),
                                           (‘778-111-1113’, ‘74 Granville St’),
                                           (‘778-111-1114’, ‘76 Nanaimo St’),
                                           (‘778-111-1115’, ‘16 Main St’);

INSERT INTO Hosts            VALUES (1, '100 Fraser St'),
                                    (2, '54 Robson St'),
                                    (3, '74 Granville St'),
                                    (4, '76 Nanaimo St'),
```

```
(5, '16 Main St'),
(5, '100 Fraser St');
```