# ANIMAL SHELTER DATABASE

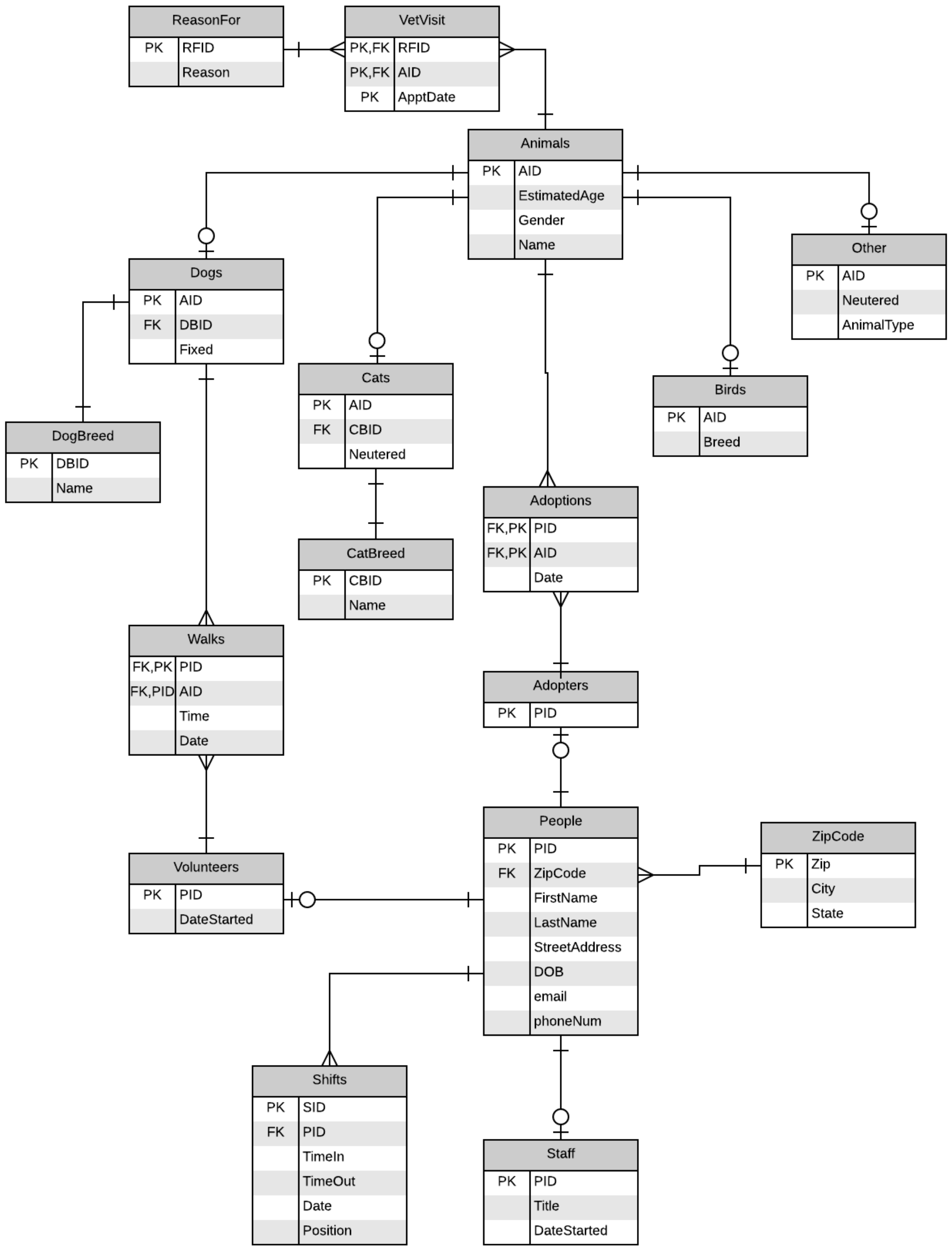
Mackenzie O’Brien

# Table of Contents

# EXECUTIVE SUMMARY

- The Animal Shelter provides vaccinations for animals as well as a temporary home for them until they are adopted. They require a database to keep track of the animals and the adoptions, as well as the volunteers and shelter staff that have shifts at the shelter. This database must be accurate to ensure the safety of the animals and that the animals are not returned to the shelter, but instead end up in safe and happy homes.
- This is intended to be used by the Animal Shelter to keep track of animals, people, and adoption records. It will allow the agency to ensure all animals are cared for and keep track of adoption histories, as well as things done on a daily basis.
- Events that are kept track of are shifts of employees/volunteers, vaccinations, adoptions, and walks.

# E-R Diagram



ReasonFor

| | |
|---|---|
| PK | RFID |
| | Reason |

VetVisit

| | |
|---|---|
| PK,FK | RFID |
| PK,FK | AID |
| PK | ApptDate |

Animals

| | |
|---|---|
| PK | AID |
| | EstimatedAge |
| | Gender |
| | Name |

Other

| | |
|---|---|
| PK | AID |
| | Neutered |
| | AnimalType |

Dogs

| | |
|---|---|
| PK | AID |
| FK | DBID |
| | Fixed |

DogBreed

| | |
|---|---|
| PK | DBID |
| | Name |

Cats

| | |
|---|---|
| PK | AID |
| FK | CBID |
| | Neutered |

CatBreed

| | |
|---|---|
| PK | CBID |
| | Name |

Birds

| | |
|---|---|
| PK | AID |
| | Breed |

Adoptions

| | |
|---|---|
| FK,PK | PID |
| FK,PK | AID |
| | Date |

Adopters

| | |
|---|---|
| PK | PID |

Walks

| | |
|---|---|
| FK,PK | PID |
| FK,PID | AID |
| | Time |
| | Date |

Volunteers

| | |
|---|---|
| PK | PID |
| | DateStarted |

People

| | |
|---|---|
| PK | PID |
| FK | ZipCode |
| | FirstName |
| | LastName |
| | StreetAddress |
| | DOB |
| | email |
| | phoneNum |

ZipCode

| | |
|---|---|
| PK | Zip |
| | City |
| | State |

Shifts

| | |
|---|---|
| PK | SID |
| FK | PID |
| | TimeIn |
| | TimeOut |
| | Date |
| | Position |

Staff

| | |
|---|---|
| PK | PID |
| | Title |
| | DateStarted |

# TYPES

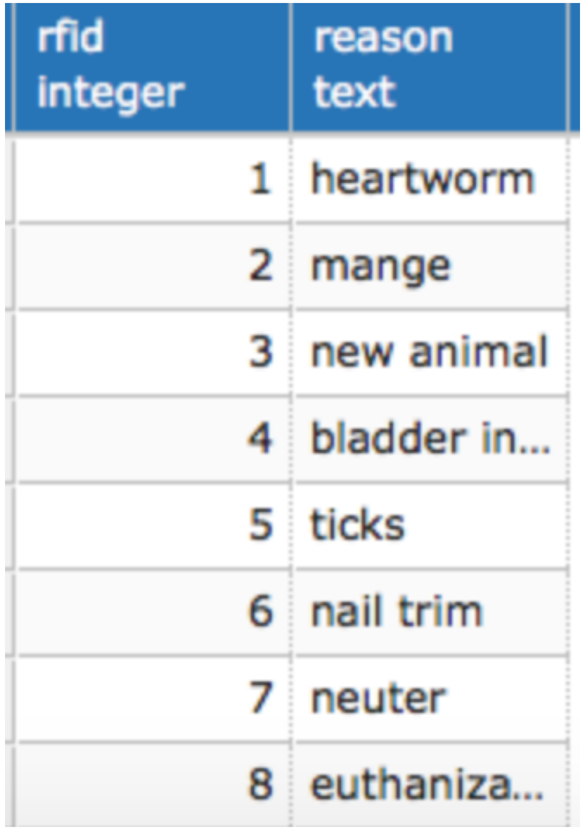
- gen type- refers to gender type male or female
  - CREATE TYPE gen as ENUM(’male’,’female’);
- fixed type-refers to whether the animal was neutured or spayed
  - CREATE TYPE fixed as ENUM(‘yes’,’no’);

ReasonFor

```
CREATE TABLE ReasonFor(
        RFID int not null,
        Reason text not null,
        primary key(RFID)
);
```

Functional Dependencies:

RFID→Reason



| rfid | reason |
|---|---|
| integer | text |
| 1 | heartworm |
| 2 | mange |
| 3 | new animal |
| 4 | bladder in... |
| 5 | ticks |
| 6 | nail trim |
| 7 | neuter |
| 8 | euthaniza... |

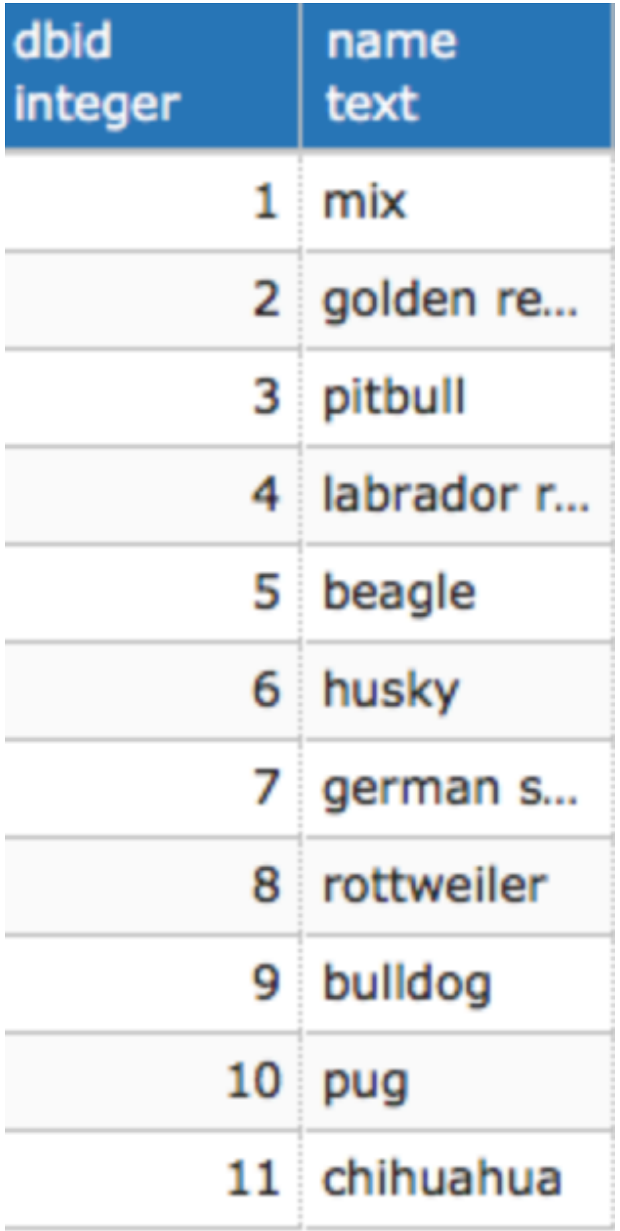
# DogBreed

Lists all dog breeds that the shelter has encountered

```
CREATE TABLE DogBreed (
        DBID int not null,
        Name text not null,
        primary key (DBID)
);
```

Functional Dependencies:

DBID →Name



| dbid<br>integer | name<br>text |
|---|---|
| 1 | mix |
| 2 | golden re... |
| 3 | pitbull |
| 4 | labrador r... |
| 5 | beagle |
| 6 | husky |
| 7 | german s... |
| 8 | rottweiler |
| 9 | bulldog |
| 10 | pug |
| 11 | chihuahua |

# TABLES

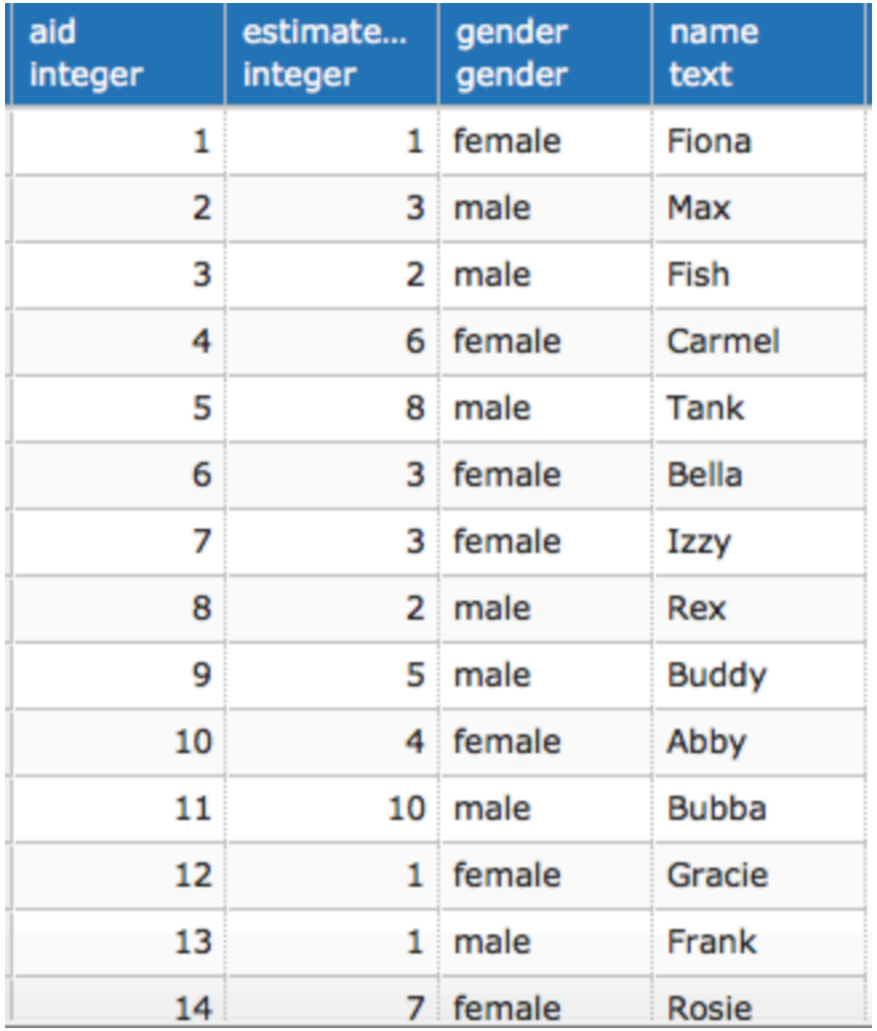
Animals

Lists all animals, both current and adopted

```
CREATE TABLE IF NOT EXISTS Animals(
AID int not null,
EstimatedAge int,
Gender gender not null,
Name text not null,
Primary key (AID)
);
```

Functional Dependencies:

AID→ Name, Gender, EstimatedAge



| aid<br>integer | estimate...<br>integer | gender<br>gender | name<br>text |
|---|---|---|---|
| 1 | 1 | female | Fiona |
| 2 | 3 | male | Max |
| 3 | 2 | male | Fish |
| 4 | 6 | female | Carmel |
| 5 | 8 | male | Tank |
| 6 | 3 | female | Bella |
| 7 | 3 | female | Izzy |
| 8 | 2 | male | Rex |
| 9 | 5 | male | Buddy |
| 10 | 4 | female | Abby |
| 11 | 10 | male | Bubba |
| 12 | 1 | female | Gracie |
| 13 | 1 | male | Frank |
| 14 | 7 | female | Rosie |

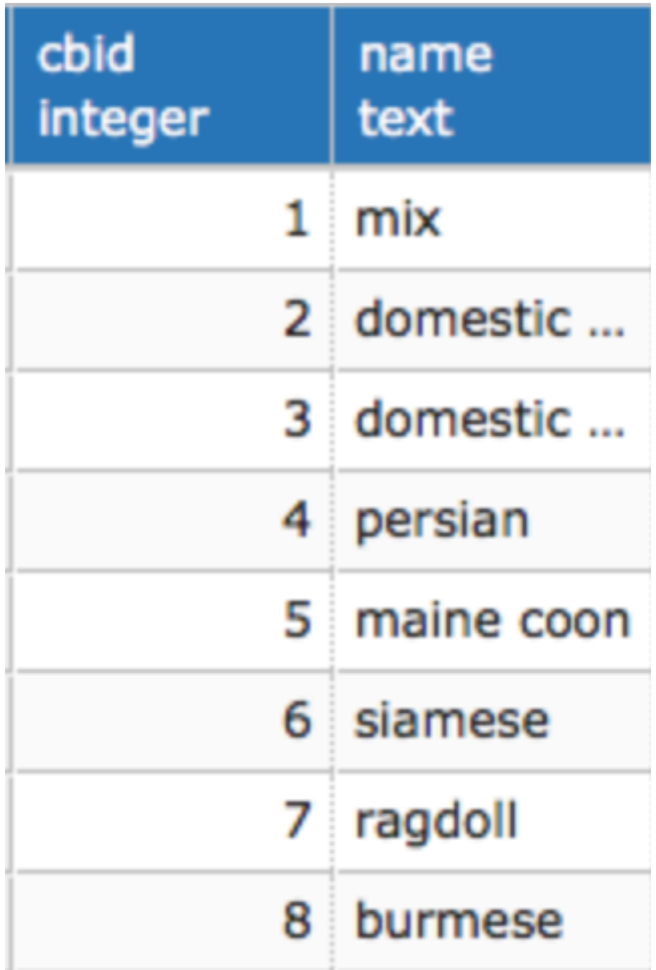
CatBreed

Lists all catBreeds that the shelter has encountered

```
CREATE TABLE CatBreed (
        CBID int not null,
        Name text not null,
        primary key (CBID)
);
```

Functional Dependencies:

CBID →Name



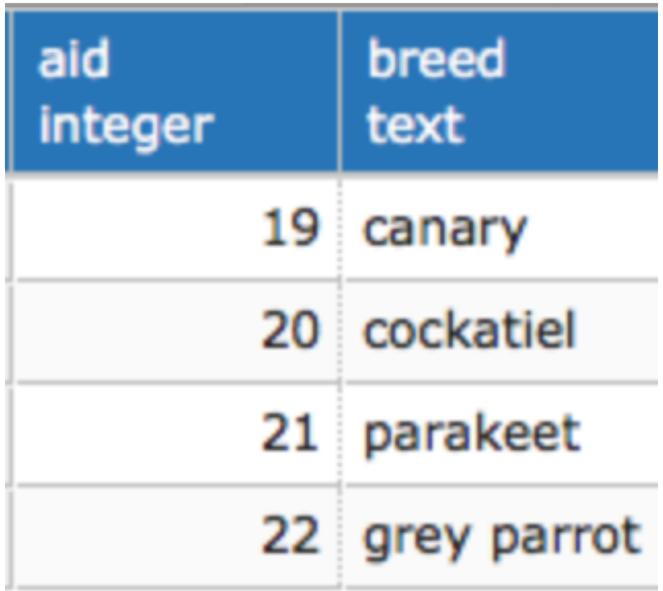
| cbid<br>integer | name<br>text |
|---|---|
| 1 | mix |
| 2 | domestic ... |
| 3 | domestic ... |
| 4 | persian |
| 5 | maine coon |
| 6 | siamese |
| 7 | ragdoll |
| 8 | burmese |

## Birds

Is a subtype of Animals, lists all birds in the shelter or that have been in the shelter. Also lists the breed of the bird.

```
CREATE TABLE Birds(
AID int not null references Animals(AID),
Breed text not null,
Primary key (AID)
);
```

Functional Dependencies: AID→Breed



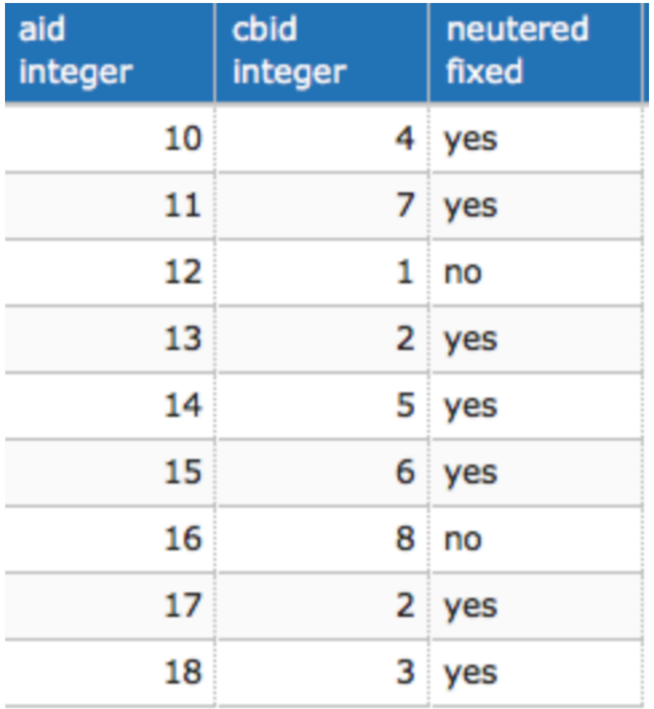
| aid<br>integer | breed<br>text |
|---|---|
| 19 | canary |
| 20 | cockatiel |
| 21 | parakeet |
| 22 | grey parrot |

Cats

Is a subtype of Animals, lists out all cats that have been through the shelter or are currently there.

```
CREATE TABLE Cats(
AID int not null references Animals(AID),
CBID int not null references CatBreed(CBID),
Neutered fixed not null,
Primary key(AID)
);
```

Functional Dependencies: AID→CBID, Neutered



| aid<br>integer | cbid<br>integer | neutered<br>fixed |
|---|---|---|
| 10 | 4 | yes |
| 11 | 7 | yes |
| 12 | 1 | no |
| 13 | 2 | yes |
| 14 | 5 | yes |
| 15 | 6 | yes |
| 16 | 8 | no |
| 17 | 2 | yes |
| 18 | 3 | yes |

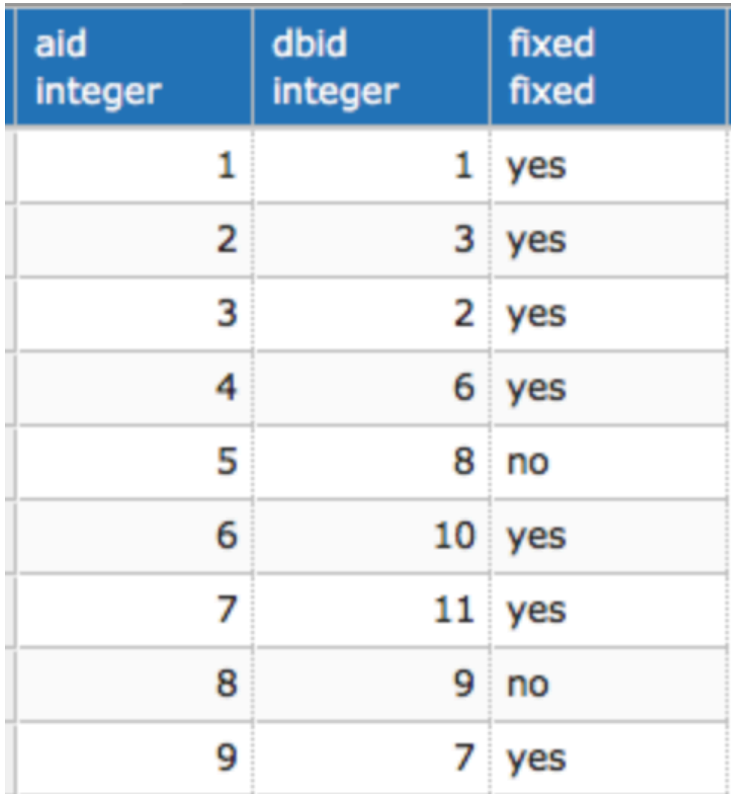
## Dogs

This table is a subtype of Animals and lists out all dogs that have been in the shelter or currently are there.

```
CREATE TABLE Dogs(
        AID  int not null references Animals(AID),
        DBID int not null references DogBreed(DBID),
        Fixed fixed not null,
        primary key (AID)
);
```

Functional Dependencies:

AID→DBID, Fixed

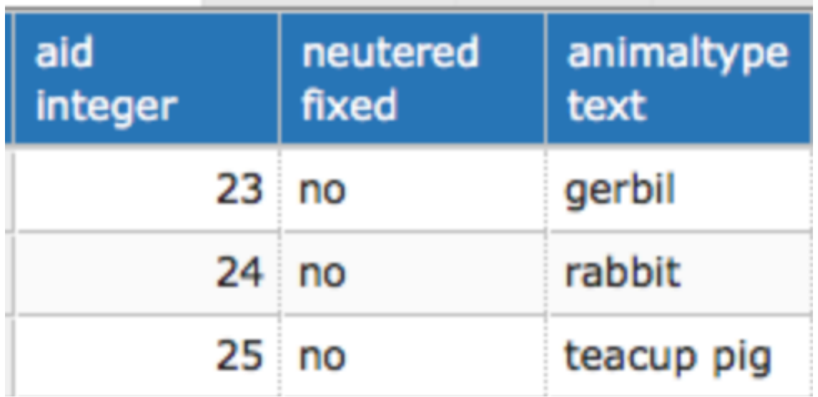


| aid | dbid | fixed |
|---|---|---|
| integer | integer | fixed |
| 1 | 1 | yes |
| 2 | 3 | yes |
| 3 | 2 | yes |
| 4 | 6 | yes |
| 5 | 8 | no |
| 6 | 10 | yes |
| 7 | 11 | yes |
| 8 | 9 | no |
| 9 | 7 | yes |

Other

List of Other Animals, specifies what kind of animal they are

```
CREATE TABLE Other(
          AID int not null references Animals (AID),
          Neutered fixed not null,
          AnimalType text not null,
         primary key (AID)
);
```



| aid<br>integer | neutered<br>fixed | animaltype<br>text |
|---|---|---|
| 23 | no | gerbil |
| 24 | no | rabbit |
| 25 | no | teacup pig |

Functional Dependencies:

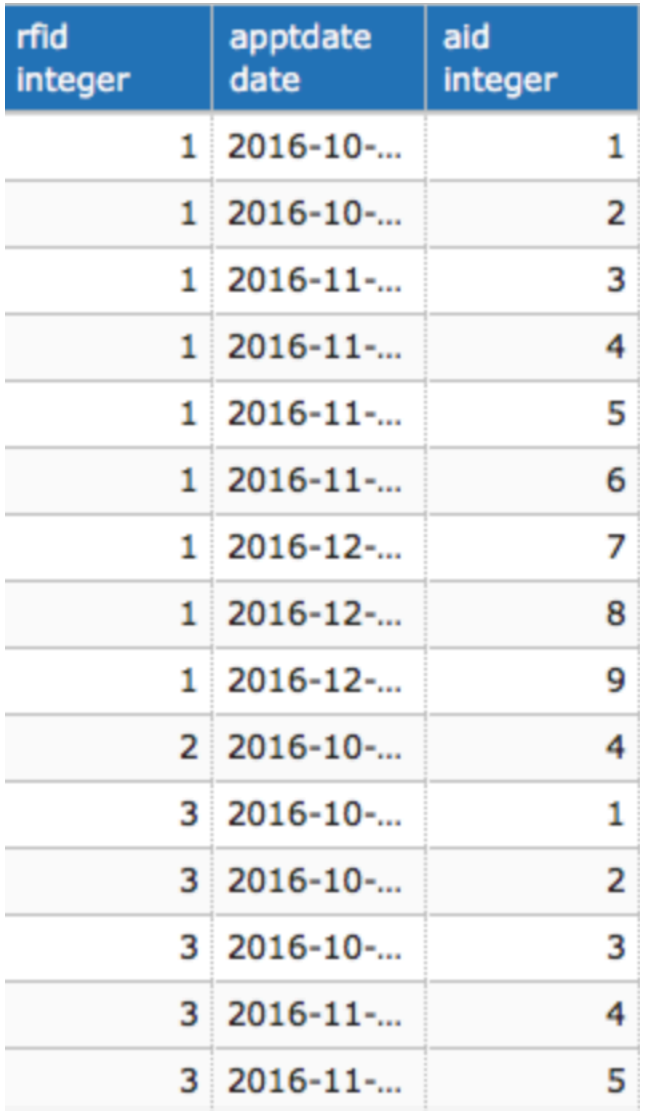
AID→Neutered, AnimalType,

VetVisit

Lists all visits to the vet for all animals and includes if they got vaccinations

```
CREATE TABLE VetVisit(
        RFID int ReasonFor(RFID),
        ApptDate date not null,
        AID int not null references Animals(AID),
        primary key (AID,ApptDate, RFID)
);
```

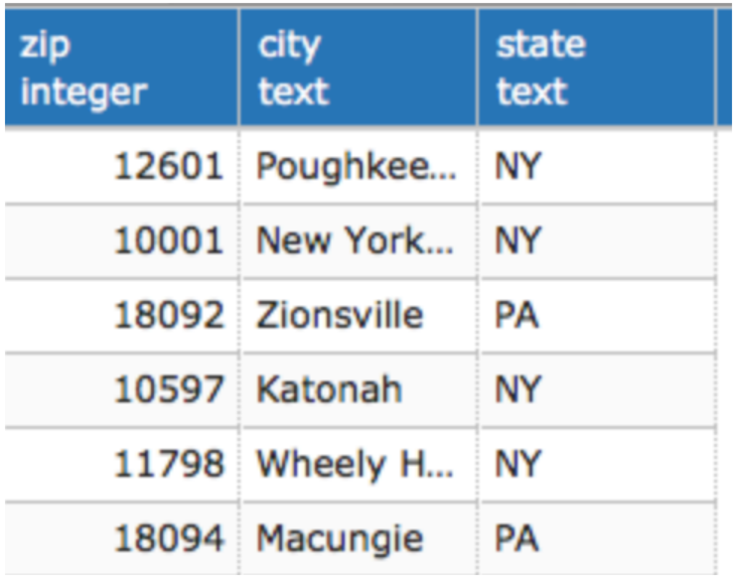
No functional dependencies



| rfid<br>integer | apptdate<br>date | aid<br>integer |
|---|---|---|
| 1 | 2016-10-... | 1 |
| 1 | 2016-10-... | 2 |
| 1 | 2016-11-... | 3 |
| 1 | 2016-11-... | 4 |
| 1 | 2016-11-... | 5 |
| 1 | 2016-11-... | 6 |
| 1 | 2016-12-... | 7 |
| 1 | 2016-12-... | 8 |
| 1 | 2016-12-... | 9 |
| 2 | 2016-10-... | 4 |
| 3 | 2016-10-... | 1 |
| 3 | 2016-10-... | 2 |
| 3 | 2016-10-... | 3 |
| 3 | 2016-11-... | 4 |
| 3 | 2016-11-... | 5 |

ZipCode- List of all zipcodes that have been used in the system

```
CREATE TABLE Zipcode(
             Zip int not null,
            City text not null,
            State text not null,
            primary key(Zip)
);
```



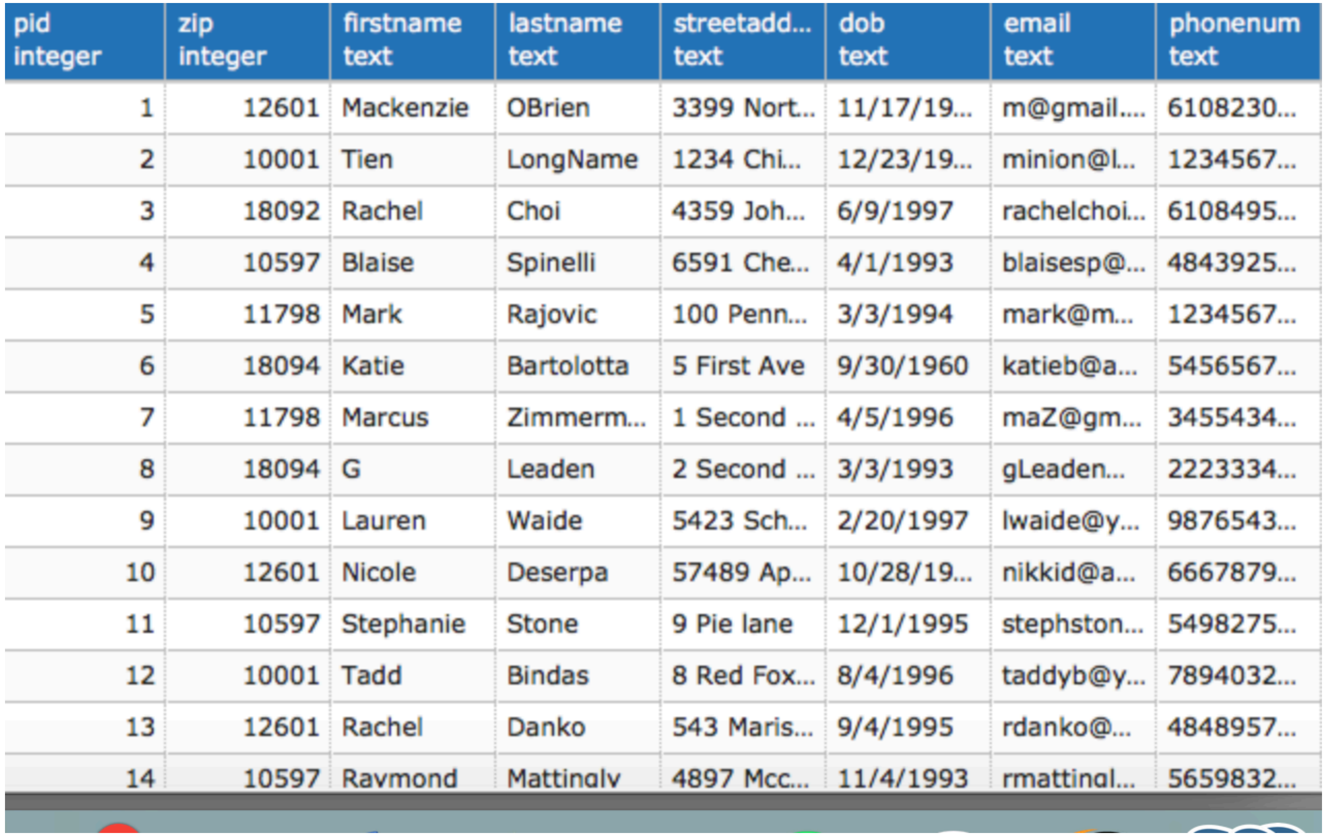
| zip<br>integer | city<br>text | state<br>text |
|---|---|---|
| 12601 | Poughkee... | NY |
| 10001 | New York... | NY |
| 18092 | Zionsville | PA |
| 10597 | Katonah | NY |
| 11798 | Wheely H... | NY |
| 18094 | Macungie | PA |

Functional Dependencies:

Zip → City, State

## People- List of all people in the system

```
CREATE TABLE People(
        PID int not null,
        Zip int not null references ZipCode(Zip),
        FirstName text not null,
        LastName text not null,
        StreetAddress text not null,
        DOB date not null,
        email text not null,
        phoneNum text not null,
        primary key (PID)
);
```



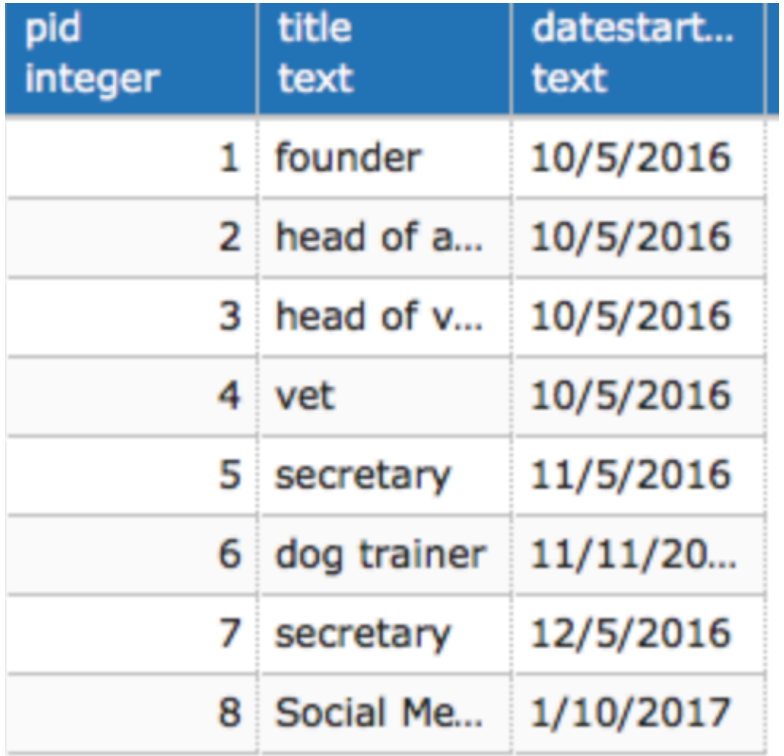
| pid<br>integer | zip<br>integer | firstname<br>text | lastname<br>text | streetadd...<br>text | dob<br>text | email<br>text | phonenum<br>text |
|---|---|---|---|---|---|---|---|
| 1 | 12601 | Mackenzie | OBrien | 3399 Nort... | 11/17/19... | m@gmail.... | 6108230... |
| 2 | 10001 | Tien | LongName | 1234 Chi... | 12/23/19... | minion@l... | 1234567... |
| 3 | 18092 | Rachel | Choi | 4359 Joh... | 6/9/1997 | rachelchoi... | 6108495... |
| 4 | 10597 | Blaise | Spinelli | 6591 Che... | 4/1/1993 | blaisesp@... | 4843925... |
| 5 | 11798 | Mark | Rajovic | 100 Penn... | 3/3/1994 | mark@m... | 1234567... |
| 6 | 18094 | Katie | Bartolotta | 5 First Ave | 9/30/1960 | katieb@a... | 5456567... |
| 7 | 11798 | Marcus | Zimmerm... | 1 Second ... | 4/5/1996 | maZ@gm... | 3455434... |
| 8 | 18094 | G | Leaden | 2 Second ... | 3/3/1993 | gLeaden... | 2223334... |
| 9 | 10001 | Lauren | Waide | 5423 Sch... | 2/20/1997 | lwaide@y... | 9876543... |
| 10 | 12601 | Nicole | Deserpa | 57489 Ap... | 10/28/19... | nikkid@a... | 6667879... |
| 11 | 10597 | Stephanie | Stone | 9 Pie lane | 12/1/1995 | stephston... | 5498275... |
| 12 | 10001 | Tadd | Bindas | 8 Red Fox... | 8/4/1996 | taddyb@y... | 7894032... |
| 13 | 12601 | Rachel | Danko | 543 Maris... | 9/4/1995 | rdanko@... | 4848957... |
| 14 | 10597 | Raymond | Mattingly | 4897 Mcc... | 11/4/1993 | rmattingl... | 5659832... |

Functional Dependencies:

PID→ Zip, FirstName, LastName, StreetAddress, DOB, email, phoneNum

Staff

```
CREATE TABLE Staff(
        PID int not null references People(PID),
        Title text not null,
        DateStarted date not null,
        primary key(PID)
);
```



| pid<br>integer | title<br>text | datestart...<br>text |
|---|---|---|
| 1 | founder | 10/5/2016 |
| 2 | head of a... | 10/5/2016 |
| 3 | head of v... | 10/5/2016 |
| 4 | vet | 10/5/2016 |
| 5 | secretary | 11/5/2016 |
| 6 | dog trainer | 11/11/20... |
| 7 | secretary | 12/5/2016 |
| 8 | Social Me... | 1/10/2017 |

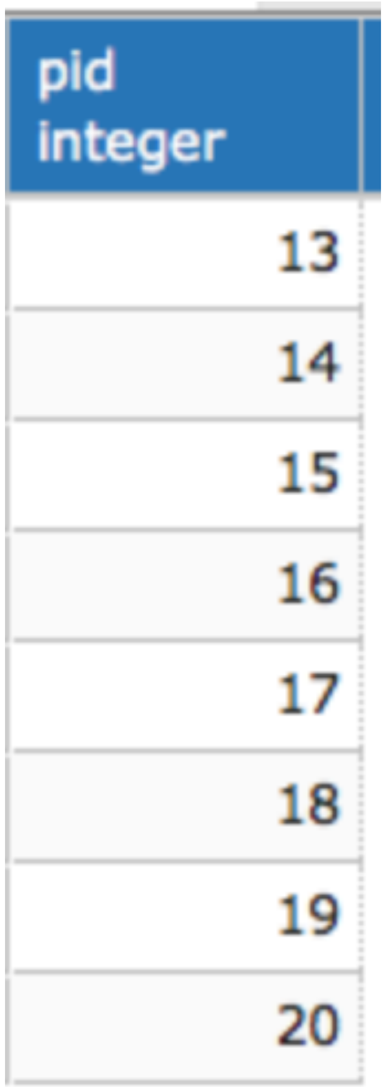
Functional Dependencies:

PID→ Title, DateStarted

## Adopters

```
CREATE TABLE Adopters(
    PID int not null references People(PID),
   primary key(PID)
);
```

Functional Dependencies: None

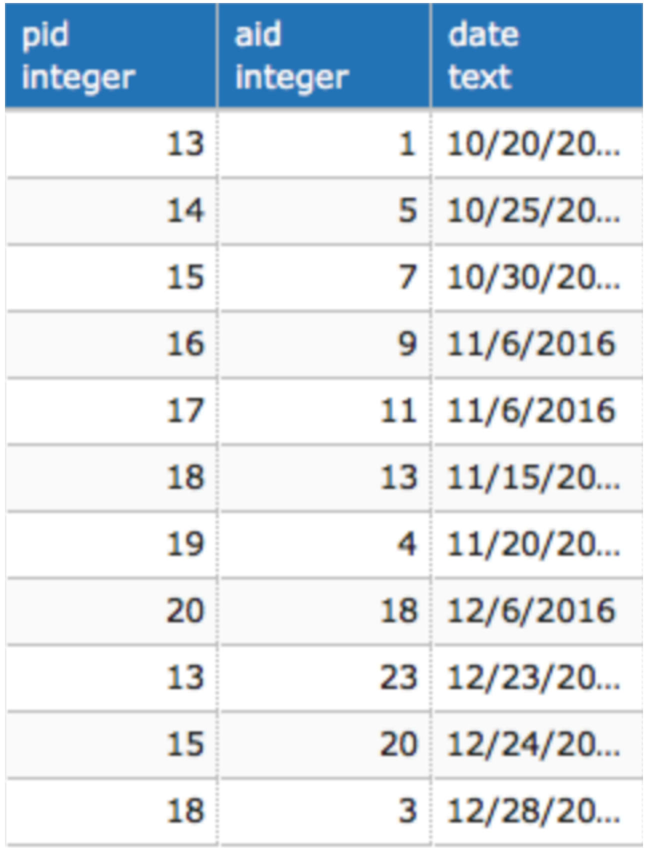


| pid<br>integer |
|---|
| 13 |
| 14 |
| 15 |
| 16 |
| 17 |
| 18 |
| 19 |
| 20 |

Adoptions

```
 CREATE TABLE Adoptions(
        PID int not null references Adopters(PID),
        AID int not null references Animals(AID),
        Date date not null,
        primary key(PID, AID)
);
```

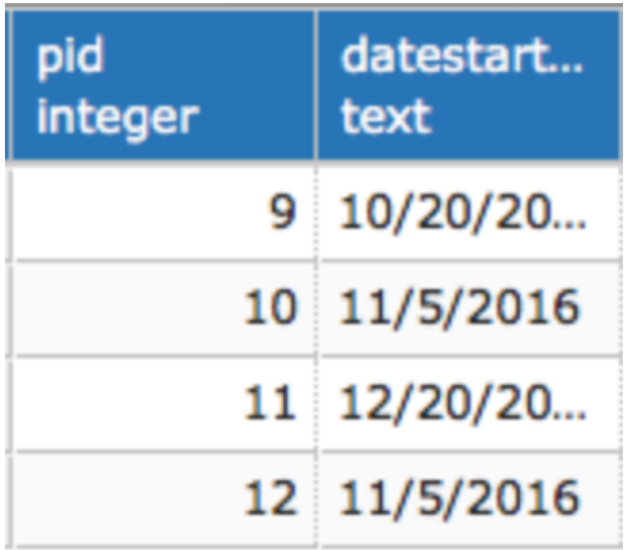
Functional Dependencies: PID, AID→ Date



| pid<br>integer | aid<br>integer | date<br>text |
|---|---|---|
| 13 | 1 | 10/20/20... |
| 14 | 5 | 10/25/20... |
| 15 | 7 | 10/30/20... |
| 16 | 9 | 11/6/2016 |
| 17 | 11 | 11/6/2016 |
| 18 | 13 | 11/15/20... |
| 19 | 4 | 11/20/20... |
| 20 | 18 | 12/6/2016 |
| 13 | 23 | 12/23/20... |
| 15 | 20 | 12/24/20... |
| 18 | 3 | 12/28/20... |

Volunteers

```
CREATE TABLE Volunteers(
        PID int not null references People(PID),
        DateStarted date not null,
        primary key (PID)
);
```



| pid<br>integer | datestart...<br>text |
|---|---|
| 9 | 10/20/20... |
| 10 | 11/5/2016 |
| 11 | 12/20/20... |
| 12 | 11/5/2016 |

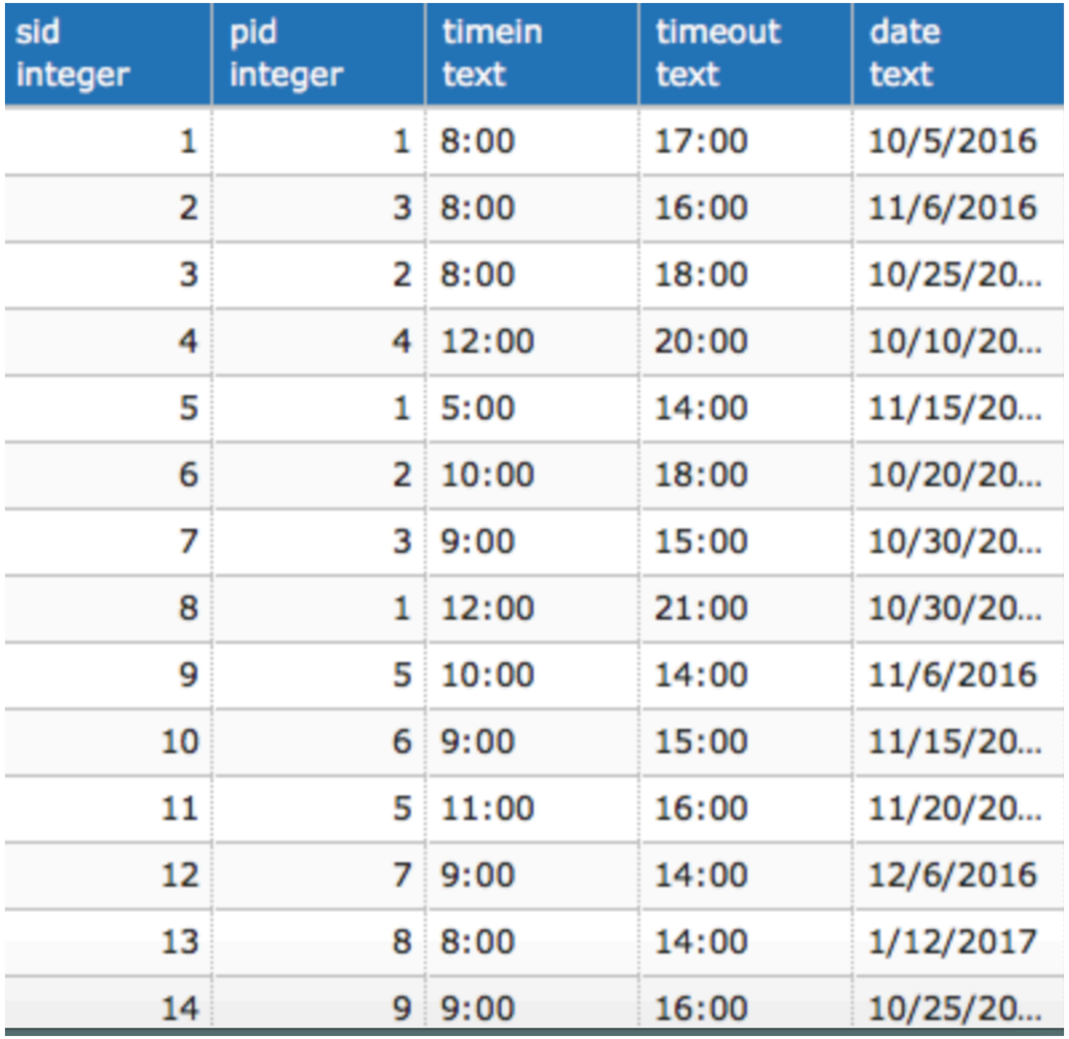
Functional Dependencies: PID→DateStarted

Shifts

```
CREATE TABLE Shifts(
        SID int not null,
        PID int not null references People(PID),
        TimeIn time not null,
        TimeOut time not null,
        Date date not null,
        primary key(SID)
);
```

Functional Dependencies:

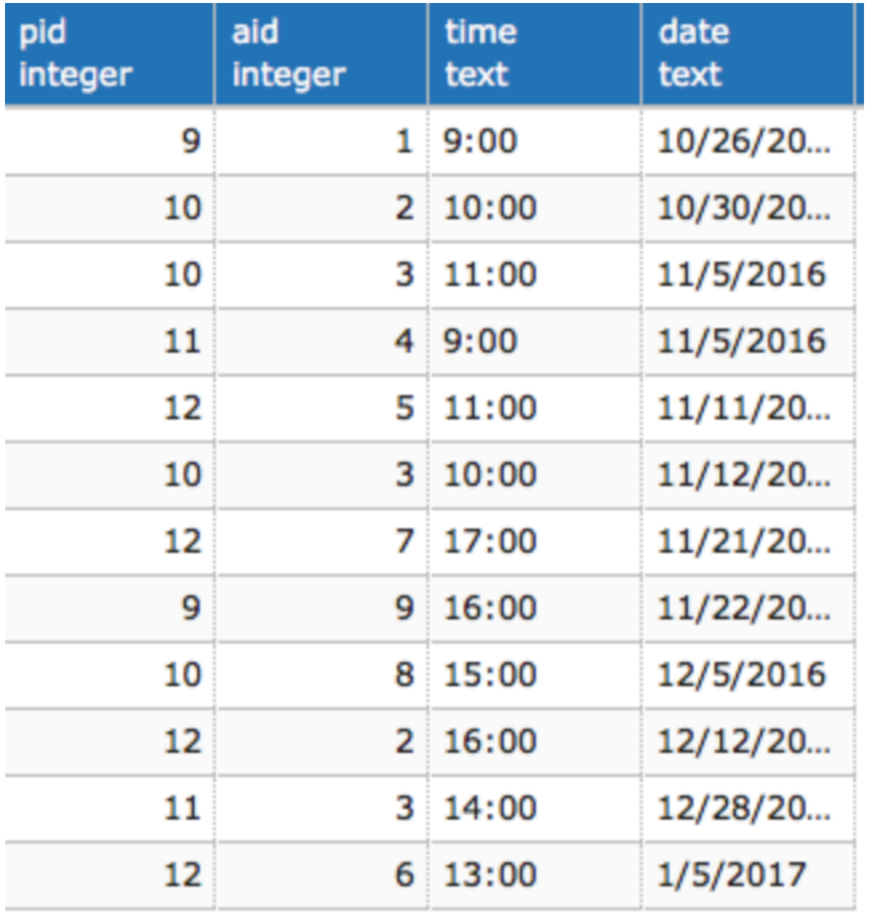
SID, PID→ TimeIn, TimeOut, Date



| sid<br>integer | pid<br>integer | timein<br>text | timeout<br>text | date<br>text |
|---|---|---|---|---|
| 1 | 1 | 8:00 | 17:00 | 10/5/2016 |
| 2 | 3 | 8:00 | 16:00 | 11/6/2016 |
| 3 | 2 | 8:00 | 18:00 | 10/25/20... |
| 4 | 4 | 12:00 | 20:00 | 10/10/20... |
| 5 | 1 | 5:00 | 14:00 | 11/15/20... |
| 6 | 2 | 10:00 | 18:00 | 10/20/20... |
| 7 | 3 | 9:00 | 15:00 | 10/30/20... |
| 8 | 1 | 12:00 | 21:00 | 10/30/20... |
| 9 | 5 | 10:00 | 14:00 | 11/6/2016 |
| 10 | 6 | 9:00 | 15:00 | 11/15/20... |
| 11 | 5 | 11:00 | 16:00 | 11/20/20... |
| 12 | 7 | 9:00 | 14:00 | 12/6/2016 |
| 13 | 8 | 8:00 | 14:00 | 1/12/2017 |
| 14 | 9 | 9:00 | 16:00 | 10/25/20... |

## Walks

```
CREATE TABLE Walks(
        PID int not null references Volunteers(PID),
        AID int not null references Dogs(AID),
        Time time not null,
        Date date not null,
        primary key(PID,AID)
);
```



| pid<br>integer | aid<br>integer | time<br>text | date<br>text |
|---|---|---|---|
| 9 | 1 | 9:00 | 10/26/20... |
| 10 | 2 | 10:00 | 10/30/20... |
| 10 | 3 | 11:00 | 11/5/2016 |
| 11 | 4 | 9:00 | 11/5/2016 |
| 12 | 5 | 11:00 | 11/11/20... |
| 10 | 3 | 10:00 | 11/12/20... |
| 12 | 7 | 17:00 | 11/21/20... |
| 9 | 9 | 16:00 | 11/22/20... |
| 10 | 8 | 15:00 | 12/5/2016 |
| 12 | 2 | 16:00 | 12/12/20... |
| 11 | 3 | 14:00 | 12/28/20... |
| 12 | 6 | 13:00 | 1/5/2017 |

Functional Dependencies: PID,AID→ Time, Date

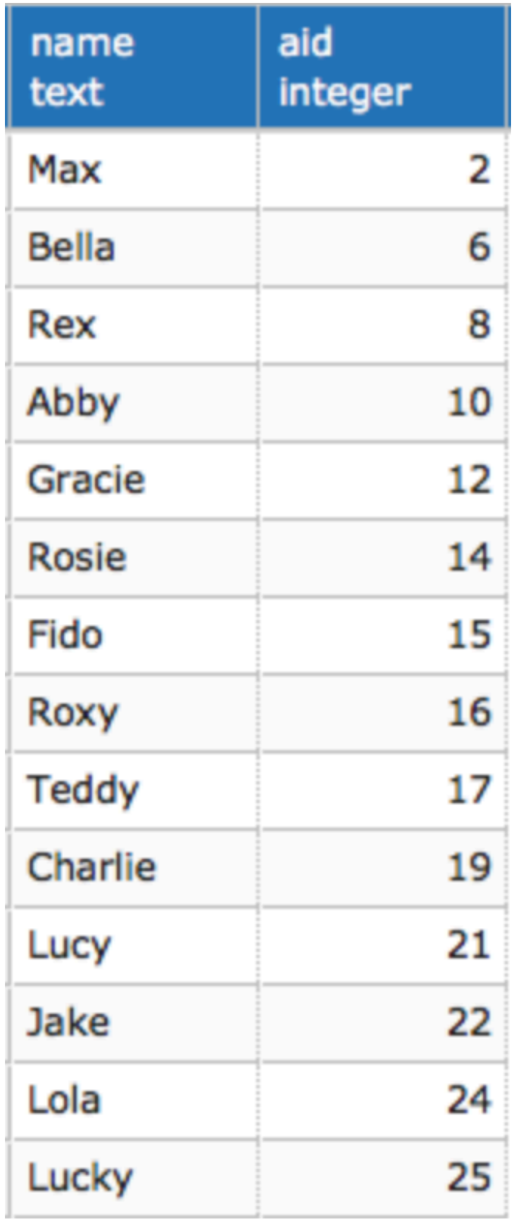
# VIEWS

currentAvailable- Selects the animals that are currently available for adoption at the shelter

```
DROP VIEW IF EXISTS currentAvailable;


CREATE VIEW currentAvailable as
        select name, animals.aid
        from animals
        where animals.aid not in(select animals.AID
                                 from Animals
                                 inner join Adoptions on Animals.AID=Adoptions.AID
                                 )    and EstimatedAge is not null
 ORDER BY Animals.AID ASC;
```

Screen shot of CurrentAvailable results



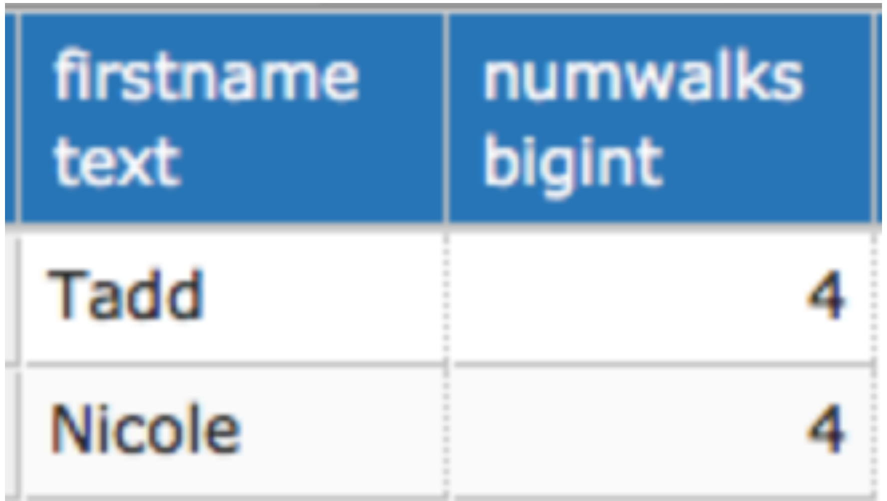
| name<br>text | aid<br>integer |
|---|---|
| Max | 2 |
| Bella | 6 |
| Rex | 8 |
| Abby | 10 |
| Gracie | 12 |
| Rosie | 14 |
| Fido | 15 |
| Roxy | 16 |
| Teddy | 17 |
| Charlie | 19 |
| Lucy | 21 |
| Jake | 22 |
| Lola | 24 |
| Lucky | 25 |

# VIEWS

MostWalks- Selects the names of the volunteers with the most number of walks- used tor a rewards program

```
DROP VIEW IF EXISTS mostWalks;
CREATE VIEW mostWalks as
                    select FirstName , count(FirstName) as numWalks
                     from People
                              inner join Volunteers on People.PID=Volunteers.PID
                              inner join Walks on People.PID=Walks.PID
                    group by FirstName
                    ORDER BY count(FirstName) DESC
                    LIMIT 2;
```

Screenshot from MostWalks query

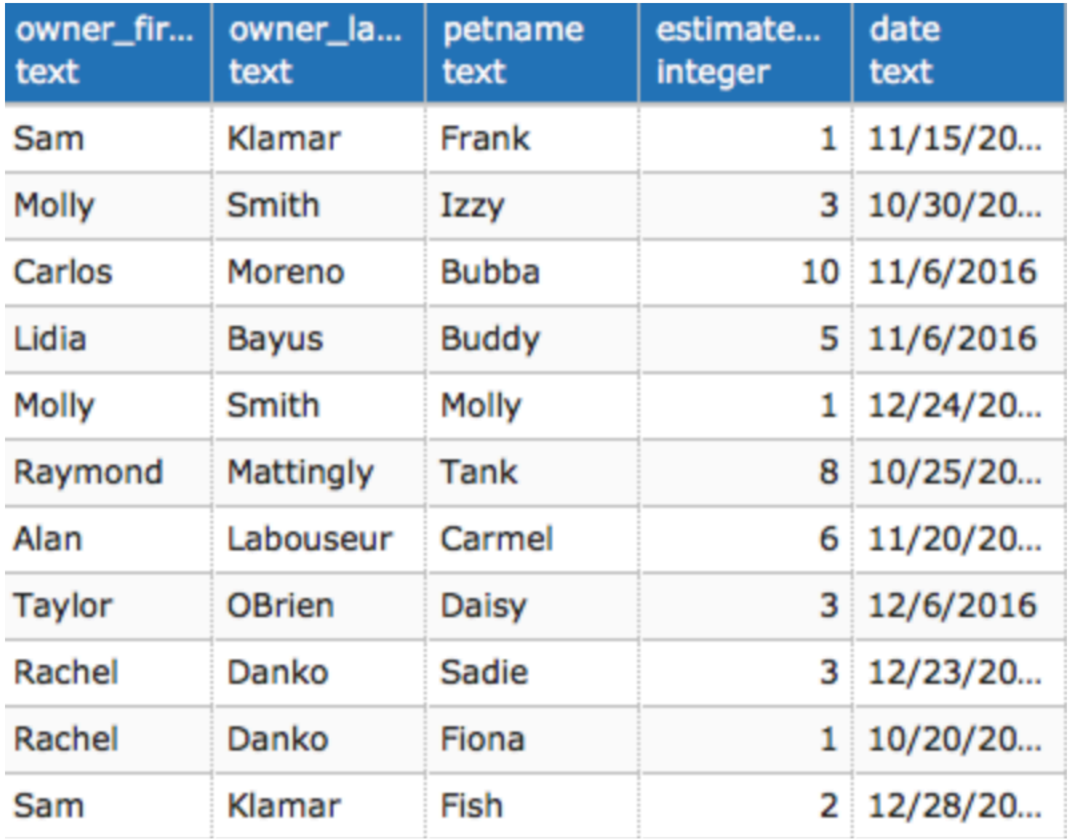


| firstname<br>text | numwalks<br>bigint |
|---|---|
| Tadd | 4 |
| Nicole | 4 |

# REPORTS

Select all adoption records and pull up the names of all parties involved

```
select distinct people.firstName as Owner_FirstName,
people.LastName as Owner_LastName,
animals.Name as PetName, animals.EstimatedAge, adoptions.date
          from adoptions
                    inner join people on people.pid=adoptions.pid
                      inner join animals on adoptions.aid=animals.aid
```



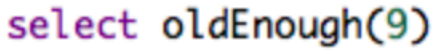
| owner_fir...<br>text | owner_la...<br>text | petname<br>text | estimate...<br>integer | date<br>text |
|---|---|---|---|---|
| Sam | Klamar | Frank | 1 | 11/15/20... |
| Molly | Smith | Izzy | 3 | 10/30/20... |
| Carlos | Moreno | Bubba | 10 | 11/6/2016 |
| Lidia | Bayus | Buddy | 5 | 11/6/2016 |
| Molly | Smith | Molly | 1 | 12/24/20... |
| Raymond | Mattingly | Tank | 8 | 10/25/20... |
| Alan | Labouseur | Carmel | 6 | 11/20/20... |
| Taylor | OBrien | Daisy | 3 | 12/6/2016 |
| Rachel | Danko | Sadie | 3 | 12/23/20... |
| Rachel | Danko | Fiona | 1 | 10/20/20... |
| Sam | Klamar | Fish | 2 | 12/28/20... |

# STORED PROCEDURES

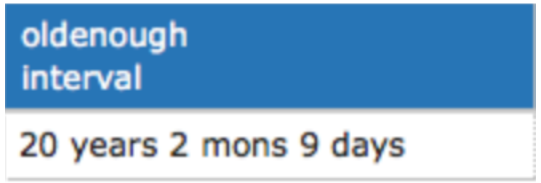
Sets the animals age to null if the Animal is put down by the vet (will not appear in available to adopt query anymore) See trigger for results

```
CREATE OR REPLACE FUNCTION euthanized()
RETURNS TRIGGER AS
$$
BEGIN
        UPDATE Animals
        SET estimatedAge=null
        where NEW.RFID=8 and NEW.aid=Animals.aid;
        RETURN NEW;
 END;
$$
language plpgsql;
```

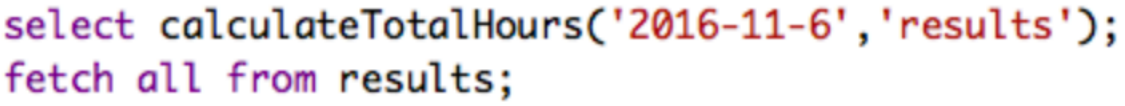
```
CREATE OR REPLACE FUNCTION oldEnough(VolunteerID int)
RETURNS INTERVAL AS
$$
DECLARE
birthday date := (SELECT people.dob
                  from people
                  inner join volunteers on people.pid=volunteers.pid
           where volunteers.pid=VolunteerID
                );
BEGIN
      RETURN age(birthday);
 END;
$$
language plpgsql
```



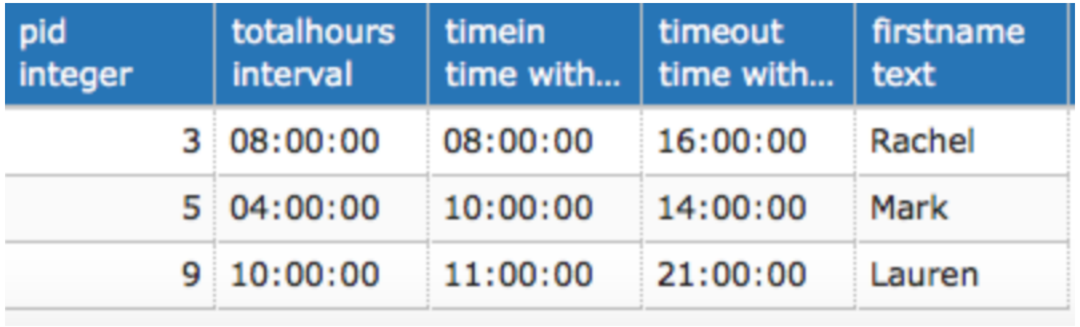
```
select oldEnough(9)
```



| oldenough<br>interval |
|---|
| 20 years 2 mons 9 days |



```
select calculateTotalHours('2016-11-6','results');
fetch all from results;
```



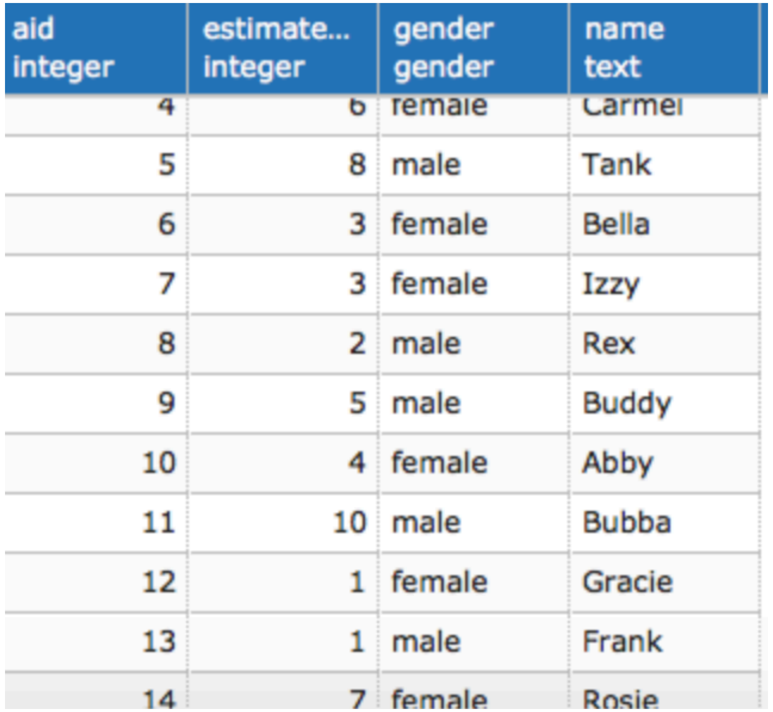
| pid<br>integer | totalhours<br>interval | timein<br>time with... | timeout<br>time with... | firstname<br>text |
|---|---|---|---|---|
| 3 | 08:00:00 | 08:00:00 | 16:00:00 | Rachel |
| 5 | 04:00:00 | 10:00:00 | 14:00:00 | Mark |
| 9 | 10:00:00 | 11:00:00 | 21:00:00 | Lauren |

```
CREATE OR REPLACE FUNCTION
 calculateTotalHours(date, refcursor)
RETURNS refcursor as
$$
DECLARE
            dayOf date :=$1;
            resultset refcursor:=$2;
BEGIN
open resultset for
            select shifts.pid, sum(timeout-timein) as TotalHours,timein, timeout , people.firstName
  from shifts
inner join people on shifts.pid=people.pid
 where shifts.date=dayOf
group by shifts.pid,people.firstName,timein,timeout
    order by pid ASC;
return resultset;
 END;$$language plpgsql;
```

# TRIGGERS

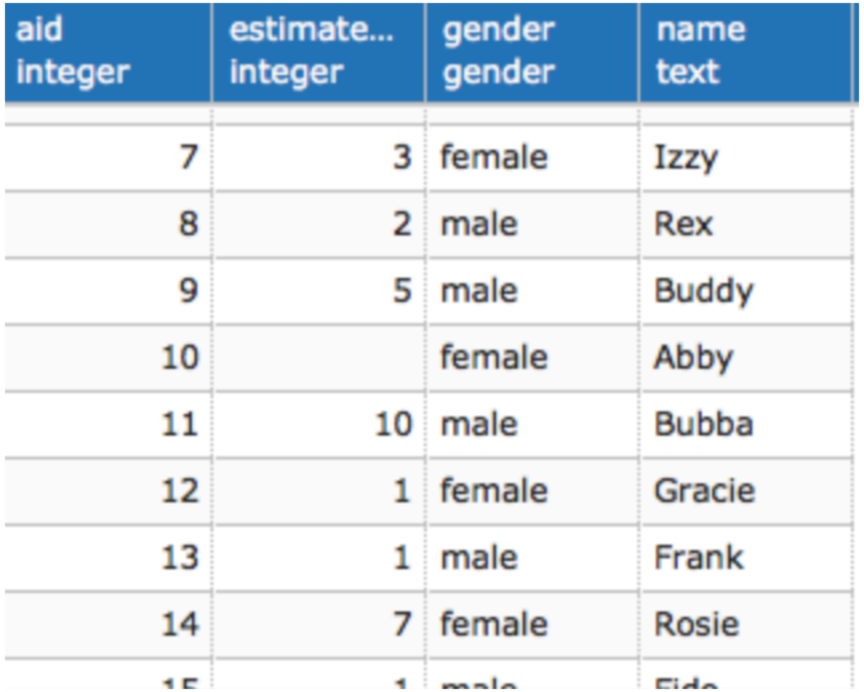
```
CREATE TRIGGER euthanize references Stored Procedure Euthanized
AFTER INSERT ON VetVisit
FOR EACH ROW
EXECUTE PROCEDURE euthanized();
```

before



| aid<br>integer | estimate...<br>integer | gender<br>gender | name<br>text |
|---|---|---|---|
| 4 | 6 | female | Carmel |
| 5 | 8 | male | Tank |
| 6 | 3 | female | Bella |
| 7 | 3 | female | Izzy |
| 8 | 2 | male | Rex |
| 9 | 5 | male | Buddy |
| 10 | 4 | female | Abby |
| 11 | 10 | male | Bubba |
| 12 | 1 | female | Gracie |
| 13 | 1 | male | Frank |
| 14 | 7 | female | Rosie |

after



| aid<br>integer | estimate...<br>integer | gender<br>gender | name<br>text |
|---|---|---|---|
| 7 | 3 | female | Izzy |
| 8 | 2 | male | Rex |
| 9 | 5 | male | Buddy |
| 10 |  | female | Abby |
| 11 | 10 | male | Bubba |
| 12 | 1 | female | Gracie |
| 13 | 1 | male | Frank |
| 14 | 7 | female | Rosie |
| 15 | 1 | male | Fido |

# ROLES

- Admin: It is the admin’s job to maintain the database, so they have full access

  create role admin;
  grant all on all tables in schema public to admin;

- Staff: It is the staff’s job to update the tables as needed

  create role staff;
  grant all on all tables in schema public to admin;

- Volunteer: It is the volunteers job to enter their information, as well as info for walks and adoptions

  create role volunteer;
  revoke all on all tables in schema public from volunteer;
  grant select on all tables in schema public to volunteer;
  grant insert on Volunteers, People, Shifts, Walks, Adoptions, Dogs, Cats to volunteer;
  grant update on Volunteers, People, Shifts, Walks, Adoptions, Dogs, Cats to volunteer;

- Vet: It is the vet’s role to take care of all the animals and to update the ReasonsFor table as needed

  create role vet;
  revoke all on all tables in schema public from vet;
  grant select on Animals, ReasonFor, Shifts, People to vet;
  grant update on ReasonFor, Shifts, People to vet;
  grant delete on ReasonFor to vet;
  grant insert on ReasonFor to vet;

# IMPLEMENTATION NOTES

When making this database, the Shelter had no data from an old system, as the shelter is a new business and thus had no records to save. Therefore, the only data in the system now is the animals that have arrived since the shelter opened.

Due to the need for record-keeping, any animals that have been put down the estimatedAges are changed to null in order to keep them in the system but not available for adoption.

In order to volunteer to walk the dogs, the volunteer must be 18, which is where the oldEnough() function comes in, as children can still be a volunteer and do other tasks at the shelter.

As an incentive to get more volunteers, the shelter implemented a program that whoever walks the most dogs gets a prize whenever the shelter decides, thus creating the need for the MostWalks function.

To check if payroll/hours are correct, the calculateTotalHours() was created.

# KNOWN PROBLEMS/FUTURE ENHANCEMENTS

- Other tables, such as EuthanizedAnimals and returnedAnimals could be added to the database, as sometimes adoptions do not work out or are ‘unadoptable’. Other attributes such as height and weight could be added to animal tables
- Adding  fostered/fosterer tables would be a good idea, as fostering helps animals adapt to a home environment and gives them extra love and attention that they may need, especially if they react badly to being in the shelter environment.
- There is no way currently to check if the volunteers walking the dogs are actually 18 or are just volunteers in general when entered into the walks table.
- Adopters should also be 18+, but nothing has been implemented to check that.
- There should be a way to add a dog/cat breed if it is not currently in the system when a dog/cat is entered into the system