Bully Boy Terrier Kennel

Database proposal by Gary Coltrane

Table of Contents

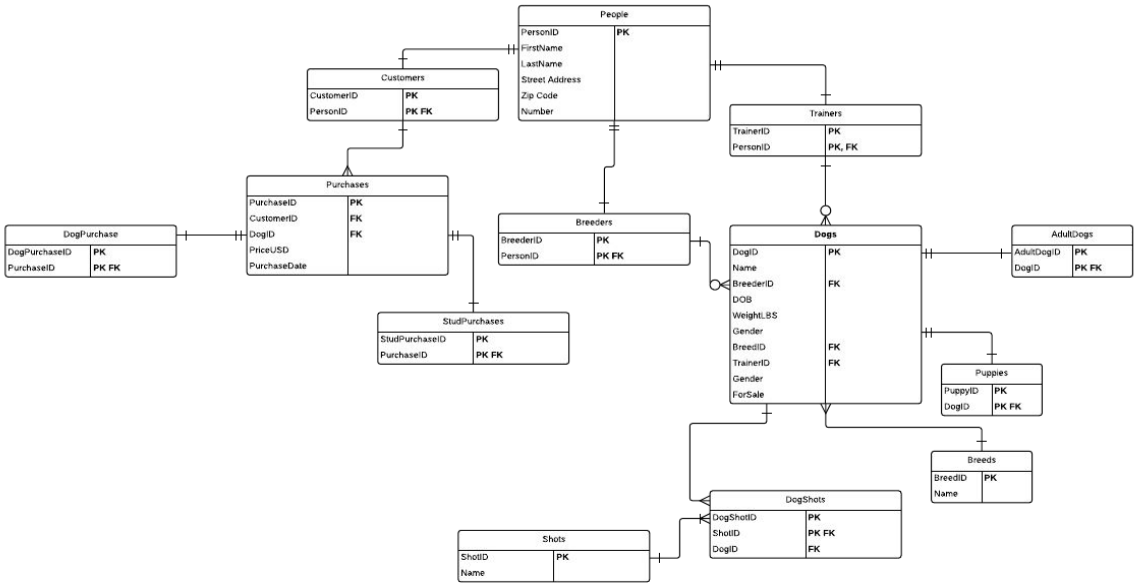
Executive Summary

Bully Boy Terrier Kennel specializes in breeding high quality dogs within the Bull family.

For safety purpose, Bully Boy Terrier pride themselves in registering each dog in AKC and providing dated shots for each dog within our kennel.

THIS document HEREBY represents the database proposal for BULLY BOY TERRIER KENNEL. Bully Boy Terrier specializes in breeding, training, and distributing dogs within the United States. In order to preserve dog safety, and quality customer service, this database must be issued. The purpose of this database is to represent that Bully Boy Terrier cares for their family of dogs, in order to ensure that the customers and team are guaranteed with high quality standards. Each table, trigger, and view are represented as a mock up based on Bully Boy Terrier sales and transactions.

Entity Relationship Diagram



Tables

People - **Stores all of the information that have had some sort of interaction or in affiliation with Bully Boy Terrier Kennel**

```
CREATE TABLE People(
        PersonID char(8) not null unique,
        FirstName text not null,
        LastName text not null,
        StreetAddress text not null,
        ZipCode integer not null,
        primary key (PersonId)
);
```

personid character	firstname text	lastname text	streetadd... text	zipcode integer
P0000001	Byron	Hardaway	123 Cherr...	104422
P0000002	Gary	Coltrane	456 Ave	902232
P0000003	Bryant	Hardaway	1800 Stre...	949302
P0000004	Bryce	Hardaway	123 Block...	19221
P0000005	Abdul	Jones	777 Ali Pl...	29932
P0000006	Sharif	Jawara	176 Sheri...	292902
P0000007	Amar	Mourning	63 Claren...	22012
P0000008	Muhamm...	Shaban	1911 E 1...	291012
P0000009	Peewee	Slapper	573 No R...	10203
P0000010	Tyshawn	Carter	424 Key ...	99102

Functional Dependencies

PersonID - > FirstName, LastName, StreetAddress, ZipCode, Number

Tables

Trainers - Stores all of the trainers that have trained dogs that are apart of Bully Boy Terrier Kennel

```
CREATE TABLE Trainers(
  TrainerID char(8) not null unique,
  PersonID char(8) not null unique,
  primary key (TrainerID, PersonID),
  foreign key (PersonId) references people(PersonID)
);
```

trainerid character	personid character
T0000001	P0000001
T0000002	P0000003
T0000003	P0000005
T0000004	P0000008
T0000005	P0000012
T0000006	P0000014
T0000007	P0000027

Functional Dependencies

TrainerID, PersonID - >

Tables

Customers - **Stores all of the customers that have bought a dog or a stud from Bully Boy Terrier Kennel**

```
CREATE TABLE Customers(
  CustomerID char(8) not null unique,
  PersonID char(8) not null unique,
  primary key (CustomerID, PersonID),
  foreign key (PersonID) references people(PersonID)
);
```

customerid character	personid character
C0000001	P0000007
C0000002	P0000009
C0000003	P0000010
C0000004	P0000011
C0000005	P0000013
C0000006	P0000015
C0000007	P0000016
C0000008	P0000017
C0000009	P0000018
C0000010	P0000019
C0000011	P0000020
C0000012	P0000021
C0000013	P0000022
C0000014	P0000023
C0000015	P0000024

Functional Dependencies

CustomerID, PersonID ->

Tables

Breeders - Stores all of the breeders that have breeded specific dogs in Bully Boy Terrier Kennel

```
CREATE TABLE Breeders(
        BreederID char(8) not null unique,
        PersonID char(8) not null unique,
        primary key (BreederID, PersonID),
        foreign key(PersonId) references people(PersonID)
);
```

breederid character	personid character
B0000001	P0000001
B0000002	P0000003
B0000003	P0000002
B0000004	P0000004
B0000005	P0000005
B0000006	P0000025
B0000007	P0000026

Functional Dependencies

BreederID, PersonID - >

Tables

Dogs - Stores all of the dogs that belong to Bully Boy Terrier Kennel

```
CREATE TABLE Dogs(
        DogID char(8) not null unique,
        name text not null,
        BreederID char(8),
        BreedID char(8) not null,
        TrainerID char(8),
        DOB date,
        WeightLBS integer,
        gender text check (gender in ('MALE', 'FEMALE')),
        ForSale text check (ForSale in ('SOLD', 'FALSE', 'TRUE')),
        primary key (DogID),
        foreign key (BreederID) references Breeders(BreederID),
        foreign key (BreedID) references Breeds(BreedID),
        foreign key (TrainerID) references Trainers (TrainerID)
);
```

Functional Dependencies

DogID - > Name, BreederID, BreedID, TrainerID, DOB, WeightLBS, Gender, ForSale

dogid character	name text	breederid character	breedid character	trainerid character	dob date	weightlbs integer	gender text	forsale text
D0000001	Bronx	B0000001	BR000002	T0000001	2012-10-...	50	MALE	FALSE
D0000002	Chrome	B0000001	BR000002	T0000001	2015-09-...	68	MALE	FALSE
D0000003	Snatch	B0000001	BR000001	T0000001	2012-05-...	70	MALE	FALSE
D0000004	Gucci	B0000001	BR000001	T0000001	2015-10-...	50	MALE	FALSE
D0000005	Prada	B0000001	BR000001	T0000001	2011-10-...	80	MALE	FALSE
D0000006	Sandy	B0000002	BR000003	T0000002	2015-02-...	100	MALE	FALSE
D0000007	Danger	B0000003	BR000003	T0000001	2013-10-...	62	MALE	FALSE
D0000008	Chanel	B0000003	BR000003	T0000004	2017-03-...	20	FEMALE	TRUE
D0000009	Banga	B0000004	BR000001	T0000002	2015-05-...	59	FEMALE	SOLD
D0000010	Nino	B0000005	BR000004	T0000002	2017-03-...	33	MALE	FALSE
D0000011	Bow Leg ...	B0000002	BR000004	T0000005	2012-05-...	58	FEMALE	FALSE
D0000012	Kurt	B0000004	BR000004	T0000006	2015-01-...	45	MALE	FALSE
D0000013	AK	B0000004	BR000002	T0000003	2013-02-...	63	MALE	FALSE
D0000014	Yaya	B0000001	BR000002	T0000005	2015-11-...	70	FEMALE	FALSE
D0000015	Richie	B0000001	BR000001	T0000001	2017-01-...	30	MALE	FALSE

Tables

Adult Dogs - Stores all of the Adult Dogs in Bully Boy Terrier Kennel

```
CREATE TABLE AdultDogs(
  AdultDogID char(8) not null unique,
  DogID char(8) not null,
  primary key (AdultDogID, DogID),
  foreign key (DogID) references Dogs(DogId)
);
```

adultdogid character	dogid character
AD000001	D0000001
AD000002	D0000002
AD000003	D0000003
AD000004	D0000004
AD000005	D0000005
AD000006	D0000006
AD000007	D0000007
AD000008	D0000009
AD000009	D0000011
AD000010	D0000012
AD000011	D0000013
AD000012	D0000014
AD000013	D0000018

Functional Dependencies

AdultDogID, DogID - >

Tables

Puppies - Stores all of the puppies that are apart of Bully Boy Terrier Kennel

```
CREATE TABLE Puppies(
  PuppyID char(8) not null unique,
  DogID char(8) not null,
  primary key (PuppyID, DogID),
  foreign key (DogID) references Dogs(DogId)
);
```

puppyid character	dogid character
PP000001	D0000008
PP000002	D0000010
PP000003	D0000015
PP000004	D0000016
PP000005	D0000017

Functional Dependencies

PuppyID, DogID - >

Tables

Purchases - Stores all of the purchases of Bully Boy Terrier Kennel

```
CREATE TABLE Purchases(
        PurchaseID   char(8) not null,
        CustomerID   char(8) not null,
        DogID        char(8) not null,
        PriceUSD     integer,
        PurchaseDate  date,
        primary key (PurchaseID),
        foreign key (CustomerID) references Customers(CustomerID),
        foreign key (DogID) references Dogs(DogId)
);
```

purchaseid character	customerid character	dogid character	priceusd integer	purchase... date
PU000001	C0000001	D0000009	1000	2017-04-...
PU000002	C0000002	D0000001	1200	2017-04-...
PU000003	C0000003	D0000003	950	2017-04-...
PU000004	C0000004	D0000007	990	2017-04-...
PU000005	C0000004	D0000011	1000	2017-04-...
PU000006	C0000005	D0000008	820	2017-04-...
PU000007	C0000006	D0000018	730	2017-04-...
PU000008	C0000007	D0000015	730	2017-04-...
PU000009	C0000008	D0000016	550	2017-04-...
PU000010	C0000009	D0000017	780	2017-04-...

Functional Dependencies

PurchaseID - > CustomerID, DogID, PriceUSD, PurchaseDate

Tables

DogPurchases - All of the dog purchases of Bully Boy Terrier Dogs

```
CREATE TABLE DogPurchases(
    DogPurchaseID char(8) not null unique,
    PurchaseID      char(8) not null,
    primary key (DogPurchaseID, PurchaseID),
    foreign key (PurchaseID) references Purchases (PurchaseID)
);
```

dogpurch... character	purchaseid character
DP000001	PU000005
DP000002	PU000014
DP000003	PU000005
DP000004	PU000008
DP000005	PU000009
DP000006	PU000010

Functional Dependencies

DogPurchaseID, PurchaseID - >

Tables

StudPurchases - All of the stud purchases of Bully Boy Terrier Dogs

```
CREATE TABLE StudPurchases(
    StudPurchaseID char(8) not null unique,
    PurchaseID     char(8) not null,
    primary key (StudPurchaseID, PurchaseID),
    foreign key (PurchaseID) references Purchases (PurchaseID)
);
```

studpurc... character	purchaseid character
SP000001	PU000002
SP000002	PU000013
SP000003	PU000016
SP000004	PU000012
SP000005	PU000007

Functional Dependencies

StudPurchaseID, PurchaseID - >

Tables

Shots - All of the shots that a dog can possibly have

```
CREATE TABLE Shots(
  ShotID  char(8) not null unique,
  name text,
  primary key (ShotID)
);
```

shotid character	name text
S0000001	Rabies
S0000002	Lyme
S0000003	Bordetella
S0000004	Giardia

Functional Dependencies

ShotID - > Name

Tables

DogShots - Each time a dog receives a shot, it gets stored into the database

```
CREATE TABLE DogShots(
  DogShotID char(8) not null unique,
  ShotID char(8) not null,
  DogID char(8) not null,
  ShotDate date,
  primary key (DogShotID, ShotID),
  foreign key (ShotID) references Shots (ShotID),
  foreign key (DogID) references Dogs (DogID)
);
```

dogshotid character	shotid character	dogid character	shotdate date
DS000001	S0000001	D0000001	2017-05-...
DS000002	S0000003	D0000004	2016-12-...
DS000003	S0000004	D0000003	2015-12-...
DS000004	S0000001	D0000001	2017-04-...
DS000005	S0000002	D0000001	2017-05-...
DS000006	S0000002	D0000011	2016-03-...

Functional Dependencies

DogShotID, ShotID -> DogID , ShotDate

Views

Available Puppies For Sale - Displays each puppy that is up for sale

```
CREATE VIEW AvailablePuppiesForSale  AS
SELECT name, weightlbs, gender from dogs inner join puppies on dogs.dogid = puppies.dogid
WHERE dogs.forsale = 'TRUE';
```

Dated shots- Displays all of the dogs that have received a shot within the year

```
CREATE VIEW  DatedShots AS
select dogs.dogid, dogs.name as DogName, shots.name as ShotName, dogshots.shotdate as
OutDatedShot from dogs
inner join dogshots on dogshots.dogid = dogs.dogid
inner join shots on dogshots.shotid = shots.shotid
where  DATE_PART('year', dogshots.shotdate)  = '2017' group by dogs.dogid, dogshots.shotdate,
shots.name;
```

Views

Available Puppies For Sale - Sample output

name text	weightlbs integer	gender text
Chanel	20	FEMALE
Nina	26	FEMALE
Robbie	20	MALE

Dated Shots- Sample output

dogid character	dogname text	shotname text	datedshot date
D0000001	Bronx	Rabies	2017-04-...
D0000001	Bronx	Lyme	2017-05-...
D0000001	Bronx	Rabies	2017-05-...

Reports

Most Expensive Stud Trainer- Displays the trainer that have sold the most expensive stud

```
SELECT people, people.firstname, people.lastname, purchases.priceusd from people inner join Trainers
  on trainers.personid = people.Personid inner join dogs
  on dogs.trainerid = trainers.trainerid
  inner join purchases on purchases.dogid = dogs.dogid inner join studpurchases on
  purchases.purchaseid = studpurchases.PurchaseID
  where purchases.priceusd = (select max(priceusd) from purchases);
```

Reports

Above Average Dogs- Displays the dogs that have been sold for above average

```
SELECT  dogs.name, purchases.priceusd from dogs
 inner join purchases on purchases.dogid = dogs.dogid
inner join dogpurchases on purchases.purchaseid = dogpurchases.purchaseid
where purchases.priceusd >  (SELECT AVG(purchases.priceusd) from purchases) group by dogs.name,
purchases.priceusd;
```

Reports

Most Expensive Stud Trainer- Sample output

people people	firstname text	lastname text	priceusd integer
(P000000...	Byron	Hardaway	1500

Above Average Dogs- Displays the dogs that have been sold for above average

name text	priceusd integer
Bow Leg ...	1000

Stored Procedures

All Shots - Returns the date and name of all of the shots for a specific dog.

```
CREATE OR REPLACE FUNCTION get_dog_shots (char(8), REFCURSOR) returns refcursor as
$$
 DECLARE
   DoggyID   char(8)     := $1;
   results     REFCURSOR   := $2;
 BEGIN
     OPEN results for
     select shots.name as ShotName, dogshots.shotdate from shots inner join dogshots on shots.shotid = dogshots.shotid
     where dogshots.dogid = DoggyID;
 return results;
end;
 $$
language plpgsql;
```

Stored Procedures

Breeder Dogs - Returns all of the dogs that have been breeded by a specific breeder.

```
CREATE OR REPLACE FUNCTION get_breeders_dogs (char(8), REFCURSOR) returns refcursor as
$$
 DECLARE
   breeder_ID   char(8)     := $1;
   results     REFCURSOR   := $2;
 BEGIN
     OPEN results for
     SELECT dogs.dogid, dogs.name from people  inner join breeders on people.personid = breeders.personid
     inner join dogs on dogs.breederid= breeders.breederid
     where breeders.breederid = breeder_ID;
 return results;
end;
 $$
language plpgsql;
```

Stored Procedures

Puppy Age - Returns the age of a specific puppy

```
CREATE OR REPLACE FUNCTION getPuppyAge(char(8))
returns interval as
        $$
DECLARE
        pupID char(8) := $1;
        birthday date := (select dogs.dob from dogs inner join puppies on dogs.dogid = puppies.dogid where puppies.PUPPYid =
PupID);
BEGIN
return age(birthday);
end;
 $$
 language plpgsql;
```

Stored Procedures

Set Dog As Sold - Sets the for sale status of a puppy to be sold, whenever it is sold to a specific customer

```
CREATE OR REPLACE FUNCTION setDogAsSold()
returns trigger as $$
BEGIN
  if new.dogid is not null then
    update dogs
      set ForSale = 'SOLD'
      where dogs.dogid = New.dogid and dogs.forsale = ‘TRUE’;
  end if;
  return new;
end;
$$ language plpgsql;
```

Trigger

Set Puppy As Sold - Whenever a there's a new purchased, whether it be a stud or a dog, the status will be set as sold.

```
Create Trigger soldDog
        after INSERT on Purchases
        for each row
execute procedure setDogAsSold();
```

Security

Bully Boy Terrier Administrator

All Administrators are each individual who have exclusive rights to Bully Boy Terrier

CREATE ROLE BBT_Admin;
GRANT ALL ON ALL TABLES IN SCHEMA PUBLIC TO BBT_Admin;

Trainers

Trainers specialize in monitoring each dog in Bully Boy Terrier. All of their rights to purchases and transactions are prohibited, because they are not allowed to sell or stud any dog.

CREATE ROLE TRAINER;
GRANT SELECT, INSERT, UPDATE, DELETE on dogs, puppies, adultdogs TO TRAINER;

Security

Breeders

Breeders are responsible to breed each dog, promote, and sell to each customer. Thus, they have access to each customer, dog, dog breeds, shots, and purchase information.

CREATE ROLE Breeder;

GRANT SELECT, INSERT , UPDATE, DELETE ON customers, dogs, puppies, adultdogs, purchases, dogpurchases, studpurchases, breeds, shots, dogshots TO Breeder ;

Notes/Enhancement/Issues

- Zip Code and Street Address must be saved in order to keep track of anybody who has purchased or is affiliated with Bully Boy Terrier
- When adding a new dog, make sure that the breed is listed within the Breeds table
- Implementing a parent and ancestor system for each dog in order to keep track of lineage would be highly essential for users.
- As of right now, Bully Boy Terrier breeds dogs of the bully family, however as the business expands, different breeding families will be added.
- Dogs in the kennel can perform in various shows, which would have a table or set of tables that track the show, date of the show, dog that performed, and outcome of the show.
- Age and weight doesn't determine if the dog is a puppy or an adult. Since each breeder specializes in identifying the dog, he or she must specifically determine whether the dog is a puppy or an adult.