Bully Boy Terrier Kennel

Database proposal by Gary Coltrane

Table of Contents

Executive Summary

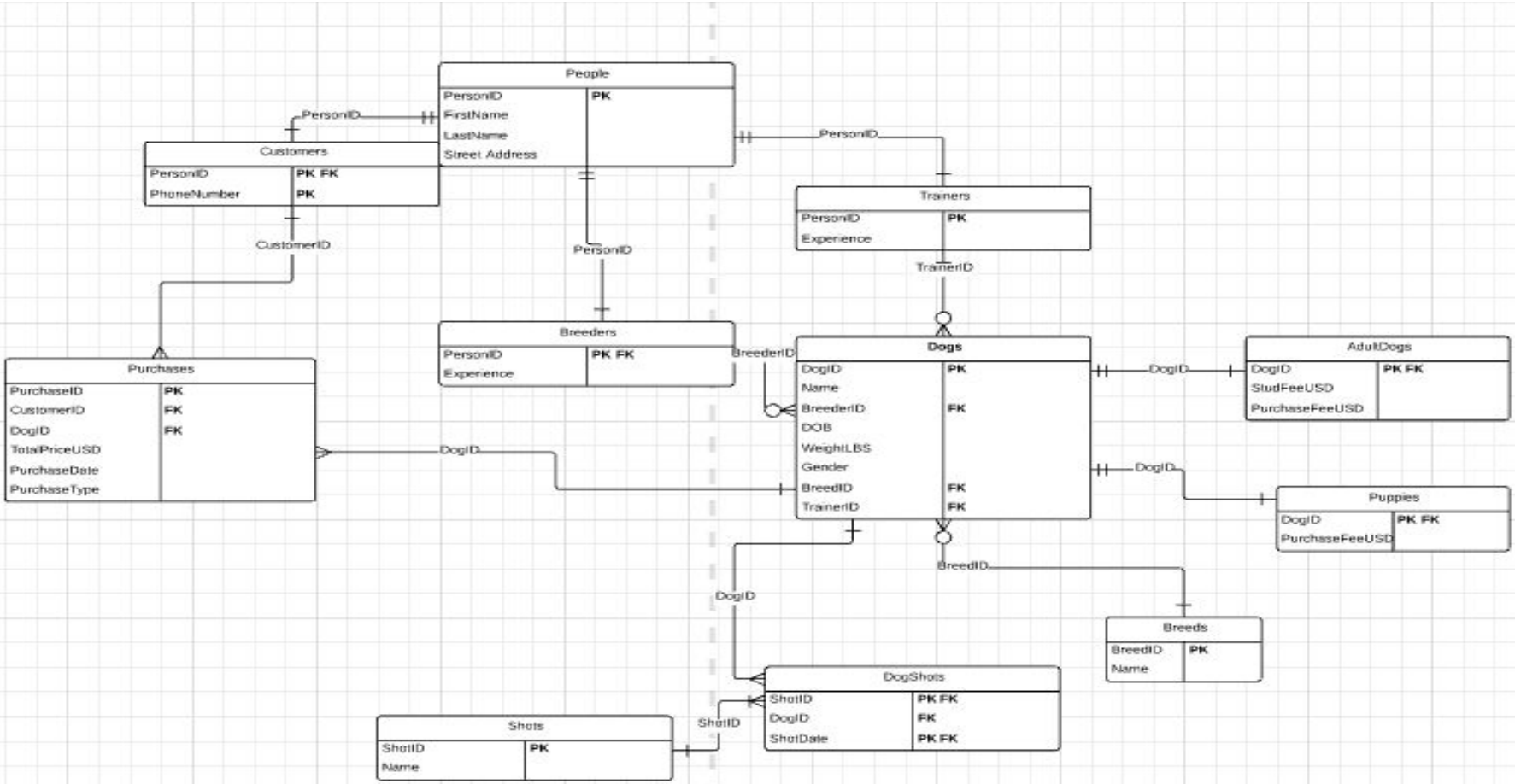
Bully Boy Terrier Kennel is a bull breeding family business that focuses on breeding, studding, training, and selling dogs within our kennel.

For safety purpose, Bully Boy Terrier pride themselves in registering each dog in the American Kennel Club 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

Bully Boy Terrier Kennel



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,
  primary key (PersonId)
);
```

personid character	firstname text	lastname text	streetadd... text
P0000001	Byron	Hardaway	123 Cherr...
P0000002	Gary	Coltrane	456 Ave
P0000003	Bryant	Hardaway	1800 Stre...
P0000004	Bryce	Hardaway	123 Block...

Functional Dependencies

PersonID - > FirstName LastName StreetAddress

Tables

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

```
CREATE TABLE Trainers(
  PersonID char(8) not null unique,
  Experience text check (Experience in ('BEGINNER', 'INTERMEDIATE', 'ADVANCED')),
  primary key (PersonID),
  foreign key (PersonId) references People(PersonID)
);
```

personid character	experience text
P0000001	ADVANCED
P0000003	INTERME...
P0000005	INTERME...
P0000008	INTERME...
P0000012	INTERME...
P0000014	ADVANCED
P0000027	INTERME...

Functional Dependencies

PersonID -> Experience

Tables

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

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

personid character	phonenu... text
P0000007	3478192...
P0000009	1874191...
P0000010	7187773...
P0000011	6467770...
P0000013	7187772...
P0000015	3890927...
P0000016	4593833...
P0000017	2932020...

Functional Dependencies

PersonID -> PhoneNumber

Tables

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

```
CREATE TABLE Breeders(
        PersonID char(8) not null unique,
        Experience text check (Experience in ('BEGINNER', 'INTERMEDIATE', 'ADVANCED')),
        primary key (PersonID),
        foreign key(PersonId) references people(PersonID)
);
```

Query Sample Note: **Selected personID as breederid**

breederid character	experience text
P0000001	ADVANCED
P0000003	ADVANCED
P0000002	INTERME...
P0000004	ADVANCED
P0000005	BEGINNER
P0000025	ADVANCED
P0000026	ADVANCED

Functional Dependencies

PersonID -> Experience

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) ,
  TrainerID char(8),
  DOB date,
  WeightLBS int,
  gender text check (gender in ('MALE', 'FEMALE')),
  primary key (DogID),
  foreign key (BreederID) references Breeders(PersonID),
  foreign key (TrainerID) references Trainers(PersonID),
  foreign key (BreedID) references Breeds(BreedID)
);
```

Functional Dependencies

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

dogid character	name text	breederid character	breedid character	trainerid character	dob date	weightlbs integer	gender text
D0000001	Bronx	P0000001	BR000002	P0000001	2012-10-...	50	MALE
D0000002	Chrome	P0000001	BR000002	P0000001	2015-09-...	68	MALE
D0000003	Snatch	P0000001	BR000001	P0000001	2012-05-...	70	MALE
D0000004	Gucci	P0000001	BR000001	P0000001	2015-10-...	50	MALE

Tables

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

```
CREATE TABLE AdultDogs(
  DogID char(8) not null,
  PurchaseFeeUSD  integer,
  StudFeeUSD  integer,
  primary key (DogID),
  foreign key (DogID) references Dogs(DogID)
);
```

dogid character	purchase... integer	studfeeusd integer
D0000001	3000	1500
D0000002	4000	2000
D0000003	2000	900
D0000004	800	300
D0000005	900	300
D0000006	900	300
D0000007	1000	500
D0000009	900	120
D0000011	500	120
D0000012	1000	750

Functional Dependencies

DogID - > PurchaseFeeUSD, StudFeeUSD

Tables

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

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

dogid character	purchase... integer
D0000008	500
D0000010	900
D0000015	600
D0000016	500
D0000017	700

Functional Dependencies

DogID - > PurchaseFeeUSD

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,
  TotalPriceUSD      integer,
  PurchaseDate  date,
  PurchaseType text check (PurchaseType in ('STUD', 'ENTIRE DOG')),
  primary key (PurchaseID),
  foreign key (CustomerID) references Customers(PersonID),
  foreign key (DogID) references Dogs(DogId)
);
```

purchaseid character	customerid character	dogid character	totalprice... integer	purchase... date	purchase... text
PU000001	P0000007	D0000009	1790	2017-04-...	STUD
PU000002	P0000024	D0000001	1700	2017-04-...	STUD
PU000003	P0000009	D0000003	950	2017-04-...	STUD
PU000004	P0000024	D0000007	690	2017-04-...	STUD
PU000005	P0000010	D0000011	600	2017-04-...	ENTIRE D...
PU000006	P0000016	D0000008	620	2017-04-...	ENTIRE D...

Functional Dependencies

PurchaseID -> CustomerID, DogID, TotalPriceUSD, PurchaseDate, PurchaseType

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(

  ShotID char(8) not null,
  DogID char(8) not null,
  ShotDate date not null,
  primary key (ShotID, ShotDate),
  foreign key (ShotID) references Shots (ShotID),
  foreign key (DogID) references Dogs (DogID)
);
```

shotid character	dogid character	shotdate date
S0000001	D0000001	2017-05-...
S0000003	D0000004	2016-12-...
S0000004	D0000003	2015-12-...
S0000001	D0000001	2017-04-...

Functional Dependencies

ShotID, ShotDate - > DogID

Views

Available Puppies By Breed- Displays each puppy for a specific breed. In this case the breed is a pitbull

```
CREATE VIEW AvailablePuppiesByBreed  AS
SELECT name, weightlbs, gender from dogs inner join puppies on dogs.dogid = puppies.dogid
WHERE  dogs.breedid = 'BR000004';
```

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.totalpriceusd from people inner join
Trainers
  on trainers.personid = people.Personid inner join dogs
  on dogs.trainerid = trainers.personid
  inner join purchases on purchases.dogid = dogs.dogid
  where purchases.totalpriceusd = (select max(totalpriceusd) from purchases);
```

Reports

Above Average Dogs- Displays the dogs that have a purchase fee above average

```
SELECT  dogs.name, adultdogs.purchasefeeusd from dogs
 inner join adultdogs on adultdogs.dogid = dogs.dogid
where adultdogs.purchasefeeusd >         (SELECT AVG(adultdogs.purchasefeeusd) from adultdogs)
group by dogs.name, adultdogs.purchasefeeusd;
```

Reports

Most Expensive Stud Trainer- Sample output

people people	firstname text	lastname text	totalprice... integer
(P000000...	Byron	Hardaway	2000

Above Average Dogs- Sample output

name text	purchase... integer
Chrome	4000
Snatch	2000
Bronx	3000

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

Stored Procedures

Puppy Age - Returns the age of a specific puppy

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

Stored Procedures

Set Trainer as Immediate - Sets the trainer experience level to immediate

```
CREATE OR REPLACE FUNCTION setTrainerAsIntermediate()
 returns trigger as $$
 BEGIN
   if new.trainerid is not null then
     update trainers
       set Experience = 'IMMEDIATE'
       where trainers.personid = new.trainerid and trainers.experience = ‘BEGINNER’;
   end if;
   return new;
 end;
 $$ language plpgsql;
```

Trigger

Set Trainer as Immediate - Every dog in Bully Boy Terrier Kennel must be trained by a trainer who has atleast immediate experience, therefore if a trainer is a beginner, the trigger would set him or her as an immediate.

```
Create Trigger IntermediateTrainer
after INSERT on dogs
for each row
execute procedure setTrainerAsIntermediate();
```

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

- 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.