**SQL commands to create database:**

create database dogproject;

use database dogproject;

**create dog table:**

non\_SQL step: import using import wizard, name columns appropriately

ALTER TABLE dog add column `dog\_id` int unsigned primary KEY AUTO\_INCREMENT FIRST;

**add weight to dog table:**

ALTER TABLE dog add column `dog\_weight` int unsigned AFTER dog\_breed;

SET SQL\_SAFE\_UPDATES = 0;

update dog set dog\_weight = 15 where dog\_breed= "Yorkshire Terrier";

update dog set dog\_weight = 12 where dog\_breed= "Shih Tzu";

update dog set dog\_weight = 30 where dog\_breed= "Labrador retriever Crossbreed";

update dog set dog\_weight = 14 where dog\_breed= "Chihuahua";

update dog set dog\_weight = 25 where dog\_breed= "Pomeranian";

update dog set dog\_weight = 18 where dog\_breed<> "Unknown" and dog\_weight is NULL;

**create owner table:**

create table owner

(

owner\_id int primary KEY AUTO\_INCREMENT,

owner\_first varchar(25) NOT NULL,

owner\_last varchar(25) NOT NULL,

owner\_street varchar(40) NOT NULL,

owner\_city varchar(40) NOT NULL,

owner\_state char(2) NOT NULL,

owner\_zip char(5) NOT NULL,

owner\_email varchar(50) NOT NULL

);

**insert data into owner table:**

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values

("Jesper", "Bratt", "50 Pine St", "Jamestown", "NY", "11840", "jesperbratt@gmail.com");

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values

("Nico", "Hischier", "40 Pine St", "Jamestown", "NY", "11840", "jesperbratt@gmail.com"),

("Jack", "Hughes", "17 Ranger Rd", "Westchester", "NY", "40404", "iseehughes@aol.com"),

("Damon", "Severson", "19 Ranger Rd", "Westchester", "NY", "40404", "dsev28@hotmail.com"),

("Douglas", "Hamilton", "400 Flame Ave", "Albany", "NY", "10200", "dham@gmail.com"),

("Ondrej", "Palat", "60 Pine St", "Jamestown", "NY", "11840", "opot@yahoo.com");

update owner set owner\_email = "niconumberone@nhl.com" where owner\_id= 2;

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values

("Patrick", "Elias", "26 Ocean Ave", "Belmont", "NY", "26262", "czechyourself@gmail.com"),

("Ilya", "Kovalchuk", "17 Ocean Ave", "Belmont", "NY", "26262", "stashedindeep@aol.com"),

("Adam", "Henrique", "99 Legends Ln", "Brooklyn", "NY", "10423", "itsover@hotmail.com"),

("Martin", "Brodeur", "30 Goalie Rd", "Brooklyn", "NY", "21200", "numberone@gmail.com"),

("Cory", "Schneider", "35 Goalie Rd", "Brooklyn", "NY", "21200", "schneidsy35@yahoo.com");

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values ("Dawson", "Mercer", "91 Drury Ln", "Buffalo", "NY", "16860", "havemercer@gmail.com"), ("Miles", "Wood", "44 Speed St", "Buffalo", "NY", "16860", "needforspeed@aol.com"), ("Nathan", "Bastian", "14 Kraken Ct", "Utica", "NY", "19091", "specialteams@hotmail.com"), ("Mackenzie", "Blackwood", "29 Goalie Rd", "Brooklyn", "NY", "21200", "justmakeasave@gmail.com"), ("Tomas", "Tatar", "90 French Rd", "Yonkers", "NY", "10011", "tatar90@yahoo.com")

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values

("John", "Marino", "6 Sullivan St", "Newbridge", "NY", "16760", "marino6john@gmail.com"),

("PK", "Subban", "8 Sullivan St", "Newbridge", "NY", "16760", "slewban@aol.com"),

("Ryan", "Graves", "33 President St", "Plattsburgh", "NY", "12222", "graveerror@hotmail.com"),

("Bobby", "Butler", "9 Butler Dr", "Flushing", "NY", "15500", "atyourservice@gmail.com"),

("Andy", "Greene", "6 Island Rd", "Plainview", "NY", "20100", "[captaingreene@yahoo.com](mailto:captaingreene@yahoo.com)");

INSERT into owner (owner\_first, owner\_last, owner\_street, owner\_city, owner\_state, owner\_zip, owner\_email) values

("Joe", "Smith", "35 Ocean Ave", "Belmont", "NY", "26262", "russianyourself@gmail.com");

**create owner\_phone table:**

create table owner\_phone(

owner\_id int,

phone\_number char(10),

FOREIGN KEY (owner\_id) REFERENCES owner(owner\_id),

PRIMARY KEY (owner\_id, phone\_number)

);

**insert data into owner\_phone:**

INSERT into owner\_phone values

(5, "9174859384");

INSERT into owner\_phone values

(5, "8925894835"),

(5, "2385475972"),

(8, "5284829805"),

(1, "9913123469"),

(15, "1587266616"),

(19, "2088490639"),

(19, "0808322696"),

(2, "4333149875"),

(7, "7053565306"),

(11, "0633306832"),

(11, "2814499519"),

(10, "1919812483"),

(13, "5310058061"),

(18, "1178372833");

**create owner\_dog table:**

create table owner\_dog(

owner\_id int,

FOREIGN KEY (owner\_id) REFERENCES owner(owner\_id),

dog\_id int unsigned,

FOREIGN KEY (dog\_id) REFERENCES dog(dog\_id),

PRIMARY KEY (owner\_id, dog\_id)

);

**insert data into owner\_dog:**

INSERT into owner\_dog values

(1, 1),

(2, 2),

(3, 3),

(4, 4),

(5, 5),

(6, 6),

(7, 7),

(8, 8),

(9, 9),

(10, 10),

(11, 11),

(12, 12),

(13, 13),

(14, 14),

(15, 15),

(16, 16),

(17, 17),

(18, 18),

(19, 19),

(20, 20),

(21, 21);

INSERT into owner\_dog values

(1, 31),

(2, 32),

(3, 33),

(4, 34),

(5, 35),

(6, 36),

(7, 37),

(8, 38),

(9, 39),

(10, 40),

(11, 41);

INSERT into owner\_dog values

(1, 51),

(2, 52),

(3, 53),

(4, 54),

(5, 55),

(6, 56);

INSERT into owner\_dog values

(1, 61),

(2, 62),

(3, 63);

INSERT into owner\_dog values

(1, 71),

(2, 72),

(3, 73);

INSERT into owner\_dog values

(1, 81),

(1, 82),

(1, 83),

(1, 84),

(1, 85),

(1, 86),

(1, 87);

**add start and end date for dog ownership (allowing the table to track current and previous owners of dogs):**

ALTER TABLE owner\_dog add column startdate date;

ALTER TABLE owner\_dog add column enddate date;

**insert data into owner\_dog for startdate and enddate:**

update owner\_dog set startdate = "2020-03-11" where owner\_id= 15;

update owner\_dog set startdate = "2020-03-11" where owner\_id= 16;

update owner\_dog set startdate = "2020-03-11" where owner\_id= 17;

update owner\_dog set startdate = "2021-04-11" where owner\_id= 18;

update owner\_dog set startdate = "2021-04-11" where owner\_id= 19;

update owner\_dog set startdate = "2022-04-11" where owner\_id= 20;

update owner\_dog set startdate = "2022-04-11" where owner\_id= 21;

update owner\_dog set startdate = "2022-01-01" where owner\_id= 12 or owner\_id = 13 or owner\_id = 14;

update owner\_dog set startdate = "2019-01-01" where owner\_id= 11;

update owner\_dog set startdate = "2019-07-01" where owner\_id= 10;

update owner\_dog set startdate = "2022-07-01" where owner\_id= 9;

update owner\_dog set startdate = "2022-07-01" where owner\_id= 8 and dog\_id = 8;

update owner\_dog set startdate = "2022-04-01" where owner\_id= 8 and dog\_id = 38;

update owner\_dog set startdate = "2022-04-01" where owner\_id= 7 and dog\_id = 7;

update owner\_dog set startdate = "2021-08-19" where owner\_id= 7 and dog\_id = 37;

update owner\_dog set startdate = "2021-09-19" where owner\_id= 4;

update owner\_dog set startdate = "2022-03-31" where owner\_id= 5 and dog\_id = 5;

update owner\_dog set startdate = "2022-02-25" where owner\_id= 5 and dog\_id = 35;

update owner\_dog set startdate = "2022-01-22" where owner\_id= 5 and dog\_id = 55;

update owner\_dog set startdate = "2022-10-22" where owner\_id= 6;

update owner\_dog set startdate = "2017-10-22" where owner\_id= 3 and dog\_id = 3;

update owner\_dog set startdate = "2018-10-22" where owner\_id= 3 and dog\_id = 33;

update owner\_dog set startdate = "2019-10-22" where owner\_id= 3 and dog\_id = 53;

update owner\_dog set startdate = "2020-10-22" where owner\_id= 3 and dog\_id = 63;

update owner\_dog set startdate = "2021-10-22" where owner\_id= 3 and dog\_id = 73;

update owner\_dog set startdate = "2021-06-22" where owner\_id= 2;

update owner\_dog set startdate = "2022-06-22" where owner\_id= 1;

update owner\_dog set enddate = "2022-06-22" where owner\_id= 18;

update owner\_dog set enddate = "2022-06-22" where owner\_id= 19;

INSERT into owner\_dog (owner\_id, dog\_id, startdate) values

(16, 19, "2022-06-23");

**create photo table:**

create table photo

(

photo\_id int primary KEY AUTO\_INCREMENT,

dog\_id int unsigned,

photo\_date date,

photo\_name varchar(40),

FOREIGN KEY (dog\_id) REFERENCES dog(dog\_id)

);

**insert photos:**

INSERT into photo (dog\_id, photo\_date, photo\_name) values

(1, "2022-11-02", "playingatthepark.jpg");

INSERT into photo (dog\_id, photo\_date, photo\_name) values

(1, "2022-11-03", "newhat.jpg"),

(1, "2022-11-03", "bubblebath.jpg"),

(2, "2022-11-01", "fetch.jpg"),

(2, "2022-11-02", "troublemaker.jpg"),

(3, "2022-11-03", "walk.jpg"),

(4, "2022-11-03", "pantry.jpg");

**create violation table:**

create table violation

(

violation\_id int primary KEY AUTO\_INCREMENT,

violation\_name varchar(40)

);

**insert violation names into violation table:**

insert into violation (violation\_name) values

("excessive barking"),

("biting"),

("disturbing the peace"),

("disobedient behavior"),

("no leash")

;

**create ticket table:**

create table ticket

(

ticket\_id int primary KEY AUTO\_INCREMENT,

owner\_id int,

FOREIGN KEY (owner\_id) REFERENCES owner(owner\_id),

dog\_id int unsigned,

FOREIGN KEY (dog\_id) REFERENCES dog(dog\_id),

violation\_id int,

FOREIGN KEY (violation\_id) REFERENCES violation(violation\_id),

ticket\_amount int,

ticket\_date date,

ticket\_status varchar(40)

);

**insert data into ticket:**

insert into ticket(owner\_id, dog\_id, violation\_id, ticket\_amount, ticket\_date, ticket\_status) values

(1, 1, 1, 85, "2022-06-30", "paid");

**move fine amount from ticket to violation:**

alter table ticket drop column ticket\_amount;

alter table violation add column `violation\_amount` int;

update violation set violation\_amount = 85 where violation\_id= 1;

update violation set violation\_amount = 180 where violation\_id= 2;

update violation set violation\_amount = 60 where violation\_id= 3;

update violation set violation\_amount = 50 where violation\_id= 4;

update violation set violation\_amount = 20 where violation\_id= 5;

**insert data into ticket:**

insert into ticket(owner\_id, dog\_id, violation\_id, ticket\_date, ticket\_status) values

(19, 19, 2, "2022-05-30", "paid"),

(19, 19, 2, "2022-04-15", "paid"),

(16, 19, 2, "2022-06-30", "paid"),

(1, 1, 5, "2022-08-30", "unpaid"),

(1, 1, 1, "2022-09-30", "unpaid"),

(1, 31, 5, "2022-08-30", "unpaid"),

(1, 31, 1, "2022-09-30", "disputed");

insert into ticket(owner\_id, dog\_id, violation\_id, ticket\_date, ticket\_status) values

(4, 4, 3, "2021-10-30", "paid"),

(4, 4, 3, "2022-10-31", "paid"),

(8, 8, 2, "2022-06-30", "paid"),

(11, 41, 5, "2019-08-30", "unpaid"),

(7, 37, 1, "2022-11-14", "unpaid"),

(7, 37, 4, "2021-08-30", "unpaid"),

(10, 10, 4, "2022-09-30", "disputed"),

(17, 17, 2, "2020-12-31", "disputed"),

(1, 85, 1, "2022-10-05", "paid"),

(1, 61, 5, "2022-11-01", "unpaid"),

(1, 81, 3, "2022-10-12", "unpaid");

**give every dog an owner:**

insert into owner\_dog (owner\_id, dog\_id) select owner\_id, dog\_id from dog, owner where dog\_id not in (select dog\_id from owner\_dog) and owner\_id = 22;

update owner\_dog set startdate = "2022-10-22" where owner\_id= 22;

**QUESTION 1:**

Identify dogs with owners without violations since 2020-11-03. Display the owner name, dog name, breed and email. Use a nested select to answer this question.

note: I am assuming this question is referring to dogs that have owners

SQL: select concat(owner\_first, " ", owner\_last) as `Owner Name`, dog\_name as `Dog Name`, dog\_breed as `Dog Breed`, owner\_email as `Owner Email` from dog,owner,owner\_dog where dog.dog\_id = owner\_dog.dog\_id and owner.owner\_id = owner\_dog.owner\_id and dog.dog\_id not in (select dog\_id from ticket where ticket\_date > "2020-11-03");

result:

Owner Name Dog Name Dog Breed Owner Email

Jesper Bratt CHIP Yorkshire Terrier jesperbratt@gmail.com

**QUESTION 2:** Identify zip codes that have never had registered pit bulls. Display the zip code. Remove duplicate zip codes. Use a nested select to answer this question.

SQL: select dog\_zipcode as `ZIP Codes with no current pit bull` from dog where dog\_zipcode not in (select dog\_zipcode from dog where dog.dog\_breed like "%Pit Bull%") group by dog\_zipcode order by dog\_zipcode;

result:

ZIP Codes with no current pit bull

10001

**QUESTION 3:** Identify dogs without current owners. Display the dog name, gender, breed and age. Use a nested select to answer this question.

SQL: select dog\_name as `Name`, dog\_gender as `Gender`, dog\_breed as `Breed`, 2022 - dog\_birth\_year as `Age` from dog where dog\_id not in (select dog\_id from owner\_dog where enddate is NULL);

result:

Name Gender Breed Age

JADE F Schipperkee 12

**QUESTION 4:** Identify current dog owners with a residence in Buffalo. Display the

owner name, dog name and breed.

SQL: select concat(owner\_first, " ", owner\_last) as `Buffalo Owner`, dog\_name as `Dog Name`, dog\_breed as `Dog Breed` from dog,owner,owner\_dog where dog.dog\_id = owner\_dog.dog\_id and owner.owner\_id = owner\_dog.owner\_id and owner\_city = "Buffalo";

result:

Buffalo Owner Dog Name Dog Breed

Dawson Mercer SAM Bull Dog, English

**QUESTION 5:** Identify pictures of male maltese older than five years old. Display the dog name, age and all photos.

SQL: select dog\_name as `Name`, 2022 - dog\_birth\_year as `Age`, photo\_name as `Photo Name` from dog, photo where dog.dog\_id = photo.dog\_id and dog\_breed = "Maltese" and 2022 - dog\_birth\_year > 5;

result:

Name Age Photo Name

HATCHI 11 playingatthepark.jpg

HATCHI 11 newhat.jpg

**QUESTION 6:** Identify dogs owned by Jesper Bratt with violations since 2019-11-03. Display the owner name, dog name, violation, date of violation and fine.

SQL: select concat(owner\_first, " ", owner\_last) as `Owner Name`, dog\_name as `Dog Name`, violation\_name as `Violation`, ticket\_date as `Violation Date`, CONCAT('$', violation\_amount ) as `Fine` from owner,dog,violation,ticket where owner.owner\_id = ticket.owner\_id and dog.dog\_id = ticket.dog\_id and violation.violation\_id = ticket.violation\_id and owner\_first = "Jesper" and owner\_last = "Bratt" and ticket\_date > "2019-11-03";

result:

Owner Name Dog Name Violation Violation Date Fine

Jesper Bratt HATCHI no leash 2022-08-30 $20

Jesper Bratt COCO excessive barking 2022-08-30 $85

**QUESTION 7:** Identify the number of dogs by gender. Display two columns and one row for each gender. The two output columns are gender and number of dogs with that gender. Use a function to answer this question.

SQL: select dog\_gender as `Gender`, count(\*) as `Count` from dog group by dog\_gender;

result:

Gender Count

M 640

F 526

**QUESTION 8:** Identify zip codes with the most Yorkshire Terriers. Display two columns and one row for each zip code. The two output columns are zip code and number of Yorkshire Terriers in that zip code. Display the zip code with the most Yorkshire Terriers first. Use a function to answer this question.

SQL: select dog\_zipcode as `ZIP Code`, count(\*) as `Yorkshire Terriers` from dog where dog.dog\_breed ="Yorkshire Terrier" group by dog\_zipcode order by count(\*) desc;

result:

ZIP Code Yorkshire Terriers

10314 7

10309 4

**QUESTION 9:** Identify the number of total fines by owner since 2019. Display three columns and one row for each owner. The three columns are owner name, number of violations and total dollar amount of fines. Display owners with the most fines first. Use a function to answer this question.

SQL: select concat(owner\_first," ", owner\_last) as `Owner`, count(\*) as `Number of Violations`, concat("$", sum(violation\_amount) ) as `Total Amount` from owner,ticket,violation where owner.owner\_id = ticket.owner\_id and violation.violation\_id = ticket.violation\_id and ticket\_date > "2019-12-31" group by owner\_first, owner\_last order by count(\*) desc;

result:

Owner Number of Violations Total Amount

Jesper Bratt 8 $460

Patrick Elias 2 $135

**QUESTION 10:** Increase all fines by 100%. Identify the SQL to perform this operation and the fine before and after this operation.

SQL: update violation set violation\_amount = violation\_amount \* 2;

Fine amount before (using SQL command: select violation\_name as `Name`, concat("$", violation\_amount) as `Amount` from violation;):

Name Amount

excessive barking $85

biting $180

disturbing the peace $60

disobedient behavior $50

no leash $20

Fine amount after (using SQL command: select violation\_name as `Name`, concat("$", violation\_amount) as `Amount` from violation;):

Name Amount

excessive barking $170

biting $360

disturbing the peace $120

disobedient behavior $100

no leash $40

**QUESTION 11:** Display the structure of all tables using the SQL Describe operation.

multiple steps:

step 1:

SQL: show tables;

result:

Tables\_in\_dogproject

dog

owner

owner\_dog

owner\_phone

photo

ticket

violation

step 2: for each table, perform the desc table\_name

SQL:

desc dog;

desc owner;

desc owner\_dog;

desc owner\_phone;

desc photo;

desc ticket;

desc violation;

result (for desc dog;):

Field Type Null Key Default Extra

dog\_id int unsigned NO PRI NULL auto-increment

**QUESTION 12:** Display the Oracle version by entering select \* from product\_component\_version;

the above did not work for me so instead I did

SQL:

SELECT VERSION();

result:

VERSION()

8.0.26