ADIL ABUWANI
DATABASE PROJECT 2
CSCI 331-33
April 18, 2018

Student-ID: 4504(Last Four Digits)

Adil.abuwani@outlook.com

RELATION 1: DOG_OWNER Relation provides all the information about the owner of the dog

DOG_OWNER (OWNERID(PK), OWNERFIRSTNAME, OWNERLASTNAME, OWNEREMAIL, STREET, CITY, STATE, NEIGHBORHOOD, ZIP)

OWNERID	OWNERFIRSTNAME	OWNERLASTNAME	OWNEREMAIL	STREET	CITY	STATE	NEIGHBORHOOD	ZIP
1	smith	bob	smith@gmail.com	queens blvd	Rego park	NY	Central Queens	11374
2	smith	jon	smithjon@gmail.com	queens blvd	Rego park	NY	West Central Queens	11375
3	tom	jon	tom@gmail.com	queens blvd	Rego park	NY	West Central Queens	11379
4	adil	abuwani	adil@gmail.com	eliot ave	Rego park	NY	West Central Queens	11385
5	ali	khan	khan@gmail.com	eliot ave	Rego park	NY	West Central Queens	11385
6	miky	rod	mikye@gmail.com	14 st	downtown manhattan	NY	lower manhattan	10002
7	lulu	smith	lulu@gmail.com	thomson ave	long island city	NY	Northwest Queens	11101
8	nini	lii	nini@gmail.com	thomson ave	long island city	NY	Northwest Queens	11101

PRIMARY KEY:

OWNERID

Degree = 9

DOMAINS:

OwnerID(PK): 0-9

OwnerName-Composite attribute that can be divided into further smaller components.

OwnerFirstName: a-z & A-Z

OwnerLastName: a-z & A-Z

OwnerEmail: a-z & A-Z & 0-9

OwnerLocation-Composite attribute

Street: a-z & A-Z

City: a-z & A-Z

State: a-z & A-Z

Neighborhood: a-z & A-Z

Zip: 5 digits

Third Normal form:

We know that a relation is of third normal form if it is in first normal form, second normal form, and in which no non-candidate key is transitively dependent on candidate key.

In the DogOwner relation, the intersection of each row and column has one and only one value. So, we know that the relation is of first normal form.

Secondly, in the DogOwner relation, the non-primary key is functionally dependent on the primary key.

Finally, we also observe that there exists no non-candidate key which is transitively dependent on candidate key. Hence the relation is of third normal form.

Since the dog owner can have multiple phone numbers, we will create a new relation to prevent from repetition.

SQL CREATE TABLE:

```
CREATE TABLE DOG_OWNER

   (   OWNERID NUMBER,

       OWNERFIRSTNAME VARCHAR(30),
```

```
OWNERLASTNAME VARCHAR(30),

OWNEREMAIL VARCHAR(320),

STREET VARCHAR(30),

City VARCHAR(30),

State VARCHAR(30),

Neighborhood VARCHAR(30),

Zip VARCHAR(5),

CONSTRAINT "DOG_OWNER_PK" PRIMARY KEY (OWNERID)

);
```

SQL INSERT VALUES IN TABLE:

```
create sequence SEQ_DOG_OWNER
start with 1 increment by 1;
```

```
INSERT INTO DOG_OWNER(OWNERID, OWNERFIRSTNAME, OWNERLASTNAME, OWNEREMAIL,
STREET, CITY, STATE, NEIGHBORHOOD, ZIP)
VALUES (SEQ_DOG_OWNER.NEXTVAL, 'smith', 'bob', 'smith@gmail.com', 'queens
blvd', 'Rego park', 'NY', 'Central Queens', '11374')

INSERT INTO DOG_OWNER(OWNERID, OWNERFIRSTNAME, OWNERLASTNAME, OWNEREMAIL,
STREET, CITY, STATE, NEIGHBORHOOD, ZIP)
VALUES (SEQ_DOG_OWNER.NEXTVAL, 'smith', 'jon', 'smithjon@gmail.com',
'queens blvd', 'Rego park', 'NY', 'West Central Queens', '11375')

INSERT INTO DOG_OWNER(OWNERID, OWNERFIRSTNAME, OWNERLASTNAME, OWNEREMAIL,
STREET, CITY, STATE, NEIGHBORHOOD, ZIP)
VALUES (SEQ_DOG_OWNER.NEXTVAL, 'tom', 'jon', 'tom@gmail.com', 'queens
blvd', 'Rego park', 'NY', 'West Central Queens', '11379')
```

(WE WILL INSERT ALL OWNERS WITH THE SAME APPROACH,AND MOVE TO THE NEXT RELATION)

RELATION 2: DOG_OWNER_PHONE relation has all the phone numbers of dog owners. DOG OWNER CAN HAV MULTIPLE PHONE NUMBERS.

DOG_OWNER_PHONE (OWNERID, OWNERPHONE)

OWNERID	OWNERPHONE
1	1231231225
1	1231231228
1	1234561223
2	6547896332
3	1235547878
3	1235547889
4	6547896332
5	1212451245

PRIMARY KEY:

The OWNERID, and OWNERPHONE both make a unique primary key and will use it as a composite key.

FOREIGN KEY:

OWNERID, references DOG_OWNER(OWNERID)

Degree = 2

Domains:

OWNERID: 0-9

OwnerPhone: 10 digits

SQL CREATE TABLE:

```
CREATE TABLE DOG_OWNER_PHONE
   (     OWNERID NUMBER,
         OWNERPHONE VARCHAR(15),
         CONSTRAINT "DOG_OWNER_PHONE_PK" PRIMARY KEY (OWNERID, OWNERPHONE)
   ) ;
```

SQL FOREIGN KEY:

```
ALTER TABLE DOG_OWNER_PHONE

ADD FOREIGN KEY (OWNERID) REFERENCES DOG_OWNER(OWNERID);
```

SQL INSERT VALUES:

```
INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (1, '1231231225');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (1, '1231231228');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (1, '1234561223');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (2, '6547896332');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (3, '1235547878');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (3, '1235547889');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (4, '6547896332');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (5, '1212451245');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
```

```
VALUES (6, '7878789665');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (7, '7856547885');
INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (8, '7896589663');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (9, '2222222222');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (11, '3333333333');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (12, '3476237763');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (13, '9876543221');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (14, '4563214560');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (15, '5878965445');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (15, '6547896335');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (16, '9998887777');

INSERT INTO DOG_OWNER_PHONE(OWNERID, OWNERPHONE)
VALUES (17, '8756589632');
```

RELATION 3: ANIMAL_IDENTITY relation has all the information about the registered dogs with unique primary keys

The ANIMAL_IDENTITY table was downloaded from NYC open data

ANIMAL_IDENTITY (ANIMALID(PK), ANIMALNAME, ANIMALGENDER, ANIMALBREED, ANIMALWEIGHT, AIMALDOB)

ANIMALID	ANIMALNAME	ANIMALGENDER	ANIMALDOB	ANIMALBREED	AGE
562	LYDIA	F	01/12/2011	Beagle Crossbreed	7
563	EMMY	F	01/06/2014	Doberman Pinscher	4
564	BARTHOLOMEW	M	01/08/2007	Jack Russell Terrier	11
565	JJYODAA	M	01/03/2010	Shih Tzu	8
566	JENNY	F	01/10/2012	Silky Terrier	6
567	ME	F	01/01/2004	Miniature Pinscher	14
568	SAM	M	01/11/2007	Labrador Retriever	11
569	CASPER	M	01/11/2012	Beagle	6
570	CARLY	F	01/06/2014	Doberman Pinscher	4
571	GABE	M	01/10/2012	Rhodesian Ridgeback	6
572	LOLA	F	01/08/2008	Rottweiler	10
573	LEXY	F	01/12/2006	Yorkshire Terrier	12
574	NAKIA	F	01/07/2012	American Pit Bull Mix / Pit Bull Mix	6
575	BACON	M	01/02/2001	Shiba Inu	17
576	BRIX	F	01/08/2006	Puggle	12
577	FRITZ	M	01/08/2005	"Schnauzer, Miniature"	13

PRIMARY KEY:

ANIMALID

```
CREATE TABLE ANIMAL_IDENTITY
   (     ANIMALID NUMBER,
         ANIMALNAME VARCHAR2(30),
         ANIMALGENDER VARCHAR2(1),
         ANIMALDOB DATE,
         ANIMALBREED VARCHAR2(255),
          CONSTRAINT "ANIMAL_IDENTITY_PK" PRIMARY KEY ("ANIMALID") ENABLE
   ) ;
```

Degree = 6

Age: Derived attribute that can be computed from DOB and todays Date

DOMAINS:

AnimalID(PK): 0-9

AnimalName: a-z & A-Z

AnimalGender: 1 letter character (M or F)

AnimalBreed: a-z & A-Z

AnimalWeight: 0-9 Kgs

Animal DOB: MM/DD/YEAR

RELATION 4: ANIMAL_ REGISTRATION shows all the dogs that have been registered with respected Owners, that includes the owner's registration start Date, and Ending Date. (This also allows us to track previous owners)

ANIMAL_ REGISTRATION(REGISTRATIONID(PK), ANIMALID(FK), OWNERID(FK), ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

REGISTRATIONID	ANIMALID	OWNERID	ANIMALBREED	REGISSUEDATE	REGEXPIRYDATE	REGSTATUS
1000	562	1	Beagle Crossbreed	04/04/2015	04/04/2020	A
1001	563	1	Doberman Pinscher	04/04/2016	04/04/2019	A
1002	564	2	BARTHOLOMEW	05/05/2015	05/05/2019	A
1003	565	2	Shih Tzu	04/06/2016	04/06/2019	A
1004	567	3	Miniature Pincher	06/07/2016	06/07/2029	A
1005	569	3	Beagle	05/05/2017	05/05/2020	A
1006	570	4	Doberman Pinscher	06/06/2017	06/06/2020	A
1007	571	4	Rhodesian Ridgeback	05/05/2017	05/05/2018	A
1008	572	5	Rottweiler	06/06/2018	07/07/2022	A
1020	585	10	German Shepherd	06/07/2017	06/07/2021	A
1021	589	11	Chihuahua	07/08/2016	07/08/2020	A
1022	591	11	Yorkshire Terrier	07/08/2016	07/08/2020	A
1023	592	13	poodle	04/04/2013	04/04/2014	N
1024	592	14	poodle	04/05/2014	04/05/2019	A
1025	592	14	Dachshund	06/07/2017	06/07/2021	A
1026	94	15	pit bull	05/06/2016	05/06/2021	A
1027	596	15	poodle	04/04/2015	04/04/2016	N

PRIMARY KEY:

REGISTRATIONID

FORIGN KEYS:

FOREIGN KEY (OWNERID) REFERENCES DOG_OWNER(OWNERID);

FOREIGN KEY (ANIMALID) REFERENCES ANIMAL_IDENTITY(ANIMALID);

Degree:7

Tuples:

DOMAINS:

REGISTRATIONID(PK): 0-9

ANIMALID(FK): 0-9 & all the valid and registered AnimalD's from DOG_IDENTITY relation.

OWNERID(FK): 0-9 & all the valid OwnerID'S from DOG_OWNER relation

ANIMALBREED: a-z & A-Z

REGISSUEDATE: MM/DD/YEAR

REGEXPIRYDATE: MM/DD/YEAR

REGSTATUS- A-active if registration is not expired, or N-Not active if registration is expired.

SQL CREATE TABLE:

```
CREATE TABLE ANIMAL_REGISTRATION(

REGISTRATIONID NUMBER,

ANIMALID NUMBER,

OWNERID NUMBER,

ANIMALBREED VARCHAR(255),

REGISSUEDATE DATE,

REGEXPIRYDATE DATE,

REGSTATUS VARCHAR(1),


CONSTRAINT ANIMAL_REGISTRATION_PK PRIMARY KEY (REGISTRATIONID) ENABLE

);
```

SQL FORIGN KEYS:

```
ALTER TABLE ANIMAL_REGISTRATION

ADD FOREIGN KEY (OWNERID) REFERENCES DOG_OWNER(OWNERID);

ALTER TABLE ANIMAL_REGISTRATION

ADD FOREIGN KEY (ANIMALID) REFERENCES ANIMAL_IDENTITY(ANIMALID);
```

SQL INSERT TO TABLE:

```
INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1000, 562, 1, 'Beagle Crossbreed', '04/04/2015', '04/04/2020',
'A')


INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1001, 563, 1, 'Doberman Pinscher', '04/04/2016', '04/04/2019',
'A')



INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1002, 564, 2, 'BARTHOLOMEW', '05/05/2015', '05/05/2019', 'A')


INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1003, 565, 2, 'Shih Tzu', '04/06/2016', '04/06/2019', 'A')


INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1004, 567, 3, 'Miniature Pincher', '06/07/2016', '06/07/2029',
'A')


INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1005, 569, 3, 'Beagle', '05/05/2017', '05/05/2020', 'A')
```

```
INSERT INTO ANIMAL_REGISTRATION (REGISTRATIONID, ANIMALID, OWNERID,

ANIMALBREED, REGISSUEDATE, REGEXPIRYDATE, REGSTATUS)

VALUES (1006, 570, 4, 'Doberman Pinscher', '06/06/2017', '06/06/2020',
'A')
```

(ALL VALUES INSERTED WITH THE SAME APPROACH.)

RELATION 5: VIOLATION_RECEIPT shows the receipt of violation given to a dog owner at the time.

VIOLATION_RECEIPT(RECEIPTID(PK), OWNERID(FK), DATEVIOLATION)

RECEIPTID	OWNERID	DATEVIOLATION
2000	1	04/22/2018
2001	1	04/04/2017
2002	2	12/12/2017
2003	2	01/02/2018
2004	3	02/03/2018
2005	5	04/05/2016
2006	5	01/02/2018
2007	7	02/18/2018
2008	9	03/03/2017
2009	7	04/04/2018

PRIMARY KEY:

RECEIPTID

FORIGN KEY:

OWNERID

Degree:3

DOMAINS:

RECEIPTID(PK): 0-9

OWNERID(FK):0-9 & all the valid OwnerID'S from DOG_OWNER relation

DATEVIOLATION: DATE

SQL CREATE TABLE:

```
CREATE TABLE VIOLATION_RECEIPT(

RECEIPTID NUMBER,

OWNERID NUMBER,

DATEVIOLATION DATE,

CONSTRAINT VIOLATION_RECEIPT_PK PRIMARY KEY (RECEIPTID) ENABLE

);
```

SQL FORIGN KEY:

```
ALTER TABLE VIOLATION_RECEIPT

ADD FOREIGN KEY (OWNERID) REFERENCES DOG_OWNER(OWNERID);
```

SQL INSERT VALUES:

```
INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2000, 1, '04/22/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2001, 1, '04/04/2017')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2002, 2, '12/12/2017')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2003, 2, '01/02/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2004, 3, '02/03/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2005, 5, '04/05/2016')
```

```
INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2006, 5, '01/02/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2007, 7, '02/18/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2008, 9, '03/03/2017')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2009, 7, '04/04/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2010, 7, '03/04/2015')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2011, 12, '03/02/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2012, 9, '01/01/2016')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2013, 2, '02/03/2018')

INSERT INTO VIOLATION_RECEIPT(RECEIPTID, OWNERID, DATEVIOLATION)

VALUES (2014, 2, '03/03/2016')
```

RELATION 6: VIOLATION_DESCRIPTION describes the violation receipt and the dog information. The violation receipt includes many violations given to the owner.

VIOLATION_DESCRIPTION(DESCRIPTIONID(PK), RECEIPTID(FK), ANIMALID(FK), VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)

DESCRIPTIONID	RECEIPTID	ANIMALID	VIOLATIONTYPE	VIOLATIONFINE	VIOLATIONSTATUS
5000	2000	562	no leash	240	paid
5001	2000	563	no leash	240	paid
5002	2001	563	loud barking	24	unpaid
5003	2002	564	loud barking	24	paid
5005	2003	565	bit a person	24	unpaid
5006	2004	567	not vaccinated	240	unpaid
5007	2004	569	not vaccinated	240	paid
5008	2005	572	bit a person	240	paid
5009	2008	582	loud barking	24	paid
5010	2009	577	not vaccinated	240	unpaid
5012	2006	572	road cross on red signal	120	paid
5013	2007	578	road cross on red signal	120	unpaid
5014	2007	578	no leash	240	unpaid

PRIMARY KEY:

DESCRIPTIONID

FORIGN KEY:

RECEIPTID

ANIMALID

Degree:3

DOMAINS:

DESCRIPTIONID(PK): 0-9

RECEIPTID(FK): 0-9 & all the valid receipt ID'S in VIOLATION_RECEIPT RELATION

ANIMALID(FK): 0-9 & all the valid and registered AnimalD's from DOG_IDENTITY relation.

VIOLATIONTYPE: a-z & A-Z

VIOLATIONFINE: 0-9

VIOLATIONSTATUS: PAID OR UNNPAID.

SQL CREATE TABLE:

```
CREATE TABLE VIOLATION_DESCRIPTION(

DESCRIPTIONID NUMBER,

RECEIPTID NUMBER,

ANIMALID NUMBER,

VIOLATIONTYPE VARCHAR(50),

VIOLATIONFINE NUMBER,

VIOLATIONSTATUS VARCHAR(1),

CONSTRAINT VIOLATION_DESCRIPTION_PK PRIMARY KEY (DESCRIPTIONID)

);
```

SQL FORIGN KEY:

```
ALTER TABLE VIOLATION_DESCRIPTION

ADD FOREIGN KEY (RECEIPTID) REFERENCES VIOLATION_RECEIPT(RECEIPTID);



ALTER TABLE VIOLATION_DESCRIPTION

ADD FOREIGN KEY (ANIMALID) REFERENCES ANIMAL_IDENTITY(ANIMALID);
```

SQL INSERT VALUES:

```
INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,

VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)

VALUES (5000, 2000, 562, 'no leash', 200, 'paid')
```

```
INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5001, 2000, 563, 'no leash', 200, 'paid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5002, 2001, 563, 'loud barking', 20, 'unpaid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5003, 2002, 564, 'loud barking', 20, 'upaid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5005, 2003, 565, 'bit a person', 20, 'unpaid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5006, 2004, 567, 'not vaccinated', 200, 'unpaid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5007, 2004, 569, 'not vaccinated', 200, 'paid')
```

```
INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5008, 2005, 572, 'bit a person', 200, 'paid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5009, 2008, 582, 'loud barking', 20, 'paid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5010, 2009, 577, 'not vaccinated', 20, 'paid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5012, 2006, 572, 'road cross on red signal', 100, 'paid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
VALUES (5013, 2007, 572, 'road cross on red signal', 100, 'unpaid')

INSERT INTO VIOLATION_DESCRIPTION (DESCRIPTIONID, RECEIPTID, ANIMALID,
VIOLATIONTYPE, VIOLATIONFINE, VIOLATIONSTATUS)
```

```
VALUES (5014, 2007, 572, 'no leash', 200, 'unpaid')
```

RELATION 7: ANIMAL_PHOTO shows all the name of the dogs, their respected owners, and images of the dog

ANIMAL_PHOTO(ANIMALID(FK), OWNERID(FK), ANIMALPHOTO)

ANIMALID	OWNERID	ANIMALPHOTO
562	19	C:\data\IMG1.JPG
606	19	C:\data\IMG2.JPG
755	18	C:\data\IMG8
778	17	C:\data\ING66.JPG
784	17	C:\data\IMG77.JPG
834	16	C:\data\IMG67.JPG

Degree=3

Composite Key: AnimalID(FK), OwnerID(FK), AnimalPhoto can form a unique key for their occurrence as a **primary Key**.

Domains:

AnimalID(FK): a-z & A-Z & all the valid and registered AnimalD's from DOG_IDENTITY relation.

OwnerID(FK): 0-9 & all the valid OwnerID'S from DOG_OWNER relation

AnimalPhoto: .jpeg format or any Image Format (Multi Value attribute as users can save many images of the dog)

SQL CREATE TABLE:

```
CREATE TABLE ANIMAL_PHOTO(

ANIMALID NUMBER,

OWNERID NUMBER,

ANIMALPHOTO VARCHAR(4000),

CONSTRAINT ANIMAL_PHOTO_PK PRIMARY KEY (ANIMALID, OWNERID, ANIMALPHOTO)

);
```

SQL FORIGN TABLE:

```
ALTER TABLE ANIMAL_PHOTO

ADD FOREIGN KEY (ANIMALID) REFERENCES ANIMAL_IDENTITY(ANIMALID);

ALTER TABLE ANIMAL_PHOTO

ADD FOREIGN KEY (OWNERID) REFERENCES DOG_OWNER(OWNERID);
```

SQL INSERT VALUES:

```
INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (562, 19, 'C:\data\IMG1.JPG')

INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (606, 19, 'C:\data\IMG2.JPG')

INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (755, 18, 'C:\data\IMG8.JPG')

INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (778, 17, 'C:\data\IMG66.JPG')

INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (784, 17, 'C:\data\IMG77.JPG')

INSERT INTO ANIMAL_PHOTO (ANIMALID, OWNERID, ANIMALPHOTO)

VALUES (834, 16, 'C:\data\IMG67.JPG')
```

QUESTIONS:

1) Identify dogs without violations in the last year. Display the owner name, dog name, breed and email. Use a nested select to answer this question.

   We will replace the last year to, from the year 4/20/2017

```
SELECT DOG_OWNER.OWNERFIRSTNAME, DOG_OWNER.OWNERLASTNAME,
DOG_OWNER.OWNEREMAIL, ANIMAL_REGISTRATION.ANIMALBREED,
ANIMAL_IDENTITY.ANIMALNAME
FROM  ANIMAL_REGISTRATION, DOG_OWNER, ANIMAL_IDENTITY
WHERE ANIMAL_REGISTRATION.OWNERID=DOG_OWNER.OWNERID AND
ANIMAL_REGISTRATION.REGSTATUS='A' AND
ANIMAL_IDENTITY.ANIMALID=ANIMAL_REGISTRATION.ANIMALID AND
ANIMAL_REGISTRATION.OWNERID NOT IN
(SELECT OWNERID
FROM VIOLATION_RECEIPT
WHERE to_date(DATEVIOLATION, 'mm/dd/yyyy')>='4/20/2017');
```

OWNERFIRSTNAME	OWNERLASTNAME	OWNEREMAIL	ANIMALBREED	ANIMALNAME
adil	abuwani	adil@gmail.com	Doberman Pinscher	CARLY
adil	abuwani	adil@gmail.com	Rhodesian Ridgeback	GABE
miky	rod	mikye@gmail.com	Yorkshire Terrier	LEXY
miky	rod	mikye@gmail.com	Shiba Inu	BACON
nini	lii	nini@gmail.com	Yorkshire Terrier	MILEY
nini	lii	nini@gmail.com	Yorkshire Terrier	BLU
nini	lii	nini@gmail.com	Border Terrier	LUKE
boo	lii	blee@gmail.com	Poodle	PUFFY
boo	lii	blee@gmail.com	Yorkshire Terrier	ELVIS
jamil	taha	jamil@gmail.com	poodle	RAWLY
jamil	taha	jamil@gmail.com	German Shepherd	HARLEY
hussain	shah	hussain@gmail.com	Chihuahua	SWEEPEA
hussain	shah	hussain@gmail.com	Yorkshire Terrier	CUPCAKES
sikku	surani	sikku@gmail.com	poodle	ROCKY
sikku	surani	sikku@gmail.com	Dachshund	ROCKY
malika	modi	malika@gmail.com	PIT BULL	CHINO
ali	smith	asmith@gmail.com	pit bull	CHIP
ali	hamza	hamza@gmail.com	pit bull	MOMO
ayesha	ramani	ayesha@gmail.com	pit bull	BELLA

2) Identify zip codes without registered pit bulls now. Display the zip code. Remove duplicate zip codes. Use a nested select to answer this question.

   We will modify the now to date 4/20/2018

```
SELECT distinct DOG_OWNER.zip
FROM DOG_OWNER
WHERE DOG_OWNER.OWNERID NOT IN
```

```
(SELECT ANIMAL_REGISTRATION.OWNERID
FROM ANIMAL_REGISTRATION
WHERE UPPER(ANIMALBREED)='PIT BULL' AND to_date(REGISSUEDATE,
'mm/dd/yyyy')>='4/20/2018')
```

ZIP
11367
11379
10002
11101
11103
11374
11385
11375

3) Identify dogs without owners now. Display the dog name, gender, breed and age. Use a nested select to answer this question.

We will modify the now to date 4/20/2018

```
SELECT ANIMALNAME, ANIMALGENDER, ANIMALBREED, AGE

FROM ANIMAL_IDENTITY

WHERE ANIMALID NOT IN

(SELECT ANIMAL_REGISTRATION.ANIMALID

FROM ANIMAL_REGISTRATION

WHERE to_date(REGISSUEDATE, 'mm/dd/yyyy')>='4/20/2018')
```

ANIMALNAME	ANIMALGENDER	ANIMALBREED	AGE
LYDIA	F	Beagle Crossbreed	7
EMMY	F	Doberman Pinscher	4
BARTHOLOMEW	M	Jack Russell Terrier	11
JJYODAA	M	Shih Tzu	8
JENNY	F	Silky Terrier	6
ME	F	Miniature Pinscher	14
SAM	M	Labrador Retriever	11
CASPER	M	Beagle	6
CARLY	F	Doberman Pinscher	4
GABE	M	Rhodesian Ridgeback	6
NAKIA	F	American Pit Bull Mix / Pit Bull Mix	6
BACON	M	Shiba Inu	17
BRIX	F	Puggle	12
FRITZ	M	"Schnauzer, Miniature"	13
SOFIE	F	Bichon Frise	5
MILEY	M	Yorkshire Terrier	11
BLU	F	Yorkshire Terrier	9
LUKE	M	Border Terrier	6
PUFFY	F	"Poodle, Miniature"	16
ELVIS	M	Yorkshire Terrier	9
More than 20 rows available. Increase rows selector to view more rows.			

4) Identify current dog owners with a residence near Queens College. Display the owner name, dog name and breed.

We will modify the now to date 4/20/2018

```
SELECT DOG_OWNER.OWNERFIRSTNAME, DOG_OWNER.OWNERLASTNAME,
ANIMAL_IDENTITY.ANIMALNAME, ANIMAL_IDENTITY.ANIMALBREED
FROM DOG_OWNER, ANIMAL_REGISTRATION, ANIMAL_IDENTITY
WHERE DOG_OWNER.OWNERID=ANIMAL_REGISTRATION.OWNERID AND
ANIMAL_REGISTRATION.ANIMALID=ANIMAL_IDENTITY.ANIMALID AND
UPPER(DOG_OWNER.STREET) LIKE '%KISSENA BLVD%' AND
UPPER(DOG_OWNER.CITY)='%FLUSHING%' AND DOG_OWNER.ZIP='11367' AND
to_date(ANIMAL_REGISTRATION.REGISSUEDATE,
'mm/dd/yyyy')>='4/20/2018';
```

OWNERFIRSTNAME	OWNERLASTNAME	ANIMALNAME	ANIMALBREED
smith	bob	LYDIA	Beagle Crossbreed
smith	bob	EMMY	Doberman Pinscher
smith	jon	BARTHOLOMEW	Jack Russell Terrier
smith	jon	JJYODAA	Shih Tzu
tom	jon	ME	Miniature Pinscher
tom	jon	CASPER	Beagle
adil	abuwani	CARLY	Doberman Pinscher
adil	abuwani	GABE	Rhodesian Ridgeback
ali	khan	LOLA	Rottweiler
miky	rod	LEXY	Yorkshire Terrier
More than 10 rows available. Increase rows selector to view more rows.			

5) Identify pictures of **female poodles less than five years old**. Display the dog name, age and all photos.
We will modify female poodles less then five years old to female beagle less than ten years old

```
SELECT ANIMAL_IDENTITY.ANIMALNAME, ANIMAL_IDENTITY.AGE,
ANIMALPHOTO
FROM ANIMAL_IDENTITY, ANIMAL_PHOTO
WHERE UPPER(ANIMAL_IDENTITY.ANIMALBREED) LIKE '%BEAGLE%'
AND ANIMAL_PHOTO.ANIMALID=ANIMAL_IDENTITY.ANIMALID
AND AGE <=9
```

ANIMALNAME	AGE	ANIMALPHOTO
LYDIA	7	C:\data\IMG1.JPG
REESE	9	C:\data\IMG2.JPG
VALLEY	7	C:\data\IMG8
LEYLA	7	C:\data\ING66.JPG

6) Identify dogs owned by **Sally Smith** with violations in the **last year**. Display the owner name, dog name, violation, date of violation and fine.
We will modify sally smith with smith bob, and the last year to 4/20/2017

```
SELECT DOG_OWNER.OWNERFIRSTNAME, DOG_OWNER.OWNERLASTNAME,
ANIMAL_IDENTITY.ANIMALNAME, VIOLATION_DESCRIPTION.VIOLATIONTYPE,
VIOLATION_DESCRIPTION.VIOLATIONFINE,
VIOLATION_RECEIPT.DATEVIOLATION
FROM DOG_OWNER, ANIMAL_IDENTITY, VIOLATION_DESCRIPTION,
VIOLATION_RECEIPT, ANIMAL_REGISTRATION
WHERE UPPER(DOG_OWNER.OWNERFIRSTNAME)='SMITH' AND
UPPER(DOG_OWNER.OWNERLASTNAME)='BOB'
AND DOG_OWNER.OWNERID=ANIMAL_REGISTRATION.OWNERID
AND ANIMAL_REGISTRATION.ANIMALID=ANIMAL_IDENTITY.ANIMALID
AND VIOLATION_RECEIPT.OWNERID=DOG_OWNER.OWNERID
AND  TO_DATE(VIOLATION_RECEIPT.DATEVIOLATION,
'MM/DD/YYYY')>='4/20/2017'
AND VIOLATION_RECEIPT.RECEIPTID=VIOLATION_DESCRIPTION.RECEIPTID
AND VIOLATION_DESCRIPTION.ANIMALID=ANIMAL_REGISTRATION.ANIMALID
```

OWNERFIRSTNAME	OWNERLASTNAME	ANIMALNAME	VIOLATIONTYPE	VIOLATIONFINE	DATEVIOLATION
smith	bob	LYDIA	no leash	200	04/22/2018
smith	bob	EMMY	no leash	200	04/22/2018

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.

```
SELECT ANIMALGENDER, COUNT(1) "NUMBER OF DOGS"
FROM ANIMAL_IDENTITY
GROUP BY ANIMALGENDER
```

ANIMALGENDER	NUMBER OF DOGS
M	604
F	495

8) Identify zip codes with the most pit bulls. Display two columns and one row for each zip code. The two output columns are zip code and number of dogs in that zip code. Display the zip code with the most pit bulls first. Use a function to answer this question.
We will modify pit bulls to poodle

```
SELECT DOG_OWNER.ZIP, COUNT(1)
FROM ANIMAL_REGISTRATION, DOG_OWNER
WHERE DOG_OWNER.OWNERID=ANIMAL_REGISTRATION.OWNERID
```

```
AND UPPER(ANIMAL_REGISTRATION.ANIMALBREED) LIKE '%POODLE%'
GROUP BY DOG_OWNER.ZIP
ORDER BY ZIP
```

ZIP	COUNT(1)
11101	3
11103	1
11105	1

9) Identify the number of total fines by owner in the last two years. 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. We will modify the last two years to 4/20/2016

```
SELECT DOG_OWNER.OWNERFIRSTNAME, COUNT(1) "TOTAL NUMBER
VIOLATION",
TO_CHAR(SUM(VIOLATION_DESCRIPTION.VIOLATIONFINE),'L99G999D99MI')
"TOTAL DOLLAR OF VIOLATION"
FROM VIOLATION_RECEIPT, DOG_OWNER, VIOLATION_DESCRIPTION
WHERE to_date(VIOLATION_RECEIPT.DATEVIOLATION,
'mm/dd/yyyy')>='4/20/2016' AND
DOG_OWNER.OWNERID=VIOLATION_RECEIPT.OWNERID AND
VIOLATION_RECEIPT.RECEIPTID=VIOLATION_DESCRIPTION.RECEIPTID
GROUP BY DOG_OWNER.OWNERFIRSTNAME,
VIOLATION_DESCRIPTION.VIOLATIONFINE
ORDER BY SUM(VIOLATION_DESCRIPTION.VIOLATIONFINE)
```

OWNERFIRSTNAME	TOTAL NUMBER VIOLATION	TOTAL DOLLAR OF VIOLATION
boo	1	$24.00
smith	3	$72.00
ali	1	$120.00
lulu	1	$120.00
tom	2	$480.00
smith	2	$480.00
lulu	2	$480.00

10) Increase all fines by 20%. Identify the SQL to perform this operation and the fine before and after this operation.

We will modify the 20% to 12%

```
UPDATE VIOLATION_DESCRIPTION
SET VIOLATIONFINE=VIOLATIONFINE*1.2;

SELECT TO_CHAR(VIOLATIONFINE, 'L99G999D99MI') "VIOLATION FINE"
FROM VIOLATION_DESCRIPTION
```

VIOLATION FINE
$240.00
$240.00
$24.00
$24.00
$24.00
$240.00
$240.00
$240.00
$24.00
$240.00
$120.00
$120.00
$240.00

11) Display the structure of all tables using the SQL Describe operation.

```
DESC DOG_OWNER
```

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOG_OWNER	OWNERID	NUMBER	22	-	-	1	-	-	-
	OWNERFIRSTNAME	VARCHAR2	30	-	-	-	-	-	-
	OWNERLASTNAME	VARCHAR2	30	-	-	-	-	-	-
	OWNEREMAIL	VARCHAR2	320	-	-	-	-	-	-
	STREET	VARCHAR2	30	-	-	-	-	-	-
	CITY	VARCHAR2	30	-	-	-	-	-	-
	STATE	VARCHAR2	30	-	-	-	-	-	-
	NEIGHBORHOOD	VARCHAR2	30	-	-	-	-	-	-
	ZIP	VARCHAR2	5	-	-	-	-	-	-

1 - 9

DESC DOG_OWNER_PHONE

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
DOG_OWNER_PHONE	OWNERID	NUMBER	22	-	-	1	-	-	-
	OWNERPHONE	VARCHAR2	15	-	-	2	-	-	-

1 - 2

DESC ANIMAL_IDENTITY

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ANIMAL_IDENTITY	ANIMALID	NUMBER	22	-	-	1	-	-	-
	ANIMALNAME	VARCHAR2	30	-	-	-	✓	-	-
	ANIMALGENDER	VARCHAR2	1	-	-	-	✓	-	-
	ANIMALDOB	DATE	7	-	-	-	✓	-	-
	ANIMALBREED	VARCHAR2	255	-	-	-	✓	-	-
	AGE	NUMBER	22	-	-	-	✓	-	-

1 - 6

DESC ANIMAL_REGISTRATION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ANIMAL_REGISTRATION	REGISTRATIONID	NUMBER	22	-	-	1	-	-	-
	ANIMALID	NUMBER	22	-	-	-	-	-	-
	OWNERID	NUMBER	22	-	-	-	-	-	-
	ANIMALBREED	VARCHAR2	255	-	-	-	-	-	-
	REGISSUEDATE	DATE	7	-	-	-	-	-	-
	REGEXPIRYDATE	DATE	7	-	-	-	-	-	-
	REGSTATUS	VARCHAR2	1	-	-	-	-	-	-
									1 - 7

DESC VIOLATION_RECEIPT

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VIOLATION_RECEIPT	RECEIPTID	NUMBER	22	-	-	1	-	-	-
	OWNERID	NUMBER	22	-	-	-	-	-	-
	DATEVIOLATION	DATE	7	-	-	-	✓	-	-
									1 - 3

DESC VIOLATION_DESCRIPTION

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
VIOLATION_DESCRIPTION	DESCRIPTIONID	NUMBER	22	-	-	1	-	-	-
	RECEIPTID	NUMBER	22	-	-	-	-	-	-
	ANIMALID	NUMBER	22	-	-	-	-	-	-
	VIOLATIONTYPE	VARCHAR2	50	-	-	-	-	-	-
	VIOLATIONFINE	NUMBER	22	-	-	-	-	-	-
	VIOLATIONSTATUS	VARCHAR2	15	-	-	-	-	-	-
									1 - 6

DESC ANIMAL_PHOTO

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
ANIMAL_PHOTO	ANIMALID	NUMBER	22	-	-	1	-	-	-
	OWNERID	NUMBER	22	-	-	2	-	-	-
	ANIMALPHOTO	VARCHAR2	4000	-	-	3	-	-	-
									1 - 3

12) Display the Oracle version by entering

```
   select *
from product_component_version;
```

PRODUCT	VERSION	STATUS
NLSRTL	11.2.0.2.0	Production
Oracle Database 11g Express Edition	11.2.0.2.0	64bit Production
PL/SQL	11.2.0.2.0	Production
TNS for 64-bit Windows:	11.2.0.2.0	Production