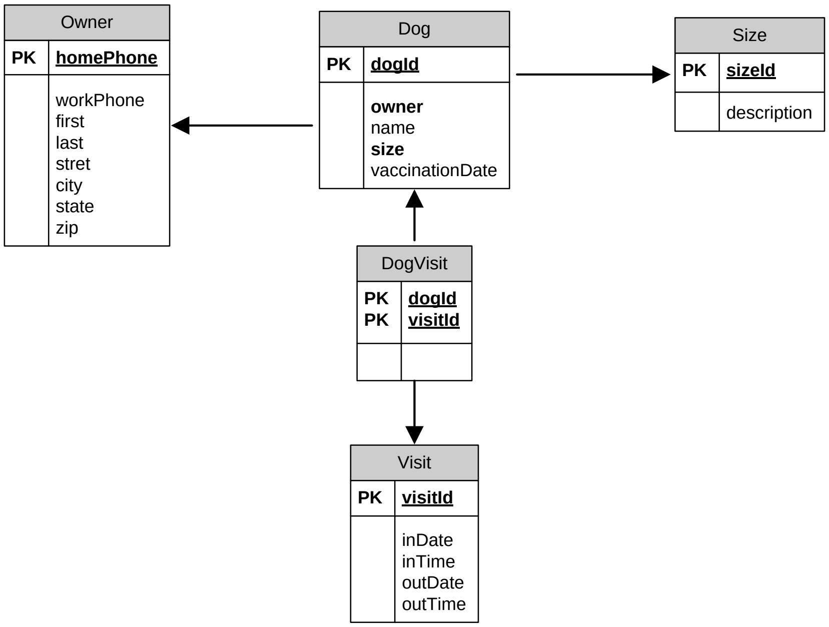
Database Design, CSCI 340, Spring 2016
Relational Algebra exercises, March 7

In the following database schema are relations are 1 to m. The side of the arrow with the arrow head represents minimal and maximum cardinalities of 1 (i.e. 1..1) and the other side represents a minimum cardinality of 0 and a maximum cardinalities of m (0..m).

Yappers Dog
Kennel



Write relational algebra queries and SQL queries to extract the following information from the Yappers db.

1. Give the first and last names of all owners in the database.

Relational Algebra	SQL

$\pi_{first, last}$ Owner	SELECT first, last
	FROM Owner

2. Give the name of all small dogs in the database. For this query say that small dogs have size=1. Don't worry about duplicate names.

Relational Algebra	SQL

$\pi_{name}(\sigma_{size=1} Dog)$	SELECT name
	FROM Dog
	WHERE size=1

3. Give the name of all small dogs in the database, but don't assume that size=1 for small dogs. Instead use the description field of the Size table.

The Size table is as follows:

sizeId	description
1	small
2	medium
3	large

Relational Algebra	SQL

$\pi_{name}(\sigma_{description='small'}(Dog \bowtie_{Dog.size=Size.sizeId} Size))$ or
$\pi_{name}(Dog \bowtie_{Dog.size=Size.sizeId} \pi_{sizeId}(\sigma_{description='small'} Size))$
	SELECT name
	FROM Dog JOIN Size
	ON
	Dog.size=Size.sizeId
	WHERE
	description='small'

4. Give the name of each dog along with its size (small, medium, large).

Relational Algebra	SQL

$\pi_{name, description}(Dog \bowtie_{Dog.size = Size.sizeId} Size)$	SELECT name, description
	FROM Dog JOIN Size
	ON Dog.size=Size.sizeId

5. Give the first and last names of all owners in the database who have kenneled a dog.

Relational Algebra	SQL
$\pi_{first,\ last}\left(Owner \bowtie_{Owner.homePhone = Dog.owner} (Dog \bowtie DogVisit)\right)$	SELECT first, last FROM Owner JOIN Dog ON Owner.homePhone=Dog.owner JOIN DogVisit ON Dog.dogId=DogVisit.dogId

6. Give the first and last names of all owners in the database who have kenneled a small dog.

Relational Algebra	SQL
$\pi_{first,\ last}\left(Owner \bowtie_{Owner.homePhone = Dog.owner}\right.$ $\sigma_{description = 'small"}\left(\left(\left(Dog \bowtie_{Dog.size = Size.sizeId} Size\right) \bowtie DogVisit\right)\right.$	SELECT first, last FROM Owner JOIN Dog ON Owner.homePhone=Dog.owner JOIN DogVisit ON Dog.dogId=DogVisit.dogId JOIN Size ON Dog.size=Size.sizeId WHERE description='small'

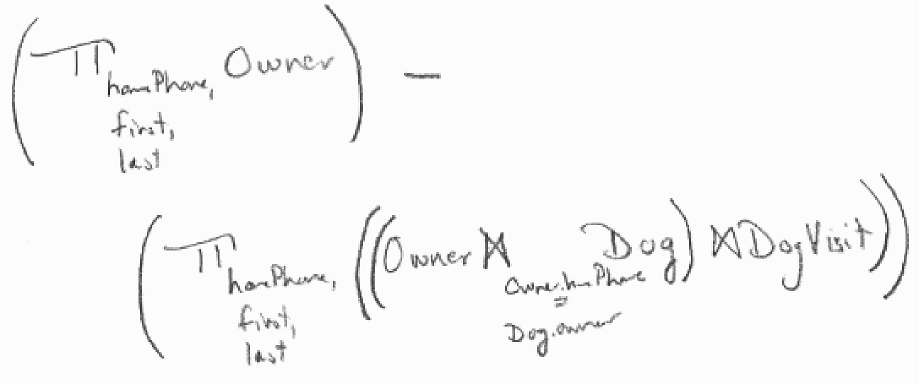
7. Give each owner name, first and last name, along with the number of dogs the db knows about. Owners with no dogs should still be listed.

Relational Algebra	SQL

${}_{first,\ last}\mathcal{J}_{COUNT\ (dogId)}\left(Owner \bowtie_{Owner.homePhone = Dog.owner} Dog\right)$	SELECT first, last,
	COUNT(dogId)
	FROM Owner LEFT JOIN Dog
	ON Owner.homePhone=Dog.owner

8. List the home phone number, first and last names, of all owners in the db that have never kenneled a dog.

Relational Algebra	SQL

 $\left(\pi_{\text{homePhone, first, last}}\ \text{Owner}\right) - \left(\pi_{\text{homePhone, first, last}}\left(\left(\text{Owner} \bowtie_{\text{Owner.homePhone} = \text{Dog.owner}} \text{Dog}\right) \bowtie \text{DogVisit}\right)\right)$	SELECT homePhone, first, last
	FROM Owner
	WHERE homePhone NOT IN
	(SELECT homePhone
	FROM Owner JOIN Dog
	ON
	Owner.homePhone=Dog.owner
	JOIN
	DogVisit
	ON
	Dog.dogId=DogVisit.dogId)