This test covers: Chapters 1, 2 and 6.
Directions: Complete all questions. Partial credit will be given. You may use one sheet of paper with handwritten notes and your take home portion to this exam. Keep in mind there are many ways to answer most questions correctly.

Note: the first thee pages are the same as your take home exam. Please do not re-answer these questions.

The situation: You have been contracted by your county to develop a database containing information about all the dogs within your county. Vital information includes the dog's name, breed, color, age, whether it has been sterilized, along with its owner and veterinarian's information. Create three relations:

- **Dog:** which includes the dog's name [dName], breed [breed], age [age], gender [gender], spayed/neutered state [intact], owner's name (first/last) [oName], veterinarian's name (first/last) [vName]
- **Owner:** with includes the owner's name (first/last) [name], personal address [address], and telephone number [phone].
- **Vet:** which includes the vet's name (first/last) [name], company name [company], company address [address], open weekends (yes or no) [weekend], and company telephone number [phone].

Please use the given titles in brackets for attributes to prevent confusion during the in class portion of the exam. For simplicity, there is no need to break apart "names", "address" or "telephone number" into separate attributes. For this situation, you may assume the following during all queries:

- All *people's* (owners and vets) names are unique. (ie. there is only one John Smith in the county, but the may be a Mary Smith and a John Diaz.)
- For multiple dogs with the same owner, the dog names are unique. (ie. John Smith can only have one dog named Spike, but John Smith can also have another dog Lassie. Further someone else other than John Smith may have a dog named Spike.)
- All dogs/people/vets live and work in the county and thus information should be in the database.
- NULLs are NOT allowed in any attribute.
- All veterinarian office names are unique.
- All telephone numbers are unique to the given person/company. (ie. Every person has a personal cell phone, and each office has ONE main phone line.)

You may **not** assume:

- Dog names are unique in general.
- There is only one vet per company; likewise there is only one company per vet.

1.) Using SQL, create a relational schema for the proposed database. Be sure to give each attribute a type and label keys. If you feel something may be controversial, explain why you chose what you chose.

```
CREATE TABLE Dog (
        dName      VARCHAR(255),
        breed      VARCHAR(255),
        age        INT,
        gender     CHAR(1), -- M/F only.
        intact     BOOL,
        oName      VARCHAR(255),
        vName      VARCHAR(255),
        PRIMARY KEY (dName, oName) -- Dog and owner name pairs are unique.
);

CREATE TABLE Owner (
        name        VARCHAR(255),
        address     VARCHAR(255), -- treat whole address as a string
        phone       CHAR(10),
        PRIMARY KEY (name)  -- Assumption that all people are unique.
);

CREATE TABLE Vet (
        name        VARCHAR(255),
        company     VARCHAR(255),
        address     VARCHAR(255),
        weekend     BOOL,
        phone       CHAR(10),
        PRIMARY KEY (name, company) -- People are unique, but could work at two+ places.
);
```

2.) No relational schema is perfect. Assuming this was the relational model settled upon, describe any short comings that may be experienced in practice. You may comment on both the given assumptions and any other issues you feel are relevant. There is no one right answer here, so be sure to provide a quality explanation.

- Name attributes (dName, oName, name and name) are inconsistent.
- Does not allow two people to own a single dog.
- Can/should rescue organizations/companies be a dog's owner?
- Other things such as dogs origins (parents), insurance coverage, vaccine types/dates, etc. are not allowable.
- Services from vets are not given.
- Names and addresses as one field each prevent sorting and searching in a traditional manner.
- Hours of operation for the vet offices.

3.) Using *relational algebra*, create a query that provides the names (only!) of all dogs owned by 'John Smith'. You may use any format of your choice (but not SQL).

$$\pi_{\text{dName}}\left(\sigma_{\text{oName = 'John Smith'}}\left(Dog\right)\right)$$

4.) Using *SQL*, create a query that provides the names and telephone numbers of all veterinarian offices open on weekends. Be sure to list each office once.

```
SELECT DISTINCT company, phone
FROM Vet
WHERE weekend = TRUE;
```

5.) A mass paper mailing is to be sent to all dog *owners*, using SQL provide a query that will result in name and address pairs that then can be used to address envelopes. Sort by address (Usually you would sort by zip code, but that does not work here since we have the entire street/city/state/zip in one attribute). To ensure that postage is not wasted, does each person receives only one mailing and why?

```
SELECT name, address
FROM owner
ORDER BY address;
```

Each person will receive one mailing because the table Owner has one entry per owner. No distinct is necessary and may cause unnecessary performance issues. Adding it would not be incorrect.

6.) A mass paper mailing is to be sent to all *veterinarians*, using SQL provide a query that will result in company name and address pairs that then can be used to address envelopes. To ensure that postage is not wasted, does each *company* receives only one mailing and why?

```
SELECT DISTINCT company, address
FROM vet;
```

Each company can receive more than one mailing because a company may employ more than one veterinarian. If the keyword 'DISTINCT' is used, however, each company should receive only one mailing.

7.) Using relational algebra, find all veterinarians that are also dog owners.

$$\pi_{\text{oName}}\left(\sigma_{\text{oName=name}}\left(Dog \bowtie Vet\right)\right)$$

8.) A new law has been passed that limits dog owners to having a maximum of 10 dogs. The dog control officer needs a list of the names and addresses of all owners currently violating this law. Provide a SQL query that will return such a list. Does it prevent duplicates?

```
SELECT oName, address, count(*)
FROM Dog, Owner
WHERE oName = name
GROUP BY oName, address
HAVING count(oName) > 10;
```

Duplicates are automatically compressed into a single tuple by the GROUP BY clause.

9.) Find the names of all owners who have intact female Labradors using a SQL query. Be sure to include each name once, and allow for the various types of Labradors, such as Black Labradors, Chocolate Labradors and Yellow Labradors.

```
SELECT DISTINCT oName
FROM Dog
WHERE gender = 'F' AND ( breed LIKE '%Labrador' );
```

10.) Describe what the following query returns in the larger context of the problem:
SELECT oName, breed, count(*)
FROM Dog
GROUP BY oName, breed

The query returns every owner's name, each breed of dog he/she has, and the count of how many dogs of that breed the owner has.

11.) You are creating an web-based application that can be accessed by the local town, village and city clerks. You want to create transactions for a few popular tasks. For each of the following scenarios, describe whether the transaction should be READ ONLY or READ WRITE, and which isolation level it should run at (READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE). Be sure to state why.

I Providing a list of all dogs owned by John Smith.

II Adding the dog 'Spike' (and relevant information) with existing owner 'John Smith'.

III Leila Goldberg is moving out of the county and all her information should be removed from the owner relation and dog relation.

IV A transaction that first provides all dogs that are patients at a given veterinarian, then in another query (but still in the same transaction) provides the names and addresses of owners that have dogs at the given veterinarian.

I. The transaction should be READ ONLY because the query is only reporting a list and not changing or adding any data. The isolation level can be as low as READ COMITTED since any changes will the query is made is unlikely and not catastrophic.

II. The transactions should be READ WRITE because the data is being added to the database. The isolation level is rather irrelevant here since no other data is being read. The lower choices would be better.

III. The transaction should be READ WRITE because data is being removed from the database. The isolation level should be SERIALIZABLE or REPEATABLE READ because the removal involves multiple databases and it will ensure a more accurate representation of data in later queries if all of the deleting actions were performed before any other reads or writes.

IV. The transaction is READ ONLY because the queries are only returning information to the customer and not altering or adding any data. The isolation level should be SERIALIZABLE to prevent phantom reads or other dirty data from appearing since it is possible other customers will be using the database and altering or adding data.

12.) The vet Jane Wolcott is retiring and all her patients (dogs) will now be assigned to Jeff Foxworthy, who is already in the database. Write a SQL query that will perform such an action.

```
UPDATE Dog
SET vName = 'Jeff Foxworthy'
WHERE vName = 'Jane Wolcott';

REMOVE FROM vet
WHERE name = 'Jane Wolcott';
```

13.) Describe what the following query returns in the larger context of the problem. If the goal was to find a male dog to breed with a female Siberian Husky, will the return be useful? Are there any issues and could it be improved? How?

SELECT oName, phone
FROM Owner, Dog
WHERE oName = name AND gender = 'M' AND intact = true;

This query is missing some key parts. Issues include not having the ages of the returned dogs. It will return puppies and dogs in their golden years, which are not suitable to breeding. Other minor issues could be returning more information such at "age, dName, address". One issue not addressed at all is the desired breed(s). Perhaps it should be limited to other Huskies.

14.) For each of the following, describe if allowing NULLs to be used in the age attribute of Dog would affect the query. Rewrite it if necessary to be NULL-friendly.

- Finding all puppies. That is:

  SELECT *
  FROM Dog
  WHERE age < 1;

- The above will work as intended. The only challenge NULL introduces is that puppies that do not have ages stored (ie. age = NULL) will not be returned. That said, there is no way to actually know the true age and they should not be returned.

- Finding all breeding aged dogs. That is:

```
SELECT *
FROM Dog
WHERE (NOT age < 2) AND (NOT age > 6);
```

- This query is actually okay too. (NOT age < 2) returns only those dogs greater than or equal to 2. (NOT age > 6); returns dogs less than or equal to 6. Combining these will provide dogs between 2 and 6. Dogs with age is NULL will not be returned. This is to be expected since there age is not determined.

15.) One issue that many programmers utilizing databases face is unintended results. What may be unusual given the following query (given in relational algebra)? Hint: Reread assumptions.

$$\pi_{\text{name,company,phone}}\left(\sigma_{\text{name}='\text{John Smith}'}\left(\text{Vet}\right)\right)$$

As discussed, this should return the "John Smith, His Company and Phone". The issue with queries like this are that while programming you must account for any special cases. If John only works at one place, the result will be as directed. If John works at two places, the return will contain two tuples. If John is retired or does not work as a Vet, the returning relation will be empty.

16.) In the take home portion of the test you were asked to create a key for each relation (question 1 on take home exam). There may be additional possible keys for each relation. For each candidate relation-key pairs below, state if it would be a valid key and why. Remember to use the assumptions and a conservative programming mindset for items not discussed.

- Relation: Dog, Primary Key: oName – Invalid. Some owners will have multiple dogs.
- Relation: Dog, Primary Key: (breed, age, intact, vName) – Invalid. One litter will have many dogs that likely will violate this this.
- Relation: Owner, Primary Key: address – Invalid. Two people could live in the same household and have dogs.
- Relation: Owner, Primary Key: phone – Valid. Based on assumptions, each person has a unique phone number. (Probably not true in reality)
- Relation: Vet, Primary Key: phone – Invalid. Two veterinarians who work at the same company will have the same phone (see assumptions)
- Relation: Vet, Primary Key: (company, address, phone) – Invalid. Same as the prior.