Homework 7 Part 1 Solutions

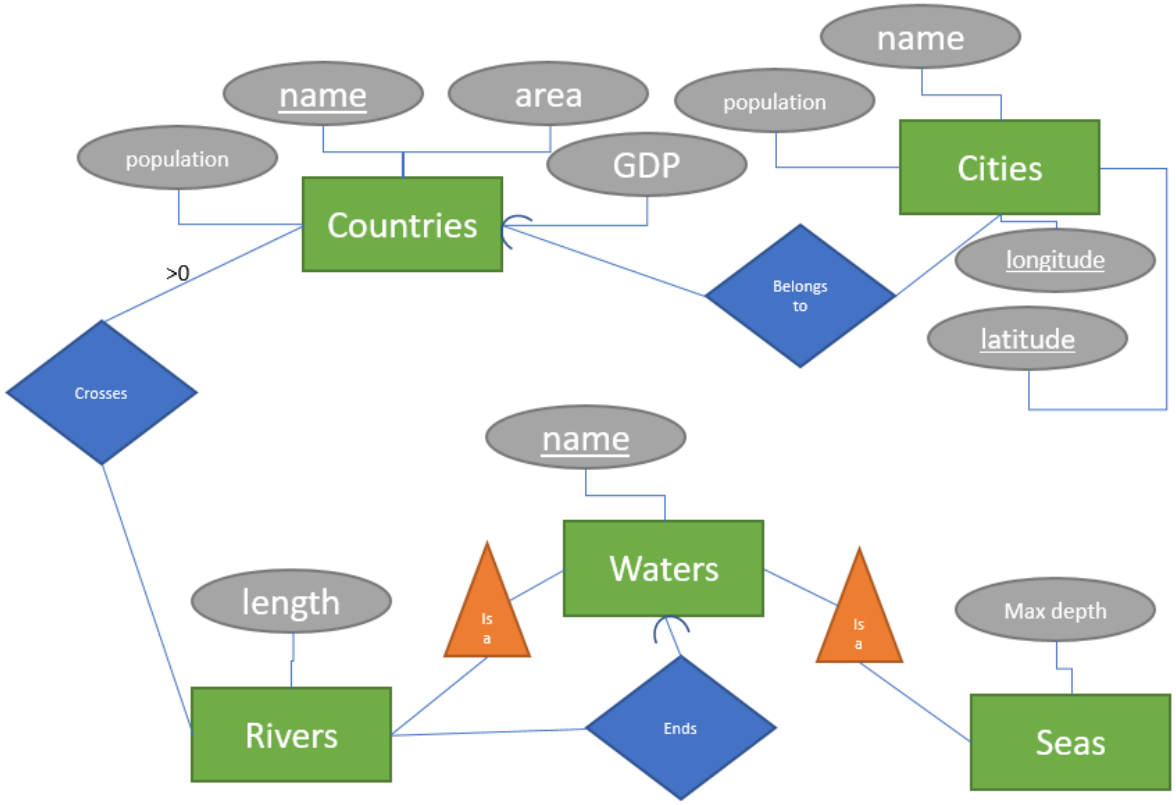
1. ***(10 points) Design an E/R diagram for geography that contains the following kinds of objects or entities together with the listed attributes. Model the relationships between the objects with edges. Note that edges between entities can be labeled with constraints.***

Entities

- ***countries (with attributes): name, area, population, gdp ("gross domestic product")***
  - ***a country's name uniquely identifies the country within all countries***
- ***cities: name, population, longitude, latitude***
  - ***a city is uniquely identified by its (longitude, latitude) (not by name, since for instance there are 41 different cities and towns are named Springfield in the US!)***
- ***rivers: name, length***
- ***seas: name, max depths***
  - ***rivers and seas are uniquely identified within all water entities by their name (e.g., "Ganges" would be a unique water entity)***

Relationships:

- ***each city belongs to exactly one country***
- ***each river crosses one or several countries***
- ***each country can be crossed by zero or several rivers***
- ***each river ends in either a river or in a sea***



2. (20 points) Consider the following E/R diagram:

Note: License plate numbers can have both letters and numbers in them. License numbers only contain numbers. Social Security numbers are only made up with numbers.

Translate the diagram above by writing the SQL CREATE TABLE statements to represent this E/R diagram. Include all key constraints; you should specify both primary and foreign keys. Make sure that your statements are syntactically correct (you might want to check them using sqlite / Azure for instance). (10 points)

```
DROP TABLE IF EXISTS CarDriver;
DROP TABLE IF EXISTS Truck;
DROP TABLE IF EXISTS Car;
DROP TABLE IF EXISTS Vehicle;
DROP TABLE IF EXISTS ProfessionalDriver;
DROP TABLE IF EXISTS NonProfessionalDriver;
DROP TABLE IF EXISTS Driver;
DROP TABLE IF EXISTS Person;
DROP TABLE IF EXISTS InsuranceCompany;

CREATE TABLE InsuranceCompany (
name varchar(256) PRIMARY KEY,
phone varchar(20)
)

CREATE TABLE Person (
ssn bigint PRIMARY KEY,
name varchar(256)
)

CREATE TABLE Driver(
licenseNo bigint PRIMARY KEY,
ssn bigint,
FOREIGN KEY(ssn) REFERENCES Person(ssn)
)

CREATE TABLE NonProfessionalDriver(
licenseNo bigint PRIMARY KEY,
FOREIGN KEY(licenseNo) REFERENCES Driver(licenseNo)
)

CREATE TABLE ProfessionalDriver(
licenseNo bigint PRIMARY KEY,
medicalHistory varchar(256),
FOREIGN KEY(licenseNo) REFERENCES Driver(licenseNo)
)

CREATE TABLE Vehicle (
licensePlate varchar(20) PRIMARY KEY,
year bigint,
insuranceName varchar(256),
maxLiability bigint,
maxLossDamage bigint,
ownerSsn bigint,
FOREIGN KEY(insuranceName) REFERENCES InsuranceCompany(name),
FOREIGN KEY(ownerSsn) REFERENCES Person(ssn)
```

```
)

CREATE TABLE Car (
licensePlate varchar(20) PRIMARY KEY,
make varchar(256),
FOREIGN KEY(licensePlate) REFERENCES Vehicle(licensePlate)
)

CREATE TABLE Truck (
licensePlate varchar(20) PRIMARY KEY,
capacity bigint,
driverLicenseNo bigint,
FOREIGN KEY(licensePlate) REFERENCES Vehicle(licensePlate),
FOREIGN KEY(driverLicenseNo) REFERENCES ProfessionalDriver(licenseNo),
)

CREATE TABLE CarDriver (
licensePlate varchar(20),
driverLicenseNo bigint,
PRIMARY KEY (licensePlate, driverLicenseNo),
FOREIGN KEY(driverLicenseNo) REFERENCES NonProfessionalDriver(licenseNo),
FOREIGN KEY(licensePlate) REFERENCES Car(licensePlate)
)
```

Which relation in your relational schema represents the relationship "insures" in the E/R diagram and why is that your representation? (5 points)

The insures relationship is represented by the columns insuranceName,maxLiability,maxLossDamage in Vehicle relation. Since a vehicle cannot have more than one insurance info, I have represented in the main vehicle relation itself.

Compare the representation of the relationships "drives" and "operates" in your schema, and explain why they are different. (5 points)

Since a truck can have at most one operator, the relationship is expressed in the truck relation itself by adding the operator's license number. But in the car case, a car can be driven by multiple drivers and hence it is represented using a separate table.

3. (20 points) Consider the following two relational schemas and sets of functional dependencies:

R(A,B,C,D,E) with functional dependencies D → B, CE → A.

S(A,B,C,D,E) with functional dependencies A → E, BC → A, DE → B.

For each of the above schemas, decompose it into BCNF. Show all of your work and explain, at each step, which dependency violations you are correcting. Make sure you describe each step in your decomposition steps. (10 points each)

a. **Relational Schema:** R(A,B,C,D,E)

Functional Dependencies: D → B, CE → A.

Iteration 1:

Let's take the set {C,E}.

{C,E}+ = {C,E,A}

{C,E}+ != {C,E} and {C,E}+ != {A,B,C,D,E}

So R(A,B,C,D,E) can be decomposed into R1(C,E,A) and R2(C,E,B,D).

Iteration 2:

Let's take the set {D} in R2(C,E,B,D).

{D}+ = {D,B}

{D}+ != {D} and {D}+ != {C,E,B,D}

So R2(C,E,B,D) can be decomposed into R21(D,B) and R2(D,C,E).

Iteration 3:

We now have R(A,B,C,D,E) decomposed into **R1(C,E,A), R21(D,B) and R2(D,C,E).**

We can't find any X such that X!= X+ and X != [all attributes]. So this is the BCNF.

b. **Relational Schema:** *S(A,B,C,D,E)*

Functional Dependencies: *A → E, BC → A, DE → B.*

Iteration 1:

Let's take the set {B,C}.

{B,C}+ = {B,C,A,E}

{B,C}+ != {B,C} and {B,C}+ != {A,B,C,D,E}

So S(A,B,C,D,E) can be decomposed into S1(B,C,A,E) and S2(B,C,D).

Iteration 2:

Let's take the set {A} in S1(B,C,A,E).

{A}+ = {A,E}

{A}+ != {A} and {A}+ != {B,C,A,E}

So S1(B,C,A,E) can be decomposed into S11(A,E) and S12(A,B,C).

Iteration 3:

We now have *S(A,B,C,D,E)* decomposed into **S11(A,E), S12(A,B,C) and S2(B,C,D).**

We can't find any X such that X!= X+ and X != [all attributes]. So this is the BCNF.

4. (15 points) A set of attributes X is called closed (with respect to a given set of functional dependencies) if X+=X. Consider a relation with schema R(A,B,C,D) and an unknown set of functional dependencies. For each closed attribute set below, give a set of functional dependencies that is consistent with it.

- ***All sets of attributes are closed.***

  Let's assume there is a FD A -> B. If this exists, then {A}+ = {A,B} != {A}.

  So the presence of any FD implies that set containing the left hand side of that FD won't be closed. So to have all sets of attributes to be closed, there should **Zero Functional Dependencies.**

- ***The only closed sets are {} and {A,B,C,D}.***

  Let's assume the following FDs:

  A -> B

  B -> C

  C -> D

  D -> A

  If one of the attributes is in the left hand side of the FD (say X), then all attributes will be on the right in X+. So for this case, the only closed attributes will be {} and {A,B,C,D}. So the required FDs for this case are

  A -> B

  B -> C

  C -> D

  D -> A

- ***The only closed sets are {}, {A,B}, and {A,B,C,D}.***

  Let's assume the following FDs:

  A -> B

B -> A

This would result in the following:

{A}+ = {A,B} – not closed

{B}+ = {A,B} – not closed

{A, B}+ = {A,B} – closed

{C}+ = {C} – closed

{D}+ = {D} – closed

Let's add the following FDs now and reevaluate.

C -> D

D -> C

{A}+ = {A,B} – not closed

{B}+ = {A,B} – not closed

{A, B}+ = {A,B} – closed

{C}+ = {C,D} – not closed

{D}+ = {C,D} – not closed

{C,D}+ = {C,D} – closed

Still we are missing a FD. Let's add another FD:

CD -> A

Now {C,D}+ = {C,D, A} – not closed

So the required FDs for this closed set is

A -> B

B -> A

C -> D

D -> C

CD -> A