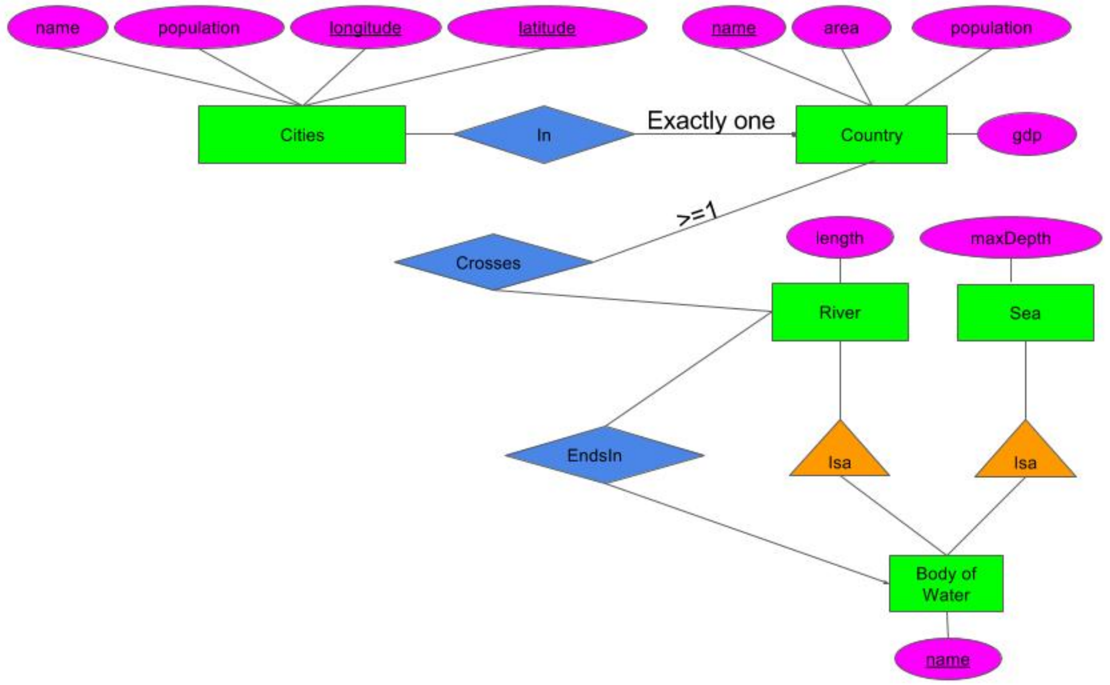

1.
2. a. The relational schema is listed below. Key constraints, as usual, are specified by underlining. For foreign key constraints, we have used the convention that a field named B.key in table A is a foreign key referencing B. We have not included the "create table" statements; they can be easily obtained from the relational schema. InsuranceCo(name, phone)

Vehicle(licencePlate, year, InsuranceCo.name, maxLiability, maxLossDamage, Person.ssn)

Car(Vehicle.licencePlate, make)

Truck(Vehicle.licencePlate, capacity, ProfessionalDriver.Driver.licenceNo)

Person(ssn, name)

Driver(licenceNo not null unique, Person.ssn) Driver(licenceNo, Person.ssn) Driver(licenceNo, Person.ssn unique)

OR

Driver(licenceNo not null unique);

ProfessionalDriver(Driver.licenceNo, medicalHistory)

NonProfessionalDriver(Driver.licenceNo)

Drives(Car.Vehicle.licencePlate, NonProfessionalDriver.Driver.licenceNo)

b. The "insures" relationship is many-one and this allows it to be included in the Vehicle relation. If a vehicle entity does not have insurance, these attributes will be NULL for it. c. Drives is many-many and Operates is many-one. Drives requires an additional relation, while operates can be inlined into the Truck relation.

3.

```
with temp as
  (select B.p1 as p1, C.p1 as p3, count(*) as y
   from person_living A, parent_child B, parent_child C
   where A.x=B.p1 and B.p2=B.p1
   group by B.p1, C.p1)
select T.p1, max(y) as z from temp T group by T.p1;
```

4.

```
T1(p,name) = Person_living(p) Join[p=name] Male(name)
T2(p) = Pi[p](T1(p,name))
T3(p1,p2,name) = Parent_child(p1,p2) Join[p1=name] Female(name)
T4(p2) = Pi[p2](T3(p1,p2,name))
T5(p) = rename[p] T4(p2)
Answer(p) = T2(p) - T5(p)
```