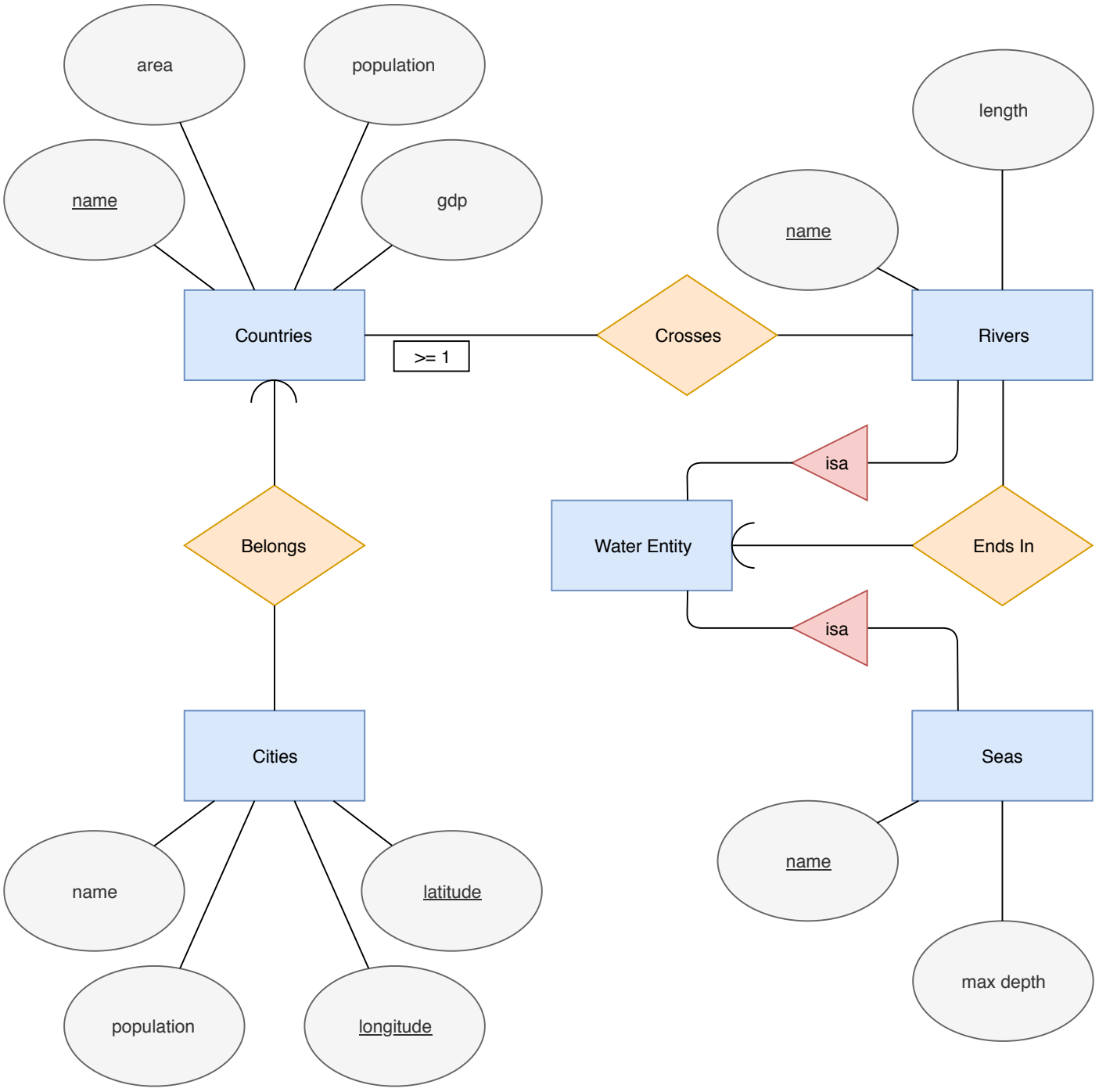
HW7 Part 1

1.

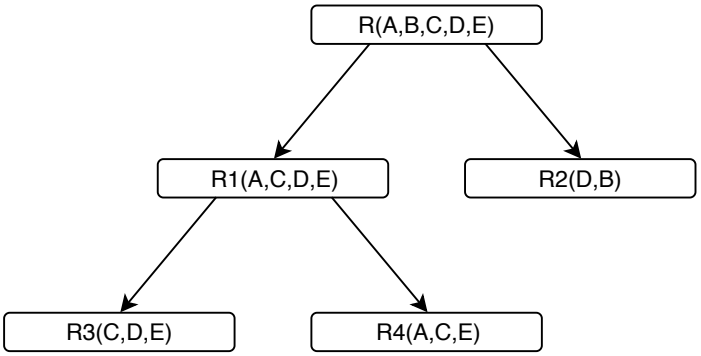


2.

```
DROP TABLE IF EXISTS InsuranceCo;
DROP TABLE IF EXISTS Person;
DROP TABLE IF EXISTS Driver;
DROP TABLE IF EXISTS NonProfessionalDriver;
DROP TABLE IF EXISTS ProfessionalDriver;
DROP TABLE IF EXISTS Vehicle;
DROP TABLE IF EXISTS Car;
DROP TABLE IF EXISTS Truck;
DROP TABLE IF EXISTS Drives;
```

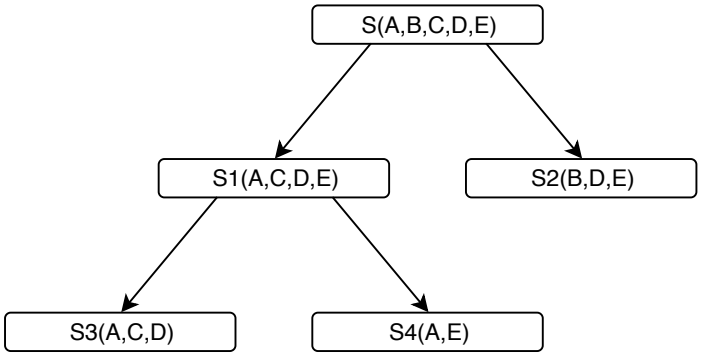
```
CREATE TABLE InsuranceCo(
   iName varchar(30) PRIMARYKEY,
   phone int
);

CREATE TABLE Person(
   SSN NUMBER PRIMARY KEY,
   name varchar(30)
);

CREATE TABLE Driver(
   SSN NUMBER PRIMARY KEY
      CHECK(licensePlate IN (select Person.SSN
                             from Person)),
   dirverID NUMBER
);

CREATE TABLE NonProfessionalDriver(
   SSN NUMBERP RIMARY KEY
      CHECK(licensePlate IN (select Driver.SSN
                             from Driver)),
);

CREATE TABLE ProfessionalDriver(
   SSN NUMBERPRIMARYKEY
      CHECK(licensePlate IN (select Driver.SSN
                             from Driver)),
   medicalHistory varchar(100)
);

CREATE TABLE Vehicle(
   licensePlate varchar(9) PRIMARY KEY,
   year int,
   maxLiability REAL,
   iName varchar(30) REFERENCES InsuranceCo(name),
   pSSN NUMBERREFERENCES Person(SSN)
);

CREATE TABLE Car(
   licensePlate varchar(9) PRIMARY KEY
      CHECK(licensePlate IN (select Vehicle.licensePlate
                             from Vehicle)),
   make varchar(30)
);

CREATE TABLE Truck(
   licensePlate varchar(9) PRIMARY KEY
      CHECK(licensePlate IN (select Vehicle.licensePlate
                             from Vehicle)),
   capacity int,
   pdSSN NUMBER REFERENCES ProfessionalDriver(SSN)
);

CREATE TABLE Drives(
   licensePlate varchar(9) REFERENCES Car(licensePlate),
   npdSSN NUMBER REFERENCES NonProfessionalDriver(SSN),
   PRIMARYKEY(licensePlate, npdSSN)
);

/* relationship "insures" is represented by joining Vehicle
** and InsuranceCo using foreign key iName in Vehicle since
** it's a many to one relationship so that each Vehicle will
** connect to at most one InsuranceCo and exactly one maxliability
** in the relationship, thus maxliability is stored in Vehicle
*/

/* relationships "drives" and "operates" are different because
** Drives is a many to many relationship, but Operates is a
** many to one relationship. Therefore, Truck stored a foreign
** key pdSSN to join with ProfessionalDriver, and Drives become
** a table storing unique pairs of licensePlate and npdSSN.
*/
```

3.



R(A,B,C,D,E)
FD: D → B, CE → A

$\{D\}^+ = \{B,D\}$, so R(A,B,C,D,E) violates BCNF. Then R has been split into R1(A,C,D,E) and R2(D,B), and R2 satisfy BCNF. R1 still violates BCNF since $\{C,E\}^+ = \{A,C,E\}$, so R1 has been split into R3(C,D,E) and R4(A,C,E). Now R3 and R4 satisfy BCNF since $\{C,D,E\}^+ = \{A,B,C,D,E\}$.



S(A,B,C,D,E)
FD: A → E, BC → A, DE → B

$\{D,E\}^+ = \{B,D,E\}$, so S(A,B,C,D,E) violates BCNF. Then S has been split into S1(A,C,D,E) and R2(B,D,E), and S2 satisfy BCNF. S1 still violates BCNF since $\{A\}^+ = \{A,E\}$, so S1 has been split into S3(A,C,D) and S4(A,E). Now S3 and S4 satisfy BCNF since $\{A,C,D\}^+ = \{A,B,C,D,E\}$ and FD BC→A doesn't work anymore since there's no B in S3.

4.

R(A,B,C,D)
(a). All sets of attributes are closed.
FD: N/A
(b). The only closed sets are {} and {A,B,C,D}.
FD: A->B, B->C, C->D, D->A
(c). The only closed sets are {}, {A,B}, and {A,B,C,D}.
FD: A->B, B->A, C->D, D->AC