HW7 Part 1

1.

Entities and relationships:

- **Countries** with attributes: area, population, name (key), gdp
- **Countries** —— >= 1 —— **Crosses** —— **Rivers**
- **Rivers** with attributes: name (key), length
- **Rivers** isa **Water Entity**
- **Seas** isa **Water Entity**
- **Water Entity** —— **Ends In** —— **Rivers**
- **Countries** —— **Belongs** —— **Cities**
- **Cities** with attributes: name, population, latitude (key), longitude (key)
- **Seas** with attributes: name (key), max depth

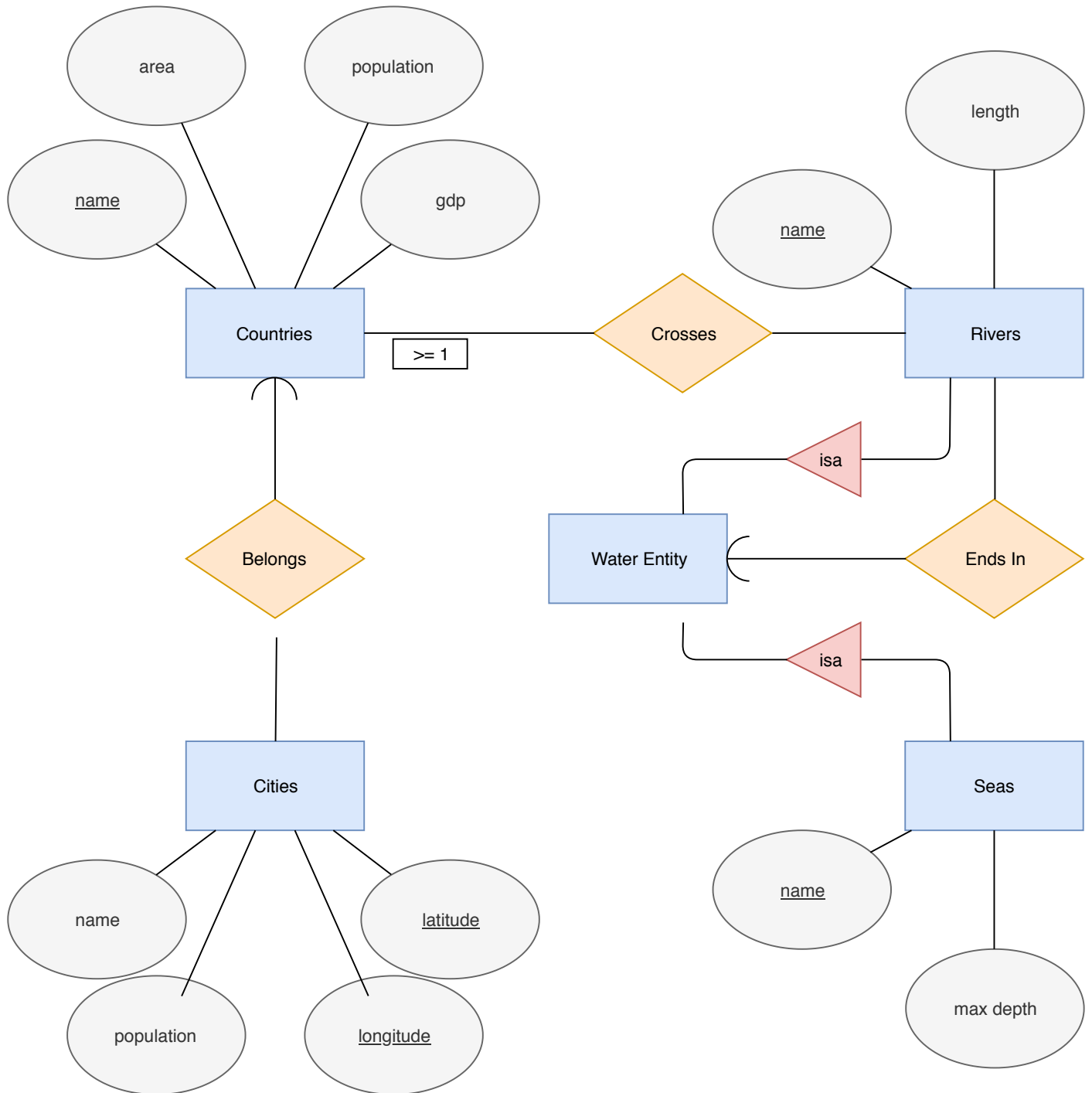
2.

```
CREATE TABLE InsuranceCo(
name varchar(30) PRIMARY KEY,
phone int
);

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

```sql
CREATE TABLE Driver(

SSN int PRIMARY KEY REFERENCES Person(SSN),

dirverID int

);
```

```sql
CREATE TABLE NonProfessionalDriver(

SSN int PRIMARY KEY REFERENCES Person(SSN)

);
```

```sql
CREATE TABLE ProfessionalDriver(

SSN int PRIMARY KEY REFERENCES Person(SSN),

medicalHistory varchar(100)

);
```

```sql
CREATE TABLE Vehicle(

licensePlate varchar(9) PRIMARY KEY,

year int,

maxLiability REAL,

iName varchar(30) REFERENCES InsuranceCo(name),

pSSN int REFERENCES Person(SSN)

);
```

```sql
CREATE TABLE Car(

licensePlate varchar(9) PRIMARY KEY REFERENCES Vehicle(licensePlate),

make varchar(30)

);
```

```sql
CREATE TABLE Truck(

licensePlate varchar(9) PRIMARY KEY REFERENCES Vehicle(licensePlate),

capacity int,

pdSSN int REFERENCES ProfessionalDriver(SSN)

);
```

```sql
CREATE TABLE Drives(

licensePlate varchar(9) REFERENCES Car(licensePlate),

npdSSN int REFERENCES NonProfessionalDriver(SSN),

PRIMARY KEY(licensePlate, npdSSN)

);
```

```sql
/* 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)

R1(A,C,D,E)   R2(D,B)

R3(C,D,E)   R4(A,C,E)
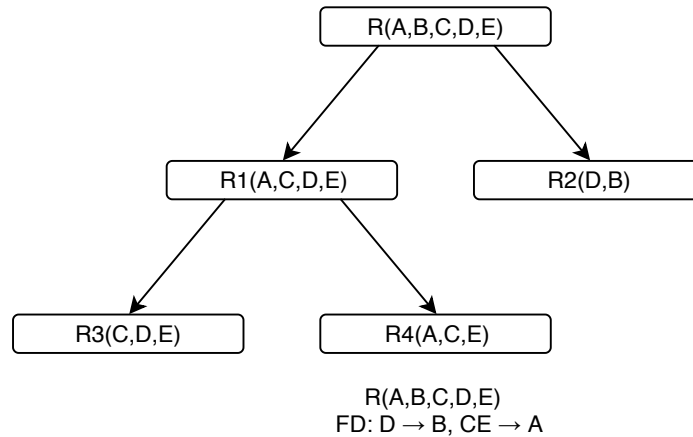
R(A,B,C,D,E)
FD: D → B, CE → A

$\{D\}^+ = \{B,D\}$, 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)

S1(A,C,D,E)   S2(B,D,E)

S3(A,C,D)   S4(A,E)
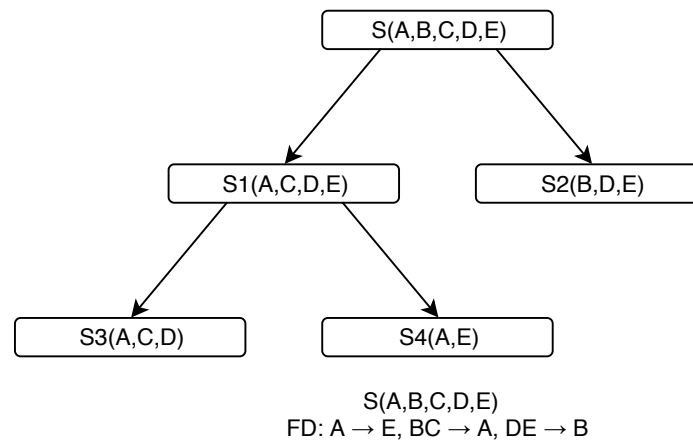
S(A,B,C,D,E)
FD: A → E, BC → A, DE → B

$\{D,E\}^+ = \{B,D,E\}$, 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