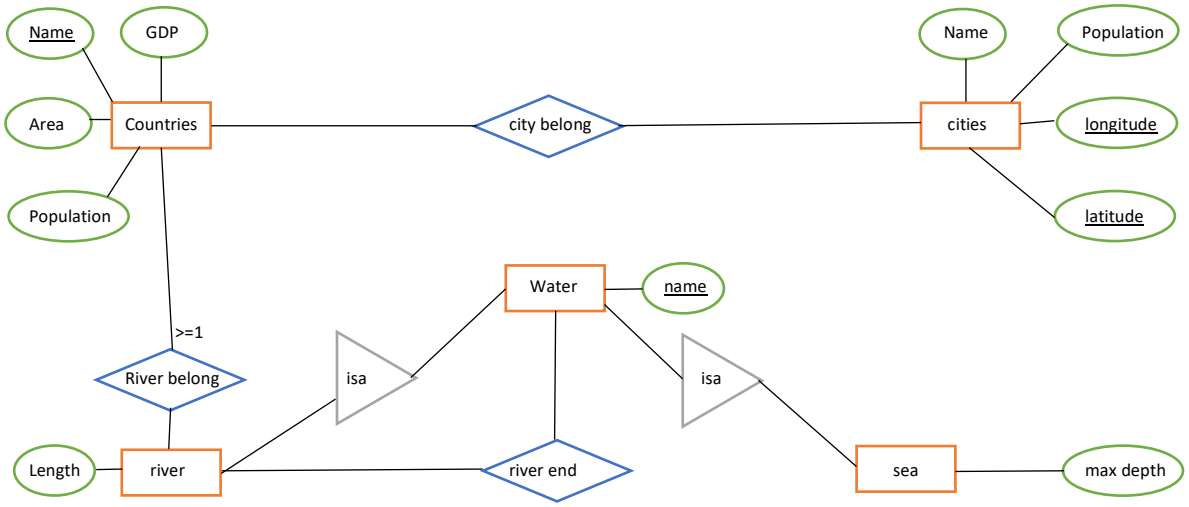
Part 1

1.



2.

a).

```
CREATE TABLE InsuranceCo (
        name varchar(20) PRIMARY KEY,
        phone INT
);

CREATE TABLE Vehicle (
        licensePlate text PRIMARY KEY,
        year INT,
        maxLiability REAL,
        name varchar(20),
        ssn NUMBER,
        FOREIGN         KEY(ssn) references Person(ssn),
        FOREIGN KEY(name) references InsuranceCo(name)
);

CREATE TABLE Person (
```

```
        ssn NUMBER PRIMARY KEY,
        name varchar(20)
);

CREATE TABLE Driver (
        ssn NUMBER PRIMARY KEY,
        driverID NUMBER,
        FOREIGN KEY(ssn) references Person(ssn)
);

CREATE TABLE NonProfessionalDriver (
        ssn NUMBER PRIMARY KEY,
        FOREIGN KEY(ssn) references Driver(ssn)
);

CREATE TABLE ProfessionalDriver (
        ssn NUMBER PRIMARY KEY,
        medicalHistory varchar(50),
        FOREIGN KEY(ssn) references Driver(ssn)
);

CREATE TABLE Car (
        make varchar(30),
        licensePlate TEXT PRIMARY KEY,
        FOREIGN KEY(licensePlate) references Vehicle(licensePlate)
);

CREATE TABLE Truck (
        capacity varchar(10),
```

```
        licensePlate TEXT PRIMARY KEY,
        ssn NUMBER
        FOREIGN KEY(licensePlate) references Vehicle(licensePlate),
        FOREIGN KEY(ssn) references ProfessionalDriver(ssn)
    );

    CREATE TABLE Drives (
        licensePlate TEXT,
        ssn NUMBER,
        FOREIGN KEY(licensePlate) references Car(licensePlate),
        FOREIGN KEY(ssn) references NonProfessionalDriver(ssn)
    );
```

b).

The Insures relation is represented by the adding the name attribute in Vehicle which is a foreign key refers to InsuranceCo name. Insures is a many to one relation where one InsuranceCo can insure many vehicles. By doing so, it is guaranteed that every licensePlate will have at most one InsuranceCo.

c).

The relation Drives between Car and NonProfessional Driver is many to many, the relation operates between Truck and Operates is many to one. For many to one relation, the primary key of ProfessionalDriver could be added to Truck since for every truck there could be at most one ProfessionalDriver. For many to many relation since keys appears repeatedly, so it could only be represented by a new table.

3.

a). R(A,B,C,D,E) FD: {D->B, CE->A}

(1) Decomposing: {A,B,C,D,E}

FD => D+ = {D,B}

a. Dependency Violation: D+ != {D} or {A,B,C,D,E}

b. Decomposes to {D,B} and {A,C,D,E}

(2) Decomposing: {A,C,D,E}

FD => CE+ = {C,E,A}

a. Dependency Violation: CE+ != {C,E} or {A,C,D,E}

b. Decomposes to {C,E,A} and {C,D,E}

(3) BCNF = {D,B}, {C,E,A}, {C,D,E}

b). S{A,B,C,D,E} FD: {A->E,BC->A,DE->B}

(1) Decomposing: {A,B,C,D,E}

FD => A+ = {A,E}

a. Dependency Violation: A+ != {A,E} or {A,B,C,D,E}

b. Decomposes to {A,E} and {A,B,C,D}

(2) Decomposing: {A,B,C,D}

FD => BC+ = {B,C,A}

a. Dependency Violation: BC+ != {B,C} or {A,B,C,D}

b. Decomposes to {B,C,A} and {B,C,D}

(3) BCNF = {A,E}, {B,C,A}, {B,C,D}

4. R(A,B,C,D)

(1) {A->A,B->B,C->C,D->D}

(2) {A->B,B->C,C->D,D->A}

(3) {A->B,B->A;C->DAB,D->CAB}