Homework 5

(20 points)

Overview

Objectives: To be able to translate an entity relationship diagrams to a relational database, and to understand functional dependencies and normal forms.
What to Submit: Submit a single `.pdf` file with your answers to Gradescope
Due Date: Check Gradescope
Formatting Instructions: link

> ⚠ **Warning**
>
> **When submitting on Gradescope, it is your responsibility to assign pages to correct questions or they may go ungraded. With so much manual grading we unfortunately do not have the time to hunt down the location of your answers in the PDF.**

Questions

1. (4 points) Design an ER diagram **using Crow's feet notation** for geography that contains the following kinds of objects or entities together with the listed attributes.

   Entities

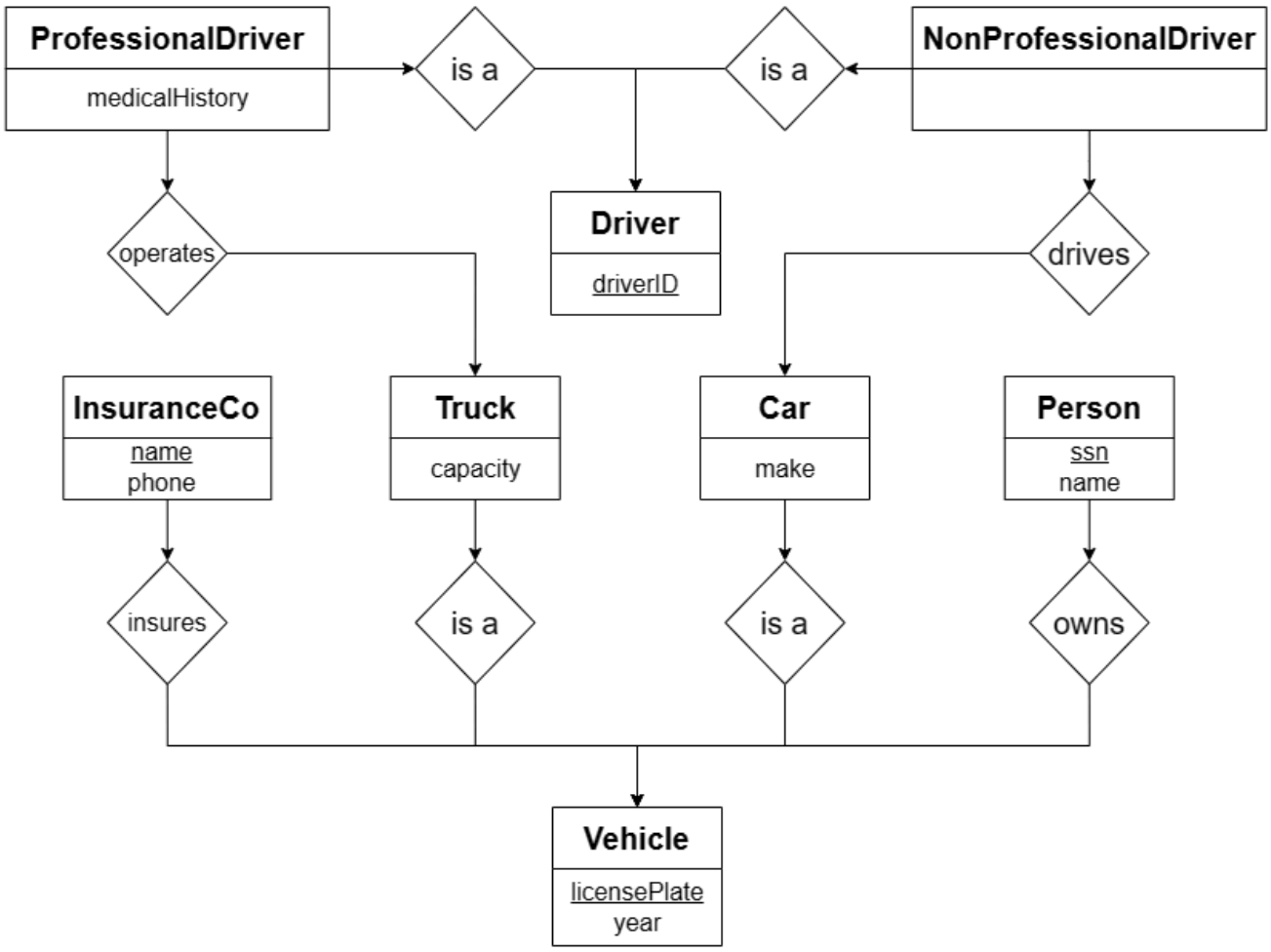
   - Countries: `name`, `area`, `population`, `gdp` ("gross domestic product")
     - a country's name uniquely identifies the country
     - every country has at least one city
   - Cities: `name`, `population`, `longitude`, `latitude`
     - a city is uniquely identified by its (longitude, latitude) (not by name, eg. there are 41 different cities and towns named Springfield in the US!)
   - Rivers: `name`, `length`
   - Seas: `name`, `maxDepth`
     - 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 at least one Country
- each Country can be crossed by zero or more Rivers
- each River ends in either one River or one Sea (not in both and not in many)

Model the relationships between the objects. Note the relationships and make sure to include any Primary and Foreign Keys. E.g. you may need to create a `river_id` for a primary key, even though we did not specify `river_id` as an attribute. You can also create composite keys when appropriate, be sure to set both attributes to `PK` in the ER diagram if you do so. Don't forget to create mapping tables for *many-to-many* relationships.

2. Consider the following flowchart. It somewhat contains the schema of a database but it is incomplete. Here are some of the attributes and their data type
    - `licensePlate` can have both letters and numbers
    - `driverID` and `ssn` contain only numbers
    - `maxLiability` is a real number (not shown in the diagram) that determines the maximum liability that the insurance company handles for a particular vehicle
    - `year`, `phone`, `capacity` are integers;
    - everything else is a string.



Note

- Assume that you must have exactly one insurance for a vehicle
- Try to follow what is shown in the flowchart and make the most logical assumptions and avoid unnecessary complications

- 2.1 (2 points) Convert the flowchart above into an ER diagram using Mermaid and Crow's feet notation. Make sure that the ERD is complete with all attributes (you may have to make up a few) and the relationship between tables is represented correctly. Ensure that the attributes have all the necessary information such as whether it is a primary key or foreign key and its data type. Again, don't forget to create mapping tables for any *many-to-many* relationships.
- 2.2 (2 points) Write SQL `CREATE TABLE` statements to represent this E/R diagram. Include all key constraints. Make sure that your statements are syntactically correct. You might want to check them using sqlite3/postgresql.
- 2.3 (1 point) How did you represent the relationship "insures" in your E/R diagram and why is that your representation?

- 2.4 (2 points) How did you represent the relationships "owns", "drives" and "operates". How and why is this representation different from the representation in 2.3?

3. For each of the schemas below, decompose it into BCNF. Show all of your work and explain, at each step, which dependency violations you are correcting.

- 3.1. (2 points) R(A, B, C, D, E) with functional dependencies
  D → B
  CE → A
- 3.2. (2 points) S(A, B, C, D, E) with functional dependencies
  A → E
  BC → A
  DE → B

4. 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.
   - Also, write down the closure for all sets of attributes, i.e. for every sub question write. $\{\}^+, A^+, B^+, C^+, D^+, AB^+, AC^+, AD^+, BC^+, BD^+, CD^+, ABC^+, ABD^+, ACD^+, BCD^+, ABC$
     I feel your pain, but it is important as it helps to check if your answer is correct or not

- 4.1 (2 points) All sets of attributes are closed.
- 4.2 (2 points) The only closed sets are *{}* and $\{A, B, C, D\}$.
- 4.3 (2 points) The only closed sets are *{}*, $\{A, B\}$, and $\{A, B, C, D\}$.