

# CSE 344 Homework 6: Conceptual Design

## Objectives:

To be able to translate from E/R diagrams to a relational database, and to understand functional dependencies and normal forms.

## Reading assignments:

Chapters: 3.1-3.4, 4.1-4.6

## Assignment tools:

SQLite (or any other relational DBMS) for problem 5 below.

## Additional files:

[mrFrumbleData.txt](#)

## Due date:

Thursday, February 27, 2014, at 11:59 pm

## Drop Box:

Turn in [here](#)

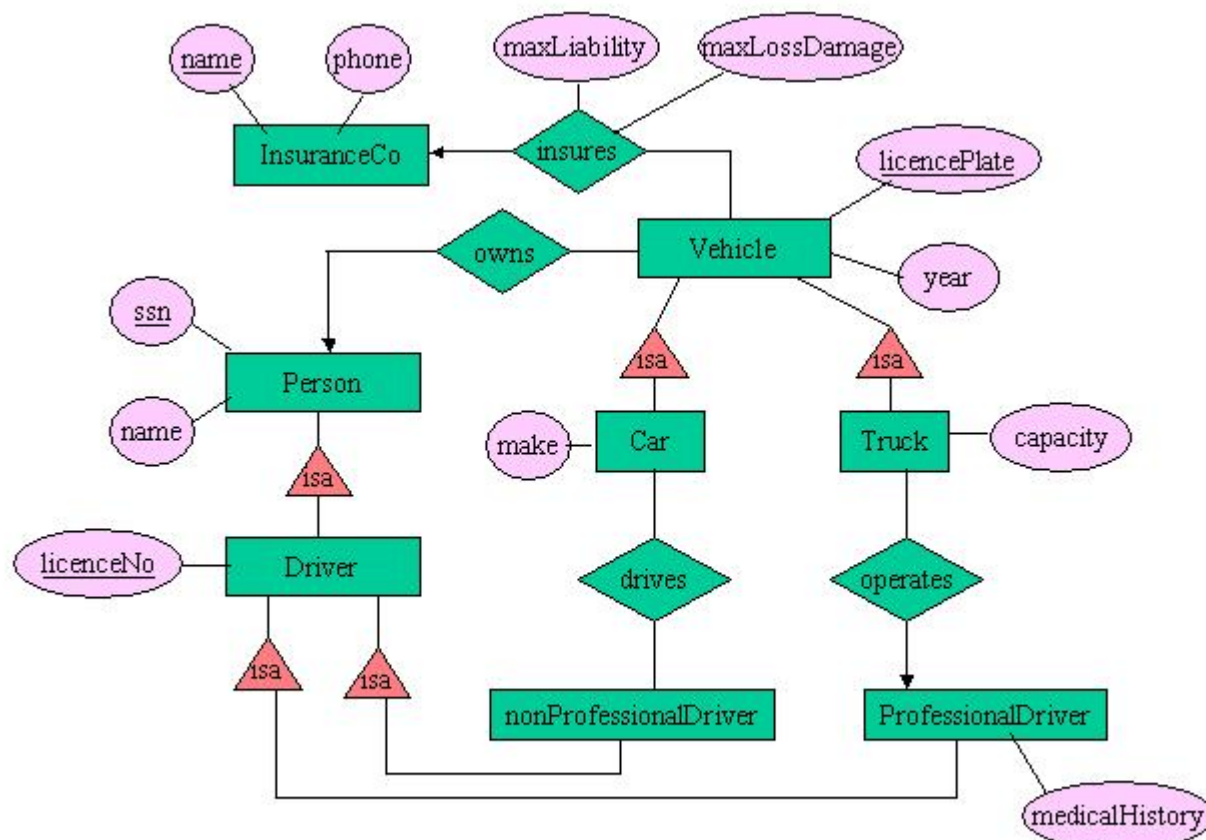
- [10 points] Design an E/R diagram for geography that contains the following kinds of objects together with the listed attributes:

- countries: name, area, population, gdp ("gross domestic product")
- cities: name, population, longitude, latitude
- rivers: name, length
- seas: name, max depths

Model the following relationships between the geographical objects:

- each city belongs to exactly one country
- each river crosses one or several countries
- each river ends in a river or in a sea

- [20 points] Consider the following E/R diagram:



- Write the SQL CREATE TABLE statements to represent this E/R diagram. Include all keys, foreign

keys, and uniqueness constraints. You do not need to run these commands in SQL.

- b. Which relation in your relational schema represents the relationship "insures", in the E/R diagram and why is that your representation?
- c. Compare the representation of the relationships "drives" and "operates" in your schema, and explain why they are different.

3. [20 points] Consider the following two relational schemas and sets of functional dependencies:

- i.  $R(A,B,C,D,E)$  with functional dependencies  $D \rightarrow B$ ,  $CE \rightarrow A$ .
- ii.  $S(A,B,C,D,E)$  with functional dependencies  $A \rightarrow E$ ,  $BC \rightarrow A$ ,  $DE \rightarrow B$ .

For each of them, decompose it into BCNF. Show all of your work and explain, at each step, which dependency violations you are correcting. You have to turn in a description of your decomposition steps.

4. [15 points] 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.

- a. All sets of attributes are closed.
- b. The only closed sets are  $\{\}$  and  $\{A,B,C,D\}$ .
- c. The only closed sets are  $\{\}$ ,  $\{A,B\}$ , and  $\{A,B,C,D\}$ .

5. [35 points] [Mr. Frumble](#) (who is a great character for small kids that always gets into trouble) designed a simple database to record projected monthly sales in his small store. He never took a database class, so he came up with the following schema:

**Sales(name, discount, month, price)**

He inserted his data into the database, then he realized that there is something wrong with it: it was difficult to update. He hires you as a consultant to fix his data management problems. He gives you this file [mrFrumbleData.txt](#) and says: "Fix it for me!". Help him by normalizing his database. Unfortunately you cannot sit down and talk to Mr. Frumble to find out what functional dependencies make sense in his business. Instead, you will reverse engineer the functional dependencies from his data instance. You should do the following steps:

- i. Create a table in the database and load the data from the provided file into that table; use SQLite or any other relational DBMS if your choosing. You don't need to turn in anything for this point.
- ii. Find all functional dependencies in the database. This is a reverse engineering task, so expect to proceed in a trial and error fashion. Search first for the simple dependencies, say **name**  $\rightarrow$  **discount** then try the more complex ones, like **name, discount**  $\rightarrow$  **month**, as needed. To check each functional dependency you have to write a SQL query. Your challenge is to write this SQL query for every candidate functional dependency that you check, such that (a) the query's answer is always short (say: no more than ten lines or so), and (b) you can determine whether the FD holds or not by looking at the query's answer. Try to be clever in order not to check too many dependencies, but don't miss potential relevant dependencies.

For this point you should turn in all functional dependencies that you found, and for each of them the SQL query that discovered it, together with the answer of the query.

- iii. Decompose the table in BCNF, and create SQL tables for the decomposed schema. Create keys and foreign keys where appropriate.

For this point turn in the SQL commands for creating the tables.

- iv. Populate your BCNF tables from Mr. Frumble's data. For this you need to write SQL queries that load the tables you created at point iii from the table you created at point i.

Here, turn in the SQL queries that load the tables, and the tables' contents after loading them (obtained by running `SELECT * FROM Table`).