CS 455 Principles of Database Systems

Homework 1: The Relational Model

Name:

Instructions: Print this assignment using single-side pages. Fill in your name above, and write in the solutions in the space provided below each question. You are allowed to use the back of each page. If you used any scratch paper to show your work, append those to the end. **Note:** It is important you use this format for gradescope.

Submission: After you've filled in the answers, scan all pages into a PDF, and submit to canvas.

Problems

1. What is the maximum number of superkeys that a relation with N attributes can have?

2. Does every relation have more superkeys than candidate keys? Explain.

3. Consider the following relations for a database that keeps track of auto sales from a used car dealership where prices can vary greatly based on the car's condition and age. First, identify the primary keys for each relation using reasonable real-world assumptions (e.g., vins are unique). Specify the foreign keys for this schema. For each foreign key that you find, specify the relation in which it exists, and the relation and attribute(s) to which it refers.

```
car(vin, model, make, price)                  options(vin, optionName, price)
sales(salespersonID, vin, date, price)        salesperson(salespersonID, name, email)
```

ADD MORE structure. Add some assumptions, tell them how to specify keys and FKs.

4. Consider again the set of relations given in the previous problem. The `car` relation is highly susceptible to redundancy. For instance, there could be many “Toyota Camarys” and “Honda Civics” with different vins.

   (a) Explain why having redundancies in `(model,make)` can be problematic for the future management of data. Use some real-life examples to illustrate. Ignore the fact that redundancies can be wasteful in terms of space.

   (b) As a database designer, what could you do to minimize redundancy in these tables? Show the resulting schema with primary key and foreign keys indicated.

5. Design a *schema diagram* for a company like Netflix that minimizes redundancy and NULL values. You should recall that an attribute cannot hold a list of values. Indicate all primary keys (underlined) and foreign key constraints (reference arrows). You need to store data on the following:

   - A Movie has a title, description, country of origin, and a release-year. Assume that there may be two movies with the same name (e.g., Godzilla), but they are never released in the same year. You may assign each movie a unique identifier, but you shouldn't need to.
   - An Actor has a name and a biography. Assume that there may be two actors with the same name. You may assign each actor a unique identifier.
   - A User has a userid and password. The company does not allow two different users to share the same userid.
   - You need to keep track of what movies the users have seen.
   - Users can rate movies from 1 to 5 stars. Note that they need not have seen the movie to rate it! A user can only rate a movie once (though that rating can change later).
   - Users can also comment on each actor and/or movie. A comment has a date, time, and the comment itself. Unlike ratings, users can leave a multiple comments on the same movie or actor.
   - Actors play certain roles in certain movies. Actors may also play multiple roles in the same movie (see: Eddie Murphy and Mike Myers).