NYU Tandon School of Engineering
Computer Science and Engineering
CS-UY 3083, Introduction to Database Systems, Fall 2019 Prof Frankl

HOMEWORK #1

Instructions: You may work alone or with a group of up to 3 people. Hand in your solutions via Gradescope as a single pdf file. Follow the Gradescope instructions to mark your solution to each problem or subproblem as indicated in the outline; otherwise the graders will have trouble finding them and will apply a small penalty to your score. **See student workflow section in GradeScope help to learn how to do this.** If you're working with a group, use GradeScope's group submission feature to indicate all members of your group.
Note: You may find it useful to use *draw.io* or another drawing tool to draw ER diagrams. Alternatively, you may draw them *neatly* by hand. You must use the notation used in class ... rectangles for entity sets, diamonds for relationship sets, etc. You may either use the double line / arrow notation for participation and cardinality contraints or the lowerbound .. uppperbound notation. Both were presented in class.

Problem 1
Draw an ER diagram, similar to those shown in class, with with two entity sets, `Person` and `Movie` and two relationship sets `Saw`, and `Favorite`. A Person has a unique ID; a name, composed of a first name and a last name; a class (such as Freshman, sophomore, etc); and some (zero, one, or multiple) e-mail addresses. Each movie has a title, a release year, and a synopsis. No two movies have the same title and the same release year. If a person saw a movie, they may rate it with some number of stars and they may write a comment about it. Each person has exactly one favorite movie. (We will not worry about whether a person can designate a movie as their favorite even if they didn't see it or saw it and gave it few stars or bad comments)

Problem 2

1. Draw an ER diagram, similar to those shown in class, with two entity sets, `Person` and `TVseries` and relationship set `Watch`. A Person has a unique name, composed of a first name and a last name; a class (such as Freshman, sophomore, etc); and some (zero, one, or multiple) e-mail addresses. A TVseries has a title, a year (the year it started), and a genre. The title and year, together, of a TVseries are unique; in other words it's possible for two different TVseries to have the same title or the same year, but no two TVseries have *both* the same title and the same year. A Person who Watches a TVseries can indicate how much they like it with a number of stars.

2. - List at least two elements of an entity set TVseries. List at least two elements of an the entity set Person.
   - List at least three elements a relationship set Watches.

   The Watches elements do not have to be true data ... it's OK to lie about what you or your friends watch, etc. In this list, you can identify people by their first and last name and identify movies by their title and release year.

3. Re-draw the ER diagram from part (1), adding attributes, entity sets, and/or relationship sets to model the following; indicate cardinality constraints for any relationship sets that are not many-to-many:

   - Each Person's date of birth
   - The frequency with which a person watches a TVSeries (e.g. indicating whether they watch it often or rarely, etc.)
   - actors, each of whom has a name (first name, last name), and a gender, and a date of birth;
   - actors appear (with regular roles) in TVseries; also indicate the role(s) that each actor plays.
   - A person's favorite actor. (A person may have at most one favorite actor).
   - Individual Episodes of a series. Each has a season number and an episode number that is unique for that particular TVseries. However episodes of different TVseries may have the same (season number, episode number). An Episode also has a synopsis.

- Episodes have guest actors who appear in that episode, even if they're not a regular in the TVSeries.

Draw *one* ER diagram that includes the entity sets and relationship sets from part 1 and all of the stuff listed in part 3. Where possible, use the nouns, verbs, and prepositions from the description above to name your entity sets and relationship sets.

Problem 3 Suppose you're designing a database for a restaurant. The restaurant manager says they want to keep track of customers who are currently seated, dishes (e.g. "hamburger", "fried chicken", etc). Customers are identified by their table number and their seat number (at the table), e.g. Table 5, Seat 2. Each dish has a unique name, a description, a category (e.g. appetizer, main dish, dessert) and a price. The database will keep track of which dishes were ordered by which customer, along with the status (e.g. "in preparation", "ready","served, finished", etc).

1. Draw an ER diagram modeling this information. It should have an entity set representing customers, an entity set representing dishes, and one relationship set.

2. While reviewing this ER diagram with you, the restaurant manager realizes that some of the dishes have different sizes with different prices (e.g., "small tomato soup for $3.00 and large tomato soup for $5.00). Draw a new ER diagram to deal with this. Hint: use a weak entity set. Think about which entity sets participate in the relationship set representing orders.

3. You did such a good job on the database for "eat-in" orders, that the restaurant has hired you to design a database for their online orders. In this case, customers do not have table numbers and seat numbers, but each customer has a unique phone number and an address, composed of a building number, street name, and apartment number. In this scenario, a customer may order multiple servings of the same dish (e.g. three hamburgers). The database will keep track of how many of which dish was ordered by which customer, along with its status (e.g. "in preparation", "ready","out for delivery", etc). The database only keeps track of current orders and you may assume that a customer can't change an order once it's in the system. Draw an ER diagram for the online orders.

4. Now the manager decides that the restaurant would also like to keep track of the date and time on which each order was made and keep historical data, so that they'll know which customers have ordered which dishes in the past. Note that adding date and time attributes to the ordered relationship set is not sufficient, as this still will not allow the a customer to order the same dish at different date/times. (Why not?). Instead, you can take one of the following approaches:

   - use a ternary relationship set, involving an additional entity set representing dates/time;
   - or, change the ordered relationship set into an entity set representing orders, with relationship sets indicating who the order is by and what item is ordered). You may represent the orders with a strong entity set, adding an OrderNumber or with a weak entity set that has *two* identifying strong entity sets.

   Draw an ER diagram for the online orders with historical data.

Problem 4 Consider the bookstore E-R in Fig 6.30 in the 7th ed of textbook (Figure 7.29 in 6th edition) also posted in resources.
Submit one E-R diagram with the following modifications:

1. Modify the E-R diagram to indicate that a book is published by at most one publisher.
2. Modify the E-R diagram to indicate every book has at least one author.
3. Indicate that the basket_id is the primary key of the Basket Entity set (which is missing due to a typo in the diagram in the slides).
4. Modify the E-R diagram to indicate that a basket is owned by exactly one customer. (However a customer could have more than one basket.)

Problem 5 Each member of your group should individually (using their own NYU login), add at least three entries to our class person-saw-movie database, by submitting this form (at least three times): `https://docs.google.com/forms/d/e/1FAIpQLSeQcW7GMkegSx22I9FNnpzSu2O2Xc8WbtfrI8_lxdXqlJbwA/viewform` Please type the title of each movie exactly as it is

shown in IMDB. We will put this data into a database for some future examples and HW problems. Note: If you copy/paste the URL, delete the space between the underscores.