

Objective: Gain experience generating ER diagrams from requirements descriptions

Assignment:

Design ER diagrams for the two problems below based on the information given. Use Visio or another drawing program. Your diagrams may need to include entities, attributes, relationships, primary keys (PKs), and cardinality and participation constraints. Use the style of ER notation shown in Elmasri & Navthe, 6th edition, Figure 7.2 and Figure 7.14. Be sure to use good naming practice with your entities, relationships, and attributes. If you make any assumptions in addition to those given in the problem descriptions, include a note in the ER diagram to explain the assumption.

Problem #1: Apartment Complex ER diagram

The owners of an apartment complex need a database to keep track various aspects of the business. The complex has 35 units in 7 buildings. For each apartment, the database needs to keep track of the apartment number, the building number, the number of bedrooms, and the current rental price. For each tenant, the database should store the tenant's name, make and model of car (if any), and phone number. Each tenant can only rent one apartment, but an apartment can be rented by up to three tenants sharing the apartment. Some apartments may not be currently rented, but should still be listed in the database. The database should also record the name and employee id of each employee working for the owners. Tenants may contact employees to place service requests. The date and problem should be recorded for each service request, along with the employee who was contacted with the request.

Problem #2: Research Database ER diagram (from Stephanie Haas)

A research team needs a database to record information about an experiment they are conducting. Participants in the experiment will be assigned to one of two groups, answer a set of questions, view a text or video presentation about the topic, and then answer the same questions again (a pre-test/post-test experimental design). The team will use data recorded in the database to determine if there are any differences between the two groups' performance.

Example queries and reports:

- Compare demographic data of the two groups (e.g., average age).
- Calculate the time each participant took on the pre- and post-tests. Compare times within and between groups. (For example, did all participants complete the pre-test in approximately the same time? Was there a difference between groups on the post-test?)
- Compare pre- and post-test answers. Compare post-test answers between groups. Count the numbers of correct and incorrect answers on pre- and post-tests. Look at which researchers worked with which participants, to see if there were any differences among participants' times or responses.

Each research team member will administer the experiment to many participants, one participant at a time. The database records the first and last names of research team members, as well as the date they completed CITI certification (that qualifies them for working with human subjects).

The database records which group a participant was assigned to (text or video), and the date on which the participant did the experiment. It also records some demographic data: age, sex, highest degree earned, and current occupation.

The database stores the questions and answer choices for the experiment. Each participant is given the same set of questions and answer choices. The questions and answer choices are the same for the pre-test and the post-test. All subjects are given the questions in the same order. For each question, the database contains the question number, the text of the question, and the text of each of the 4 answer choices (A, B, C, and D). For example, the text of question 1 is "How important are cats to the global economy?" The texts of the

answer choices are: A) not at all, B) somewhat, C) a great deal, D) the world would fall apart if it weren't for cats.

The database records 4 timestamps: the time the participant starts reading the first question of the pre-test, the time the participant answers the last question of the pre-test, the time the participant starts reading the first question of the post-test, and the time the participant answers the last post-test question. It records the participant's answers for each question of the pre-test (before viewing the presentation) and each question of the post-test (after viewing the presentation).

Advice, Hints, and Guidance:

- Think about the difference between the process or work flow of running the experiment, and the information that should be stored in the DB.
- Don't add foreign keys (FK) to the ER: they represent relationships, and the ER represents relationships using the diamonds.
- Don't add any additional capabilities to the DB!
- Completeness. Does the model include all the important entities, actions, or ideas from the description or scenario?
- Over-completeness. Does the model include unimportant details, distracter items, or ideas not in the scope of the description or scenario?
- Assumptions. Does the model make unusual or unrealistic assumptions about the situation? A description or scenario may not completely specify the situation; you may need to make further assumptions. If you do:
 - Check to be sure the information isn't in the description,
 - Don't contradict (directly or by inference) information given in the description,
 - Make sure your assumptions are realistic and reasonable, and
 - Document any assumptions you make as part of your homework.
 - In real life, having to make assumptions might be a sign that you need to gather more information from your client.
- Naming. Does the model utilize good naming practice? Names should be taken from the description (the client) wherever possible. If no names are given, or the names are confusing, you should develop your own naming scheme. Names should be clear, descriptive, and non-repetitive.
- Clarity. A model is more effective if it is well-arranged on the paper or screen, but sometimes a model is complex. If it is very crowded on the page, and is confusing as a result, you should try to re-arrange or layer it, but don't fuss unnecessarily.

Grading:

I will be looking for accuracy, completeness, and clarity.

How to turn in your assignment:

Prepare your assignment using a word processor and drawing software. ALL assignments you submit should have your name, Onyen, homework number, and date at the top left of the first page.

Save your assignment into a SINGLE PDF file with one page for each problem in the assignment. Your ER diagram for each problem should fit on a single page.

Name your file according to the following convention:

`youronyen-h1.pdf`

Replace *youronyen* with your actual Onyen (e.g. my assignment would be `rcapra-h1.pdf`). The character between *youronyen* and the "h1.pdf" part should be a single minus sign (not an underscore). You

could also call this character a dash. There should be no spaces or other characters in the filename. Files with names that do not follow this convention will not be graded.

Submit your file electronically through the Sakai by going to the Assignments area and finding the “H1” assignment. After you think you have submitted the assignment, I strongly recommend checking to be sure the file was uploaded correctly and that you can view it by clicking on it from within Sakai. If I cannot open or view your file, I cannot grade it.

If for some reason you need to re-submit your homework file, you must add a version number to your filename so that we will know which file is the most recent. Sakai is configured so that it will only accept 3 total submissions. Use the following file naming convention if you need to re-submit:

Your first submission: youronyen-h1.pdf
Your second submission: youronyen-h1-v2.pdf
Your third submission: youronyen-h1-v3.pdf

Sakai is also configured with a due date and an “accept until” date. Submissions received after the due date (even just 1 minute!) will receive a 10% penalty per day. The “accept until” date is 5 days after the due date. Submissions will not be accepted after the “accept until” date and will have a score of zero recorded.