CS 281 Spring 2021
Project Topics

Carefully study the following project descriptions and try to capture:

- Strong and weak entities,
- Relationships (binary, ternary and specialization (IS-A) relationships), role indicators for an entity in a relationship, mapping cardinalities (1-1, 1-M, M-M) and aggregations (i.e., a relationship as a whole serve as an entity in another relationship),
- Keys.

REMARK: In each of the below scenarios, we try to provide examples of almost *all* of the above E/R elements, so please study these scenarios very carefully!

You are free (and indeed ***expected to***) add more details (and, thus entities/relationships) to the scenarios below. While designing your database, also keep in mind that in the upcoming stages of the project, you have to specify interesting/sophisticated queries over this database.

- *Hunger Games:* You are going to implement a database system to track the progress of tributes in an annual televised battle of Hunger Games. Users of this system are going to be mentors, game makers, and sponsors. For each user, you should store a unique SSN, name, surname, password, and date of birth. For game makers, you should also store a monthly salary, and for sponsors, you should also store a credit card number. For each game, you should store a unique year and a description. Each game is assigned to exactly one game maker for administrative purposes. A game maker can be assigned to multiple games. For each tribute, you should store a unique tribute number, name, surname, gender, district number, age, and status (“Healthy”, “Injured”, or “Dead”). Multiple tributes can compete in a game; however, a tribute can compete only in one game. Each tribute is given exactly one mentor, but a mentor could be assigned to many tributes. For each gift, you should store a unique gift name, price, and description. Sponsors can send gifts to multiple tributes; in this case, you should also store the amount of the gift that a sponsor sends to a tribute. Tributes cannot receive the gifts immediately, for this purpose the mentor of the tribute should authorize the gift; in this case, you should store the information about which mentor authorizes which gift sending operation, as well as an authorization date. You should store the interaction of tributes among each other, such as “B set a trap for A”. Each interaction should be stored using a source tribute and a target tribute, as well as a date and a description. Awards are given to the mentors for their accomplishments. Each award is uniquely identified by an award name and the SSN of the mentor who received the award. Each mentor can receive multiple awards, but an award only belongs to one mentor. Game makers can include additional rules for a game. In this case, you should store which game maker sets the rule for which game, as well as the content of the additional rule. There could be multiple additional rules for a game. Multiple game makers can set additional rules for a game.

- *Repair Company:* You are going to implement a database system for a company that offers repair services for a wide range of products. Users of this system are going to be technicians, managers, and customers. For each user, you should store a unique e-mail, password, name, and surname. For customers, you should also store a telephone number, and for technicians, you should store a monthly salary. For each address, you should store a unique address number, building number, street name, and city name. Each customer can have multiple addresses, but an address belongs to exactly one customer. For each service request, you should store a unique request number, request date, and description. All service requests should be associated with exactly one address. Furthermore, all service requests belong to a service type. There could be multiple service requests of the same service type. A service type is stored with a unique type number, type name, and description. Technicians can have a specialty in multiple service types, and there could be multiple technicians that have a specialty in the same service type. Managers forward service requests to technicians; in this case you should store which managers forwards which service request to which technician. Technicians resolve service requests by creating repair records. For each repair record, you should store a unique record number, resolve date, service fee, and description. Spare parts can be used in a repair record, in this case, you should also store the amount of the spare part that is used. Multiple spare parts can be used in a repair record, and a spare part can be used in multiple repair records. Spare parts are uniquely identified by the combination of manufacturer name and model number. For each spare part, you should also store stock amount, description, and unit price. Customers can review the service when it is complete. In this case, you should store a score in range [0:10] and a comment. Reviews should be associated with customers and repair records. There could be at most one review for the same repair record, but the same customer can review multiple repair records. Reviews should be approved by managers before it is publicly visible, in this case, you should store which manager approves which review on which date.

- *Detective Bureau:* You are going to implement a database system to keep track of evidence and crime. Users of this system are going to be investigators, field officers, and prosecutors. For each user, you should store a unique SSN, name, surname, phone number, password, and address. For field officers, you should also store a location. For each crime scene, you should store a unique scene number, address, and description. Field officers can find evidence in crime scenes. For each piece of evidence, you should store a unique evidence number, object name, description, and finding date. A piece of evidence can belong to exactly one crime scene, but in a crime scene, there could be multiple pieces of evidence. While including a piece of evidence in a crime scene, you should also record the field officer that is responsible for that evidence. Investigators assign field officers to crime scenes; multiple field officers can work at the same crime scene, and a field officer can work in multiple crime scenes. You should keep track of which investigator assigns which field officer to which crime scene on which date. For each suspect, you should store a unique phone number, name, surname, age, and gender. Investigators interrogate suspects to create reports. For each report, you should store a unique report number and report content. Each report should be associated with exactly one suspect and exactly one investigator. The same investigator can create multiple reports and there could be multiple reports for the same suspect. Prosecutors can charge suspects for their crimes, for this purpose an indictment is created. For each indictment, you should store a unique file number, status ("Ongoing", "Guilty", "Not Guilty"), crime description, and punishment description. Each indictment should be associated with exactly one suspect and exactly one prosecutor. Pieces of evidence can be related to indictments; in this case, you should also store a text that describes the role of this evidence. One evidence can be related to multiple indictments, and one indictment can be related to multiple evidence. Each piece of equipment is uniquely identified by an equipment name and the SNN of the field officer that owns this equipment. Each piece of equipment can belong to exactly one field officer, and the same field officer can have multiple pieces of equipment. For each piece of equipment, you should also store the date of registration and weight.