

1. Read Chapter 3.

- a. Compare and contrast the database and the software design processes. Are agile modeling practices applicable to database design as well?

Database design is significantly less adaptive than a agile software design. Databases need to be well thought out and future proofed well before they're implemented. Making simple changes can end up being far more complicated in a database than making changes to a piece of software to fit a customers needs.

- b. Explain the following terms: *entities*, *relationships* (including cardinality and participation constraints), *attributes* (simple, composite, multi-valued and derived).

An entity is something that must be able to exist independently of a table. For example, several separate entities could be joined in a single table by an attribute that they all share.

A relationship is what connects all entities in a table, whereas an attribute is anything that refers to or describes a database element.

- c. Explain *weak entities* and *identifying relationships*.

A weak entity cannot be identified by its attributes alone and requires a foreign key to make it unique. This key is its identifying relationship.

- d. Model a person-household database (from class 2) using an ERD.

\*In this model, each person would need to have an identifying key that would connect each person would be the household id (e.g. family name or id)

- e. Skim the material in Section 3.8 on UML diagrams; we won't use them here but we did in CS 262 and will again in unit 11.

2. Note the ERD reference sheet in Figure 3.14.

2. Read Chapter 9.

- a. Map the person-household ERD model from above to a relational database model.

- b. Explain how to map from a UML model to a relational database model. We will address this issue more carefully in a later unit (on object-relational mapping).

A UML diagram defines which relationships are one-to-one, many-to-one, one-to-many, and many-to-many. Knowing this, we can determine which tables will need a column for foreign keys, and which tables will need to be joined (for many to many or one to many). Additionally, we can determine which attributes will be foreign and primary keys.