Devere Anthony Weaver
COP4710 - Database Systems
Chapter 4 ER Textbook Notes

Chapter 4 - The Basic ER Diagram: A Data Modeling Schema

4.2 - What is a Data Modeling Schema?

data modeling schema - a method that allows us to model or illustrate a database
- often in the form of a graphic diagram although other means exist

Entity-Relationship diagram (ER) - graphic tool used to facilitate data modeling
- subset of *semantic models* (models intended to elicit meaning from data)
- goal is to abstractly describe or portray data
- conceptual model
- can be used to model to-be created database
- can be used to reverse engineer an existing database

Schema - a permanent, fixed description of the structure of the data
- when we agree that we have captures the correct depiction of reality within our conceptual model, ER diagram, this becomes the schema

Recognize there are levels of abstraction we can use in discussing databases
- abstract - stay in this level longer such that it is easier to change the model and then decide the physical implementation later
- concrete - details of logical model, physical realizations, files, etc

Mapping - process of choosing a logical model and then moving to a physical database file system from a conceptual model
- bridge between design concept and physical reality

4.3 - Defining a Database - Some Definitions

entity - a thing about which we store data
- a person, place, object, event, or concept
- name of entity should be generic and should be able to accommodate changes over time

relationship - connection between entities; link or association between entities
- usually denoted by verb phrases

attribute - category of data that describes an entity or relationship

entity sets - we store collections of data about entities

instance - an entity instance is a single occurrence of an entity

attribute - property or characteristic of an entity

Database modeling begins with a description of "what is to be stored"
- this usually comes from the user

ER Design Methodology
1. Select one primary entity from the database requirements description and show attributes to be recorded for that entity
    - once a primary entity and some attributes are choses, the next tasks will be to draw a diagram of the first impression entity, translate the diagram into English, and present the English back to the user for feedback

2. Use structured English for entities, attributes, and keys to describe the database that has been elicited.

3. Show sample data.

4.4 - A First "Entity-Only" ER Diagram: An Entity with Attributes

This section focuses on using the "Chen-like" diagram.
- entity represented in a box with ovals attached to represent the attributes of the entity
- several alternative versions exist, pick one

4.5 - More About Attributes

There exist several different kinds of attributes:
- simple, atomic
- composite
- multivalued
- derived

Properties of an attribute:
- name
- description
- formate
- length
- atomicity

Simple or Atomic Attribute

Can't be broken down further or subdivided.

If an attribute is non-atomic, this must be depicted in an ER diagram.

<u>Composite Attribute</u>

composite attribute - formed by combining or aggregating related attributes

In an ER diagram, these composite attributes need to be included in the diagram as branches stemming from the composite attribute

<u>The Multivalued Attribute</u>

These are attributes that can take on more than one value for a given instance of an entity.

In an ER diagram, these are denoted with a double oval to indicate their multivalued nature.

<u>Derived Attribute</u>

As the name implies, these attributes are derived or calculated from other data in the database.

In an ER diagram, this type of attribute is denoted by a dashed oval

<u>Keys</u>

candidate key - name given to an attribute that may be thought of as a unique identifier for an entity

 primary key - the name given to the candidate key that is chosen to be the unique identifier

A unique identifier in an ER diagram as the attribute named underlined
   ● when creating an ER diagram, it is not necessary to choose the primary key at this point

strong entities - entities that have at least one identified key
   ● *regular entities* (Chen)

weak entities - entities that depend on other entities for their existence
   ● may no have candidate keys
   ● depend on some other entity to furnish a unique identifier to give the entity a reason to be recorded
   ● denoted with a double box

4.6 - English Description of the Entity

Use a structured English grammar and substitute the appropriate information from the entity diagram.

NOTE: Simply check this section as a reference when writing the English phrases for our ER diagram.

4.7 - Mapping the Entity Diagram to a Relational Database

This section on mapping is concerned with placing data into relational tables that we will visualize as a physical organization of data.

General Mapping Rules:
1. Mapping strong entities - develop a table for each strong entity and make the indicated key of the strong entity the primary key of the table, if more than one candidate key is indicated in the ER diagram, choose one for the primary key.

2. Mapping atomic attributes - for entities with atomic attributes, map the entities to a table and form columns for each atomic attribute (form the columns)

3. Mapping composite attributes - for entities with composite attributes, map entities to a table and form columns of each elementary part of the composite attribute
   ● this simple means to break up the composite attribute and create individual rows for each of its components
   ● in the notation, the components are typically denoted using dot notation

4. Mapping multivalued attributes - form a separate table for the multivalued attribute, record a row for each value of the multivalued attribute together with the key from the original table; key of new table will be the concatenation of the multivalued attribute plus the key of the owner entity; remove multivalued attribute from table
   ● the example given in the text created a separate table that only contained the primary key from the original entity table and the multivalued attribute broken down such that the primary key relates to multiple values (bad wording)
   ● every row is unique therefore a candidate key in these types of tables can simply be the entire row (subset of all the attributes)