Lecture 1B (Jan. 10, 2018)

CS 410/510 - Databases

Logistics

- Monday: introduced class and topic
- Sign-up sheet up for whiteboard photography. Should actually work now!
- Homework 1 will be out by Monday

Any questions about class logistics?

Entities and Relationships (and Attributes)

Remember that we have 3 kinds of things we look for:

- **Entities** are the objects that we are describing
  - An **entity type** is a kind or type of entity
- **Attributes** are data about an entity
  - Name
  - Text
  - Birthday
- **Relationships** connect one entity to another
  - A sales rep *makes* a sale, which is *sold to* a customer
  - A **relationship type** is a type of relationship

In E-R diagrams, we are depicting entity types, relationship types, and attributes, which describe how the entities and relationships are stored.

Draw out a simple sales rep model

entity : entity type :: object : class

Keys

Each entity type should have a **primary key**: an attribute that uniquely identifies the entity.

There are two kinds of primary keys

- **Natural** primary keys are based on some attribute of the data that naturally occurs (e.g. name)
- **Synthetic** primary keys are made up

We often create synthetic primary keys even when natural keys are available

- What if we need to extend to additional data w/ colliding natural keys?
- What if natural key rules change?
- What if we misunderstand natural key?
- What if natural key ‘state’ is different from entity state?
    - Example: student IDs and SSNs

A good primary key is

- **Stable** - it will not change for the life of the entity
- **Unique -** if it is not unique, it cannot be used to identify the entity

Later on we will talk about **candidate keys -** sets of attributes that could function as a primary key.

We are not yet dealing with *foreign keys*. We don’t use them in E-R modeling.

Attribute Types

- **Multi-valued** attributes have more than one value (e.g. a list or a set)
- **Composite** attributes have attributes themselves
- **Derived** attributes can be computed from other attributes (or related entities)

When should you have a multi-valued or composite attribute and when should you have an entity?

- If it is meaningful to talk about a particular element of a MV attribute, or a particular composite attribute value, then it should be an entity.
- Otherwise: whatever helps you communicate the intent of the model most clearly.
- Attributes may evolve into entities through *model evolution*

More on Relationships

Relationships have properties:

- **Degree** - how many entity types participate? 2 is common, but we can have more (e.g. 3)
- May have **role names**
- **Cardinality ratio**: how many entities of each type participate?
- They can also have **attributes**

Example: research paper authorship

More Diagramming

Start diagramming tweets

Syntax Summary

See Figure 3.14 (page 83) in the Elmasri and Navathe text.

The Three-Schema Model

There are three types of database schemas we consider:

- **User** schemas (or **conceptual** schemas) - how users see the data
    - More than one of these! Different users have different needs and different views of the world
- **Logical** or **internal** schema - how we logically model the data
- **Physical** schema - how the data is stored in the database

Physical has 2-3 different layers itself, in principle:

1. Mapping to our database model (e.g. relational)
2. Mapping to our specific database management system (e.g. MS-SQL Server or PostgreSQL)
3. Mapping to sequences of bytes on disk (DBMS does this for us)

DBMS does (3), we need to do a combination of 1 and 2.

Model Evolution

Models are not static. They evolve and change, as user needs change, as we better understand user needs, etc.

Modeling and development are iterative processes. Agile!

Examples

- Twitter
- Research papers
- Sales

Additional Concepts

Probably next week!

- Weak entities
- Totality

- The Relational Model