CSCI 403
Database Management

13 – Database Modeling with Entity-Relationship Diagrams

3 Levels of Database Design

- Conceptual (this lecture)
  - Understand data entities & relationships between them
  - Communication with stakeholders at this level
  - ERD (Entity-Relationship Diagrams)
- Logical (next lecture)
  - Mapping from design to an actual DBMS
- Physical (not covered)
  - Bare-metal stuff – usually for DBAs (database administrators) only
  - Where files live, network architectures, etc.

Entity-Relationship Diagram (ERD)

- Peter Chen, 1976
- Visual language for database modeling/design
- Two major components:
  - Entities – "nouns"
    - Things or objects with independent existence
    - E.g., persons, products, companies, courses
  - Relationships – "verbs"
    - How entities interact or refer to each other
    - E.g.
      - A person **supervises** a department
      - An instructor **teaches** a course
      - A customer **purchases** a product

Mines Courses ERD

- Examples will be based on a "Mines courses database"
  - Similar to, but not the same as, what is in the CSCI 403 database
  - We'll actually develop a "complete" model as a class
- This will let us examine all elements of an ERD

Entities

Visualized as a rectangle; name of entity goes in rectangle:

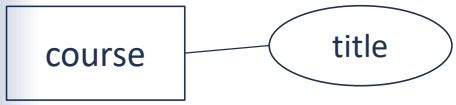
Attributes

Properties of an entity, shown as ovals.

A course has a "title" property, for instance:

Attributes – Entities

Attributes are attached to their entities with straight lines:



Key Attributes

Attributes may be designated as *keys*.

A key in ERD is basically like a primary key – it uniquely identifies an instance of an entity.

Key attributes have their names underlined:

Composite Attributes

Composite attributes represent some property of a database which is not atomic, but composed of parts.

Example: address may be composed of street address, city, state, zip.

Composite attributes are represented with additional ovals connected to the composite:

Why Composites?

- Can just add individual attributes to entity
- However, cannot define composite keys that way – must use a composite attribute.

Multivalued Attributes

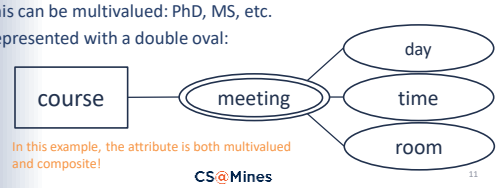
Multivalued attributes represent something that can have multiple entries.

Example:

An instructor might have a "degree" attribute, representing their academic credentials.

This can be multivalued: PhD, MS, etc.

Represented with a double oval:



In this example, the attribute is both multivalued and composite!

Derived Attributes

A value which can be derived from other attributes – i.e., it doesn't need to be stored.

Example: a person's age can be derived from their birthdate.

Derived attributes have a dashed outline:

I don't have an example of this for our Mines courses database, but maybe we'll find one together.

Still To Come: Weak Entities

Another entity type: we'll revisit in a bit.

First, we need to understand relationships!

Relationships

Drawn as a diamond:

Relationships – Entities

Must attach to two or more entities:

Cardinality Ratios

Decorate connectors with cardinality ratios: how many instances of each entity participates in the relationship with an instance on the other side?

Choices:

1 : 1 - "one-to-one"

1 : N – "one-to-many"

N : N – "many-to-many"

Participation

Does the existence of an instance of an entity depend on their being an instance on the other side?

If so, then that entity has *total* participation in the relationship (represented by a double line):

Otherwise, participation is *partial*.

Participation vs Cardinality

- Cardinality identifies a maximum
- Participation identifies a minimum
- A modern extension:
  - Cardinality ratios specify min & max, e.g.:
    - 0..1 : N
    - 1..1 : 1..4
    - Etc.

Weak Entity

- A "weak" entity depends for its existence and identity on relationship with some other entity
  - Relationship is called the *identifying* relationship
  - Other entity is called the *owning* entity
- A weak entity has only a *partial* key
  - Does not by itself uniquely identify instances of the entity
  - Makes a complete key only with addition of key from owning entity

Weak Entity Visuals

- Weak entities – double-bordered rectangle
- Identifying relationships – double-bordered diamond
- Partial key – dashed underline

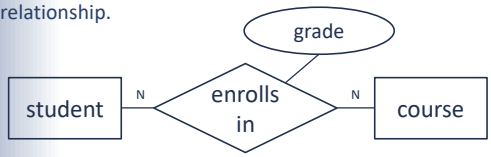
Contrived Weak Entity Example

Relationship Attributes

Relationships may also have attributes.

These represent values which don't properly belong to either entity, but to the combination.

These are represented just as ovals attached to the relationship.



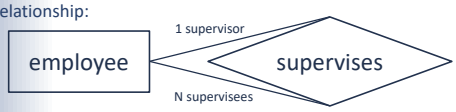
Recursive Relationships

A relationship can connect an entity to itself.

The classic example of this is an **employee** entity, with a **supervises** relationship.

Since employees supervise other employees, the relationship is recursive.

You can annotate the connecting lines to make clear the relationship:



N-ary Relationships

Relationships can connect more than two entities, where appropriate (this is rare).

A classic example connects the entities **vendor**, **part**, and **project**.

The relationship models the idea that various projects use various parts from various vendors.

Your Turn!

As a class:

- Brainstorm entities for a hypothetical Mines courses database (think Trailhead, only better)
- Brainstorm relationships
- Iteratively develop a model

Up Next

- Next lecture:
  ERD-to-relational mapping