

BUS 362

Assignment 06 - Entity Relationship Diagrams

Ashton Charbonneau

2017-06-22

# Entity Relationship Diagrams

# Purpose

- Visual representation of a database
- How is data stored? How are individual pieces of data related to other pieces? How do we store complex data?
- Unrelated to processes or how data is used, information about the ERD cannot be derived from DFDs or use cases

# Principles

- Never repeat data that can/will change
- Make your database as close to reality as you can
- Flexibility is more important than space/performance

# Components

- Entities
- Attributes
- Relationships
  - Modality
  - Cardinality
  - Label (verb)

# Entities

- Think of each entity as a spreadsheet containing data (a table/relation)
  - Each row is considered an instance of that entity (a tuple)
  - Each column is considered an attribute for that entity
- Many businesses use multiple spreadsheets to track different things about their business - even more businesses use databases to track things in the same manner
- You only need entities that describe information you wish to store

# Attributes

- Attributes are a list of what an instance of that entity can contain
- Not all entities need to have every attribute filled in, they can be left null
- Each entity will have a primary key (unique and non-null)
- Each attribute has a domain that restricts the possible values

# Examples

- Universities track students, classes, professors
- Airlines track flights, customers, seats, planes, bags, etc
- Libraries track customers, librarians, books, instances of books
- Facebook tracks users, friendships, albums, events, games (all of which have their own databases), groups, wall posts, comments, statuses, etc
- ERDs in real businesses are massive - on the small side, they can be on the order of hundreds of entities



# Primary Keys

- A value that is unique to that row (tuple)
- No future or past instance of the entity can have the same primary key
- Used to identify a single row (tuple) in a table (relation)
- The easiest way to ensure that your primary key is unique is to generate a new one every time a new row is added
- Can be composed of multiple values
- Only one primary key per attribute

# Foreign Keys

- If an entity is on the many side of a 1:M relationship, you need to track which other entity it is related to
- List the primary key of another entity as an attribute
- If it is a 1:1 relationship, you can choose which side has the foreign key

# Relationships

- Labeled as a verb in each direction
- 1:1, 1:M, M:N
- *Entity1* has/is of *Entity2*
- *Student* takes/participates in *Class*
- Can be between any two entities

# Cardinality

- Called the cardinality constraint
- 1 or many
- Whether an entity can be associated with only one entity at a time, or multiple
- Example
  - Each student can take many classes at a time
  - Each student can only have one library card at a time

# Modality

- Also called a participation constraint
- Defines the minimum number of relations that are possible with the connected entity
- Describes whether or not the relationship is optional
- 1 or 0
- Example
  - Some students are not associated with any classes, some classes are not associated with any students
  - All students have a library account, all library accounts have a student

# Resolving M:N Relationships

- Create an intermediate entity
- Not always obvious
- *Student* >-< *Class*
- *Student* -< *Class Enrollment* >- *Class*

# Higher Level Courses

- Querying databases
- Entities can be related to themselves (recursive relationships, eg. hierarchical relationships)
- Entities can have multiple relationships between them (eg. department and boss, team and team captain)
- More abstract diagrams
- Aggregate entities, multiple attribute types, candidate keys/superkeys
- Relational algebra/calculus

# Textbook Pages

- 5th Edition: Chapter 6 - Data Modeling (p224 - 254)
- 6th Edition: Chapter 6 - Data Modeling (p187 - 216)



## Examples

# Examples

1. Model a system for a university that tracks classes. It should keep track of the teachers, the students, the teaching assistants, and student classes over time (including grades)
2. Model a system for an airline to keep track of flights. It should include customers, pilots, and some basic flight details.

# Assignment

# Instructions

- Using VISIO, draw the ERD. You can either try to build the diagram directly or use your use cases and a DFD to help you identify appropriate entities and attributes. Indicate cardinality and modality using crow's foot notation. Label the relationships and list the attributes, indicating the identifiers.
- Keep your diagram simple, try to avoid complicated assumptions.
- List non-obvious assumptions on the ERD