CSE 403
Software Engineering
Spring 2023

#10: Data modelling
Logistics

WEEK 4

04/17  L: Data modeling
04/18  T: DUE: GPS!!!
04/19  L: Architecture
       Design & Architecture (DnA)
04/20  P:
04/21  L: Design
From Requirements to System Design
From Requirements to System Design

Client

ideas, target users, desired behavior

questions, suggestions, diagrams, feasibility

Developer
From Requirements to System Design

Client

ideas, target users, desired behavior

questions, suggestions, diagrams, feasibility

Developer

Requirements
From Requirements to System Design

Client

ideas, target users, desired behavior

questions, suggestions, diagrams, feasibility

Developer

Requirements

Data Model

SW Architecture and Design

What

How
Life-cycle stages

Virtually all SDLC models have the following stages
- Requirements
- Design
- Implementation
- Testing
- Maintenance

Do you remember this?
From Requirements to System Design

Client

- ideas, target users, desired behavior
- questions, suggestions, diagrams, feasibility

Developer

Today

Requirements

Data Model

SW Architecture and Design
Data Modelling
Goals for today

● How to model data?
  ○ Identify Entities
  ○ Identify Attributes
  ○ Identify Relationships
  ○ Assign Keys
    ○ (Normalization to reduce redundancy)
    ○ (Denormalization to improve performance)

● Common “language” for data modelling
  ○ ER (Entity-Relationship) diagrams
  ○ Just one out of many possibilities (diagrams, tables, text)

● Develop a data model for a course-registration system
ER diagrams: overview

- An Entity Relationship (ER) diagram is a **graphical representation** of a **data model**.
- It shows the **relationship** between **entities** (e.g., people, objects, events, or concepts) within a system.
- It can be mapped to a **relational** (database) **schema**.
ER diagrams: graphical syntax

- An entity $E$
ER diagrams: graphical syntax

- An entity $E$

- An attribute $A$ of entity $E$
ER diagrams: graphical syntax

- An entity $E$

- An attribute $A$ of entity $E$

- A relationship $R$ between two entities $E1$ and $E2$
ER diagrams: graphical syntax

- An entity $E$

- An attribute $A$ of entity $E$

- A relationship $R$ between two entities $E_1$ and $E_2$

- An attribute $B$ of relationship $R$
ER diagrams: rules

- An interconnecting line is only allowed between:
  - a box and a diamond,
  - a box and an oval,
  - a diamond and a oval.
- An oval must have exactly one connecting line.
- Names of boxes must be unique in the diagram.
- Names of ovals must be unique per box/diamond.
ER diagrams: rules

- An interconnecting line is only allowed between:
  - a box and a diamond,
  - a box and an oval,
  - a diamond and a oval.

- An oval must have exactly one connecting line.

- Names of boxes must be unique in the diagram.

- Names of ovals must be unique per box/diamond.

Questions?
A first example

Let’s model a simple course registration system:

- Students
- Instructors
- Courses
A first example: identify entities

What attributes should we add?
A first example: identify attributes

- **Student**
  - NetID
  - Name

- **Instructor**
  - ID
  - Name

- **Course**
  - CourseID
  - Title
A first example: identify attributes

What relationships should we add?
A first example: identify relationships

Student
- NetID
- Name

Instructor
- ID
- Name

Course
- CourseID
- Title

Student takes Course

Instructor teaches Course
ER diagrams: keys and cardinalities

- A **key** is an (underlined) attribute, or a set of attributes, which uniquely identifies an entity.

[Diagram of ER model:
- **Student** entity with attributes:
  - NetID
  - FirstName
  - LastName
- **Course** entity with attributes:
  - CourseID
  - Title
- Relationship labeled **takes** between Student and Course entities]
ER diagrams: keys and cardinalities

- A key is an (underlined) attribute, or a set of attributes, which uniquely identifies an entity.
- A key can be artificial or natural.
A key is an (underlined) attribute, or a set of attributes, which uniquely identifies an entity.

- A key can be artificial or natural.
- The cardinalities define the kind of relationship (one-to-one, one-to-many, or many-to-many).
- There are different notations for cardinalities. For example:
  - $1 = (1,1)$
  - $c = (0,1)$
  - $m = (1,*)$
  - $mc = (0,*)$

ER diagrams: keys and cardinalities
ER diagrams: weak entities

- A **weak entity** can’t exist on its own (if a building is torn down, its rooms disappear).
ER diagrams: weak entities

- A weak entity can’t exist on its own (if a building is torn down, its rooms disappear).
- A weak entity is only uniquely identifiable in reference to another entity.
ER diagrams: self references and roles

- A self reference is usually explicitly annotated with roles to clarify the meaning of the self-referencing relationship.

Think about (but never draw) the following:

- A self reference is usually explicitly annotated with roles to clarify the meaning of the self-referencing relationship.

Think about (but never draw) the following:
Putting it all together

Let’s **augment** our **model** of a course registration system:

- Prerequisites
- Assignments
- Points/grades

Instructions

https://tinyurl.com/cse403-ER
Putting it all together

Student

NetID
Name

Course

CourseID
Title

teaches

Instructor

ID
Name
Putting it all together

- **Student**
  - NetID
  - Name
  - Grade
  - Takes (0,*

- **Instructor**
  - ID
  - Name
  - Teaches

- **Course**
  - CourseID
  - Title
  - Has
  - Prereq
  - Requires

- **Assignment**
  - MaxPoints
  - A_Id
  - Submits

- **Relationships**
  - Takes (0,*)
  - Teaches 1
  - Has
  - Prereq for
  - Prereq

- **Attributes**
  - Points
  - (0,*)
Additional material, not discussed in class
ER diagrams: generalization

- An is_a relationship represents a generalization relationship between two entities.
ER diagrams: generalization

- An is_a relationship represents a generalization relationship between two entities.
- Attributes (including keys) are “inherited”.

![ER Diagram]

- Person
  - SSN
  - Name
  - is_a
  - Student
    - GPA
  - Instructor
    - SRTI
ER diagrams: generalization

- An is_a relationship represents a generalization relationship between two entities.
- Attributes (including keys) are “inherited”.
- Additional attributes can be defined.