Database Design and the E-R Model

CISC637, Lecture #5

Ben Carterette

Database Design

- Why are we using a database?
  - We have data (probably a lot of it)
  - We want to be able to maintain it easily and quickly
  - We want to be able to query it and get answers fast
  - We need to support many simultaneous users
- How do we design a database to support **efficient** and **correct** updates and queries?

Database Design Process

- Start with enterprise requirements
  - → Translate to a conceptual model
  - → Translate to a logical model
  - → Translate to a physical model

- **Iterate** each step to ensure that everything that *needs to be* captured and that *can be* captured *actually is* captured

Database Design Process

- Start with enterprise requirements
  - →Translate to a conceptual model
    - Entity-Relationship model (E-R model) – subject of this week
  - →Translate to a logical model
    - Relational model – subject of past two weeks
  - →Translate to a physical model
    - Implement in a database

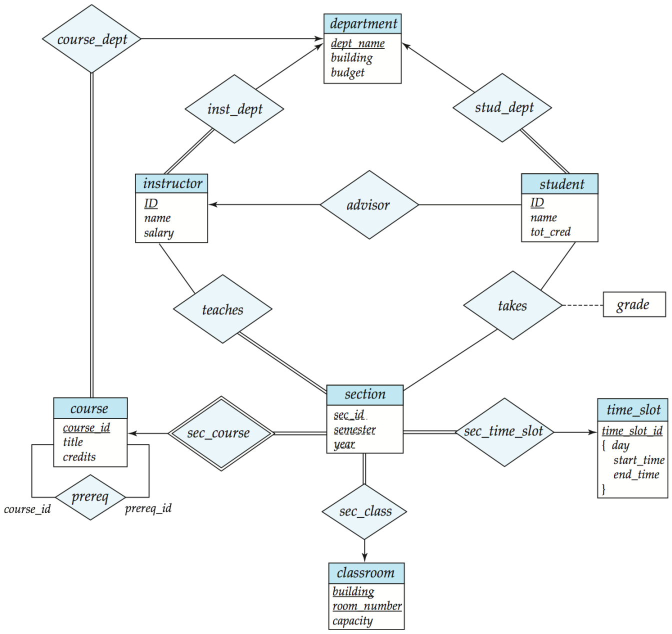
E-R Model

- **Entity-relationship model**
  - A data model for representing *entities* and *relationships* between them
- **Entities** represent objects in the enterprise
  - Each entity has **attributes** that describe it
  - An **entity set** is a set of entities with the same set of attributes

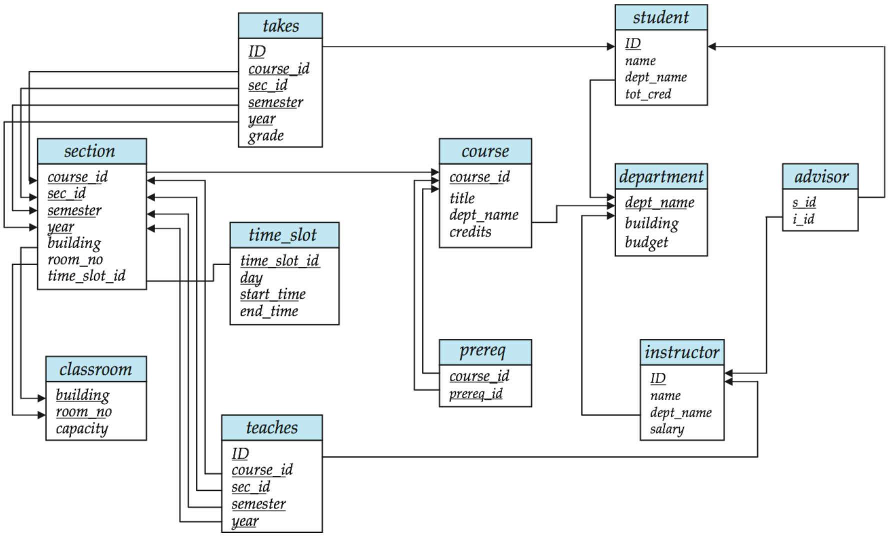
- **Relationships** represent associations between two or more entities
  - Relationships can also have attributes
  - A **relationship set** is a set of pairs of entities with the same association

University Requirements

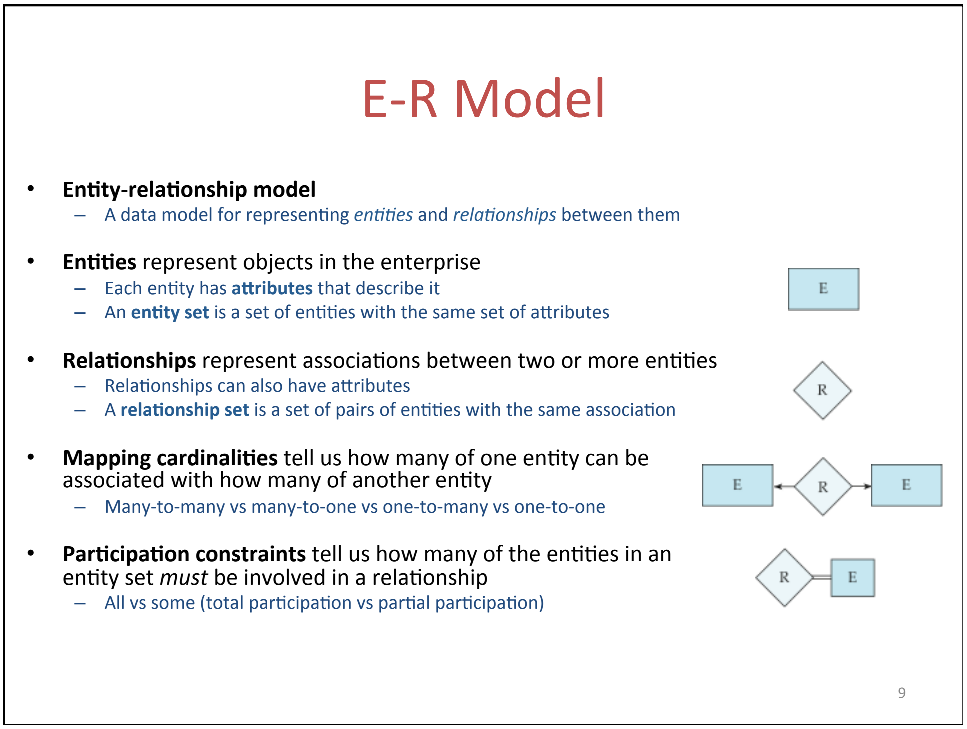
- Requirements regarding people in the university:
  - instructors are identified by an ID num, and we need to store their name and salary
    - every instructor must also be associated with exactly one department
  - students are identified by an ID num, and we need to store their name and total credits
    - every student must also be associated with exactly one department
  - a student can have at most one instructor as advisor
    - instructors can advise any number of students
- Requirements regarding courses and course scheduling:
  - courses have IDs, titles, and number of credits
    - every course must be associated with exactly one department
  - courses are scheduled into sections
    - a section of a course is identified by a section number, semester, and year
    - a section must be associated with exactly one course, though a course can have multiple sections
    - each section takes place in exactly one classroom (which has a certain capacity), and at exactly one of a pre-determined set of time slots
  - students take a section of a course for a grade
  - each section is taught by one or more instructors
  - some courses have one or more prerequisite courses
- Requirements regarding departments:
  - departments are identified by a name, and we need to store their home building and budget



Copyright © Ben Carterette 7



Copyright © Ben Carterette 8



E-R Model

- **Entity-relationship model**
  - A data model for representing *entities* and *relationships* between them
- **Entities** represent objects in the enterprise
  - Each entity has **attributes** that describe it
  - An **entity set** is a set of entities with the same set of attributes
- **Relationships** represent associations between two or more entities
  - Relationships can also have attributes
  - A **relationship set** is a set of pairs of entities with the same association
- **Mapping cardinalities** tell us how many of one entity can be associated with how many of another entity
  - Many-to-many vs many-to-one vs one-to-many vs one-to-one
- **Participation constraints** tell us how many of the entities in an entity set *must* be involved in a relationship
  - All vs some (total participation vs partial participation)