

# LBSC 690: Information Technology

## Homework Solution 05

### Databases

William Webber  
CIS, University of Maryland

Spring semester, 2012

## Requirements: entities

- ▶ The digital drawings will be uploaded to a server external to the database, and referred to in the database by URL.
- ▶ For each drawing, the database must record the title of drawing, the date it was drawn, which student drew it, and what class they were in when they drew it.
- ▶ For each student, the database must record the student's given (first) name, family (last) name, and a list of all the classes the student is in.
- ▶ For each class, the database must record the name of the class (e.g. "Art", "Geography"), the year level of the class, and the name of the class teacher.

## Requirements: relations

- ▶ A drawing is drawn by only one student, but a student can draw many drawings.
- ▶ A student can belong to many classes, and each class can have many students.
- ▶ A drawing can be drawn in only one class, but each class can have many drawings made in it.

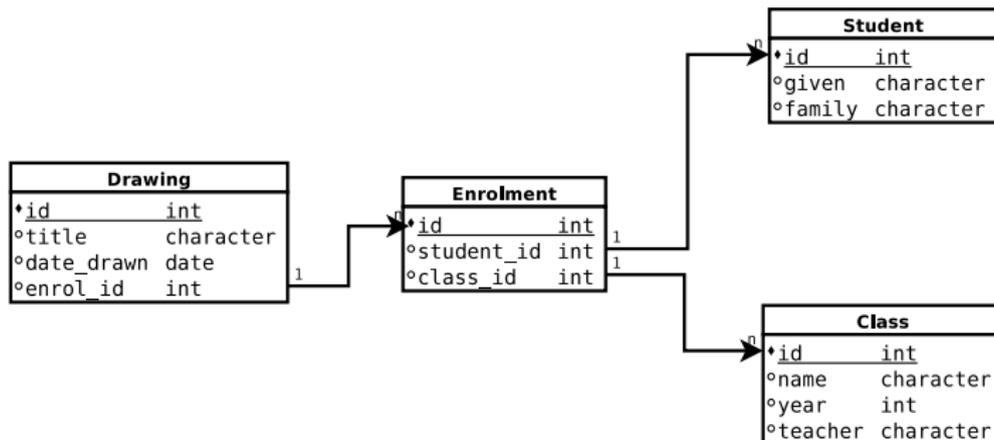
# Issues: student to class

- ▶ We're required to record all classes a student is in
  - ▶ A standalone drawing database is not necessarily a good place to record this
  - ▶ But it depends on how the application will work
- ▶ Students are in many classes; classes have students; so this is a **many-to-many** relation.

## Resolution: link entity

- ▶ Relational model can't directly store **many-to-many**
- ▶ Instead, break out separate entity that holds the relation
- ▶ Here, natural name for relation representing “a student taking a class” is `Enrolment`

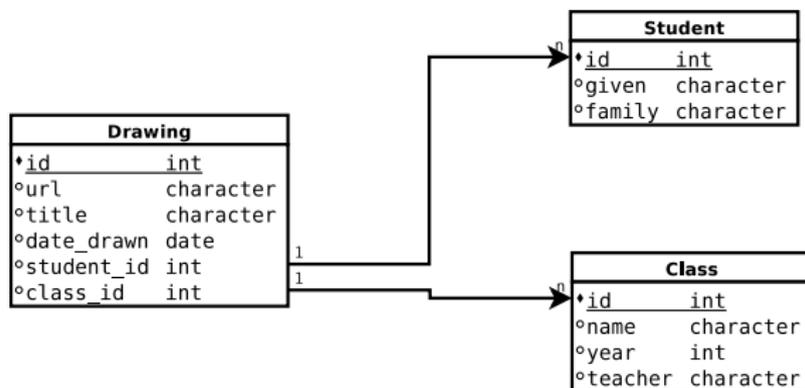
# Solution



# Criticism

- ▶ Requirement that we record student enrolments in drawing database is questionable:
  - Not directly necessary to store required information
  - Replicates data that is probably held elsewhere
  - + Enforces a real-world constraint
  - + Could be useful in interface (provide drop-down list of enrolments for a student)
- ▶ Alternate model would exclude the `Enrolment` entity
- ▶ Store information directly in `Drawing`

## Alternate solution



- ▶ **Note** that this solution violates the formal requirement that the database stores all enrolments

# External data sources

- ▶ Relation with external databases a recurrent problem with database design.
- ▶ Putting everything in a single database, while attractive, often impractical:
  - ▶ Legacy systems
  - ▶ Proprietary systems
  - ▶ Organizational boundaries
  - ▶ Over-integration leading to excessive complexity
- ▶ Approaches:
  - ▶ Autonomy (duplicate information without direct updating)
  - ▶ Updates (import data on regular occasions, e.g. nightly, weekly)
  - ▶ Live link (provide interface for one application to query database of another)