IS5102
Database Management Systems

Lecture 3:   E-R Diagrams

Alexander Konovalov
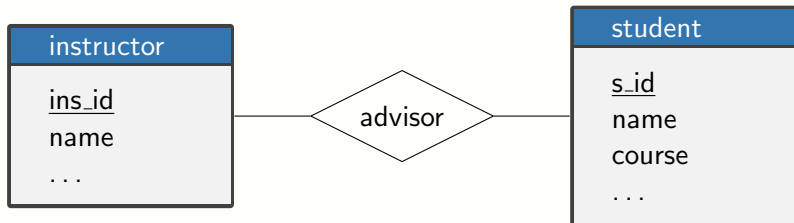
alexander.konovalov@st-andrews.ac.uk

(with thanks to Susmit Sarkar)

2021

► Entity Relationship Modeling

► Entities, Relationships, Attributes

► Drawing E-R models

- ▶ Rectangles represent entity sets.
- ▶ Diamonds represent relationship sets.
- ▶ Attributes listed inside entity rectangle
- ▶ Underline indicates primary key attributes

one-to-many relationship between an instructor and a student

▶ an instructor is associated with several (including 0) students via advisor
▶ a student is associated with at most one instructor via advisor
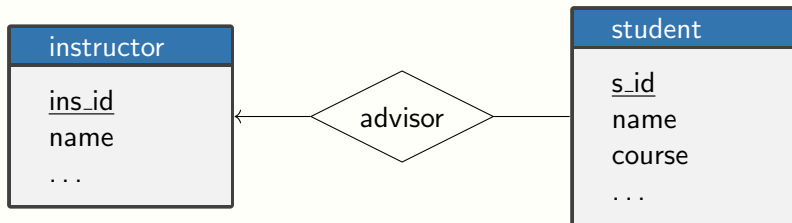
- ► Consolidating E-R Models

- ► Refinements to Weak Entities

- ► Specialisation, Generalisation

- ► Some common pitfalls

▶ An entity set with a primary key is called a **strong entity set**

▶ An entity set without a primary key is called a **weak entity set**

▶ For example, consider entity set of offices in company's buildings: room 0.11 might exist in Jack Cole Building and in the School of Mathematics and Statistics

▶ The existence of a weak entity set depends on the existence of a **identifying entity set** (also called an **owner entity set**)

  ▶ It must relate to the identifying entity set via a **total**, **many-to-one** relationship set from the identifying to the weak entity set
  ▶ This relationship set is called an **identifying relationship** and is depicted using a **double diamond**

▶ In our example, the identifying identity set is `building`

▶ The **discriminator** (or **partial key**) of a weak entity set is the set of attributes that distinguishes among all the entities of a weak entity set that depend on one particular strong entity

▶ The **primary key** of a weak entity set is formed by

the primary key of the strong entity set on which the weak entity set is existence dependent,

plus the weak entity set's discriminator

- ▶ We put the **identifying relationship** of a weak entity in a double diamond.
- ▶ We underline the **discriminator** of a weak entity set with a dashed line.
- ▶ Primary key for office — (`building_id, floor, room_no`).
- ▶ Note the double line (total participation) and the arrow (one-to-many).
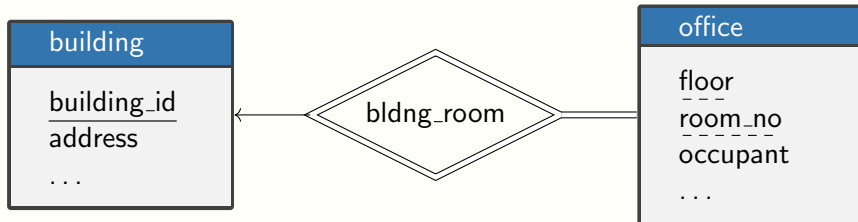
▶ Note: the primary key of the strong entity set is not explicitly stored with the weak entity set, since it is implicit in the identifying relationship

▶ If `building_id` were explicitly stored, office could be made a strong entity ...

... but then the relationship between office and building would be duplicated by an implicit relationship defined by the attribute `building_id` common to building and office

Use of entity sets vs multivalued composite attributes

- ▶ A week entity set could be more appropriate as an attribute if it participates only in its identifying relationship, and has a few attributes
- ▶ A phone type (office, home, mobile) and number could be a multivalued attribute for a person
- ▶ Use of phone as an entity allows extra information about phone numbers, other relationships e.g. between phone numbers and offices, etc.

Use of entity sets vs relationship sets

- ▶ Possible guideline is to designate a relationship set to describe an **action** that occurs between entities

► There are some relationships that are naturally non-binary

   Example: `proj_guide`

► Sometimes an n-ary relationship set shows more clearly that several entities participate in a single relationship.

► In general, any non-binary relationship can be represented using binary relationships by creating an artificial entity set.

► Placement of relationship attributes e.g., attribute date as attribute of advisor or as attribute of student

In general, any non-binary relationship can be represented using binary relationships by creating an artificial entity set.

- ▶ Replace R between entity sets A, B and C by an entity set E, and three relationship sets:
    1. RA, relating E and A
    2. RB, relating E and B
    3. RC, relating E and C
- ▶ Create a special identifying attribute for E
- ▶ Add any attributes of R to E
- ▶ For each relationship $(a_i, b_i, c_i) \in R$, create
    1. a new entity e_i in the entity set E
    2. add (e_i, a_i) to RA
    3. add (e_i, b_i) to RB
    4. add (e_i, c_i) to RC

- Chapter 8, Database Design
- Chapter 7, Database System Concepts