SWEN304: Entity-Relationship Data Model

Dr. Dionysios Athanasopoulos

Lecturer
dionysios.athanasopoulos@vuw.ac.nz
Office: EA111, Easterfield building, Kelburn Campus

Entity-Relationship Data Model

To design a database application,

- we firstly perform the conceptual modeling of the application.
- a popular high-level conceptual model is the **Entity-Relationship** (ER) model
  - it includes a set of modeling concepts and diagrammatic notation (ER diagrams)
  - the main concepts are:
    - entity (type)
    - attribute
    - relationship (type)
    - cardinality
    - constraints
- ER has been evolved in the **Enhanced Entity-Relationship** (EER) model.

Entity - Attribute

Entity

- An entity is the basic object of an ER model;
- an entity is a thing in the real world with an independent existence
- an entity may be an object with
  - physical existence (e.g. employee, car)
  - conceptual existence (e.g. company, university)

Attribute

- Each entity has attributes;
- each attribute is a property of an entity (e.g. employee's name, age, job).

Attribute Categories and Entity Types

An attribute can be

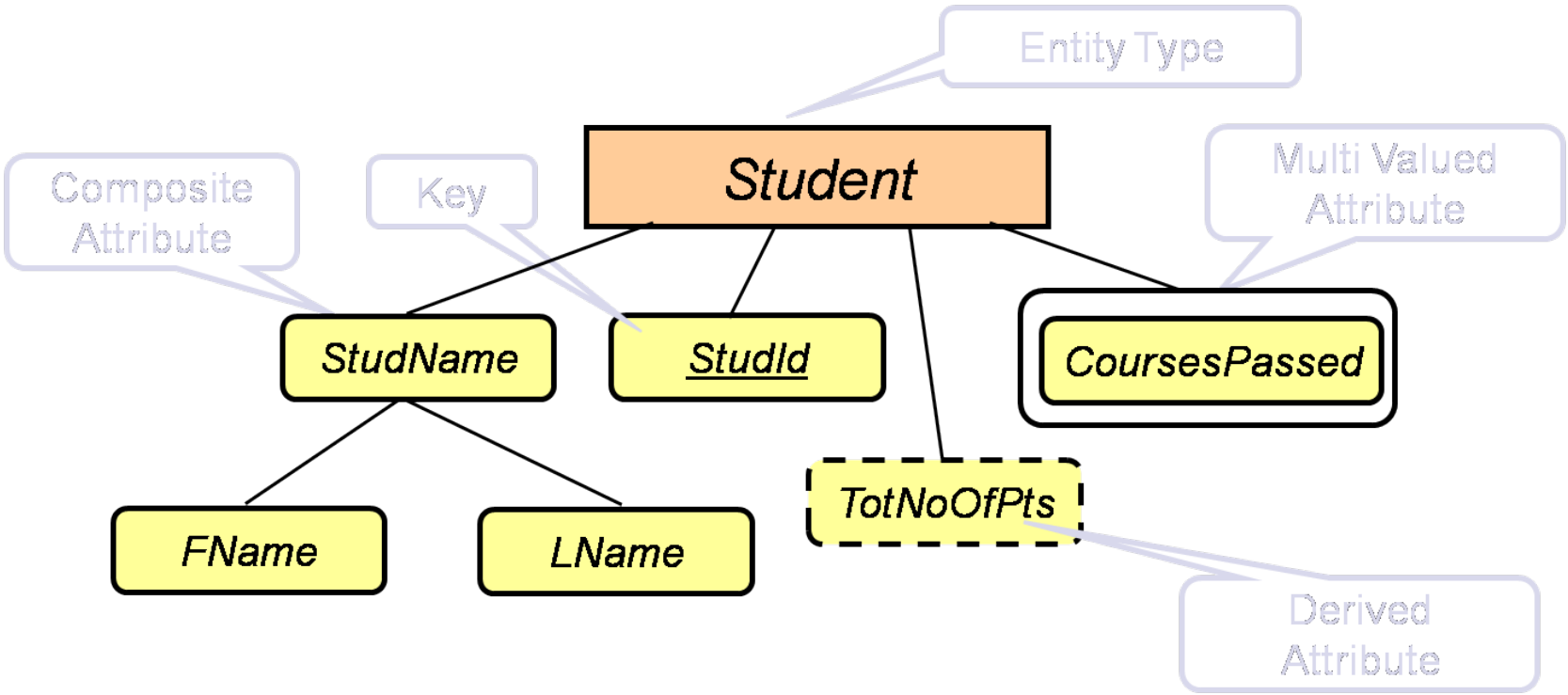
- simple (e.g. street name)
- composite (e.g. address)
- single-valued (e.g. student id)
- multi-valued (e.g. enrolled courses) - **double oval**
- derived (e.g. total number of points of a student) - **dotted oval**
- complex, i.e. composite and multi-valued
- key (e.g. student id) - **underlined**.

An attribute is represented by an **oval**.

Entities with the same attributes are grouped into an **entity type**;

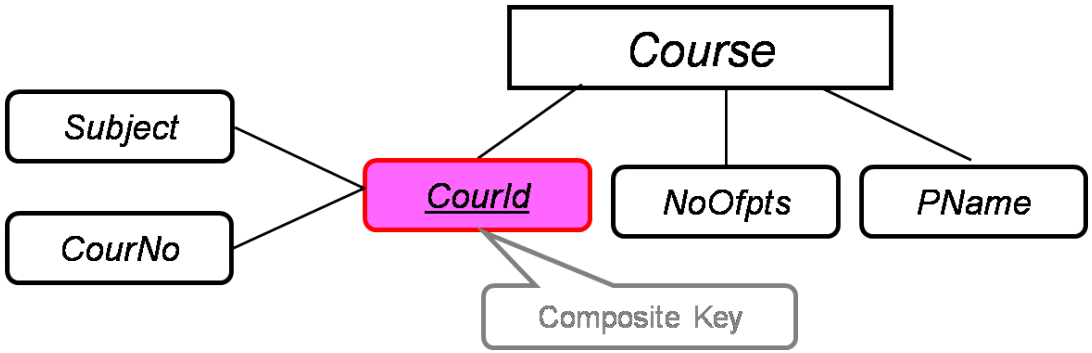
- such entities are called instances of a given entity type;
- an entity type is represented by a **rectangle**.

Graphical Representation of ER Diagram



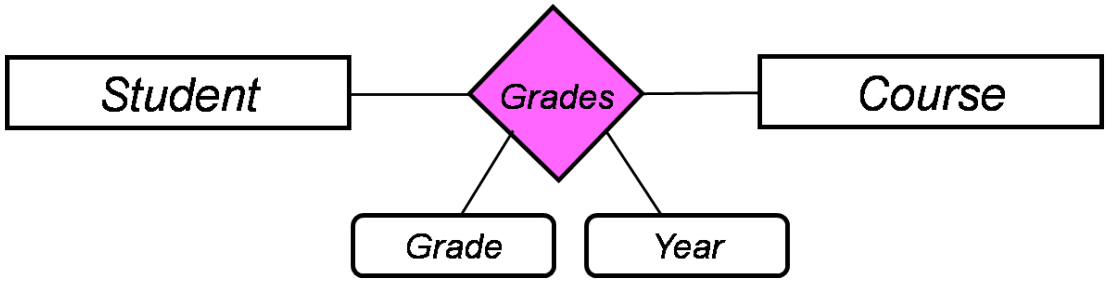
Key of Entity Type

A key may be composite.

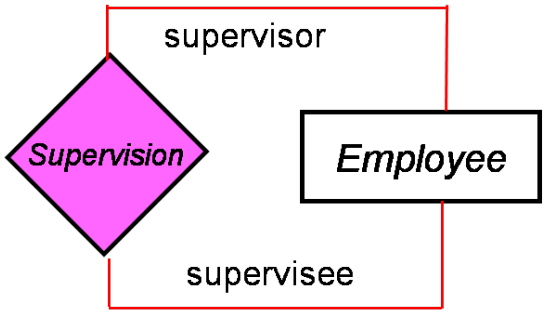


Entity Relationship

- A relationship is an **association** between two or more entities.
- A relationship is represented by a **diamond box** connecting associated entity types.
- A relationship may have its own attributes.



- An entity may participate in a relationship type with distinct **roles** (recursive relationship).

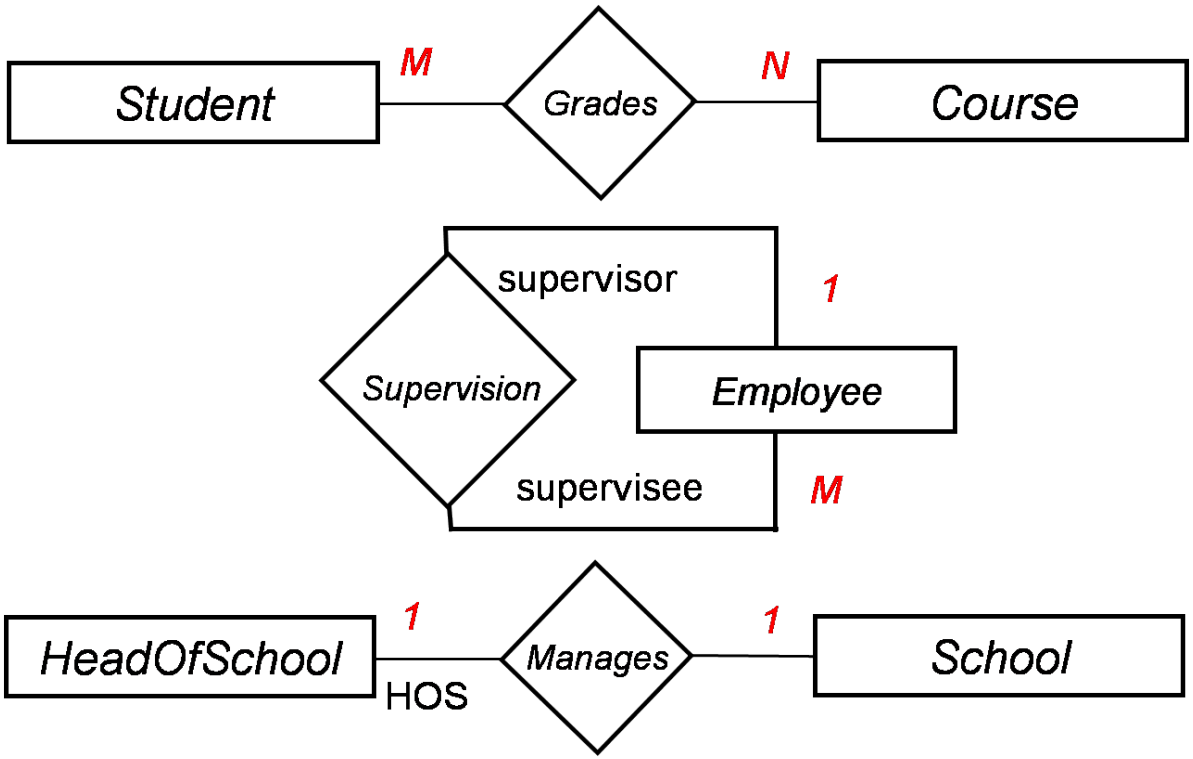


Constraints on Relationship Types

There are two constraints on relationship types (structural constraints):

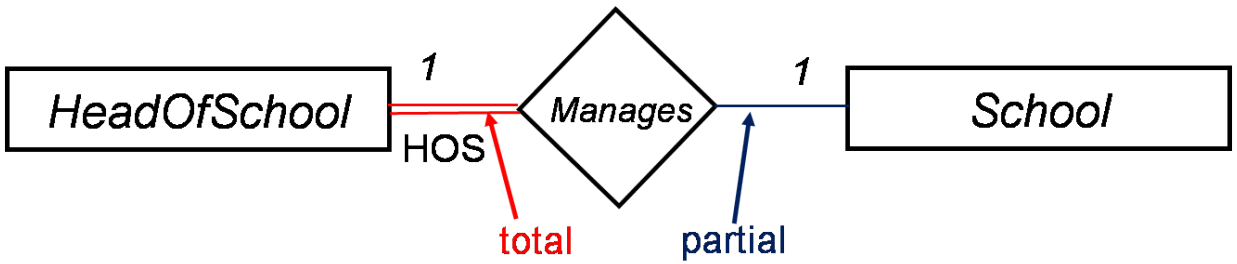
1. **cardinality** ratio
   - M : N (many to many)
   - M : 1 (many to one) or 1 : M (one to many)
   - 1 : 1 (one to one)
2. **participation** constraint.

Cardinality Ratio: Example



Participation Constraint

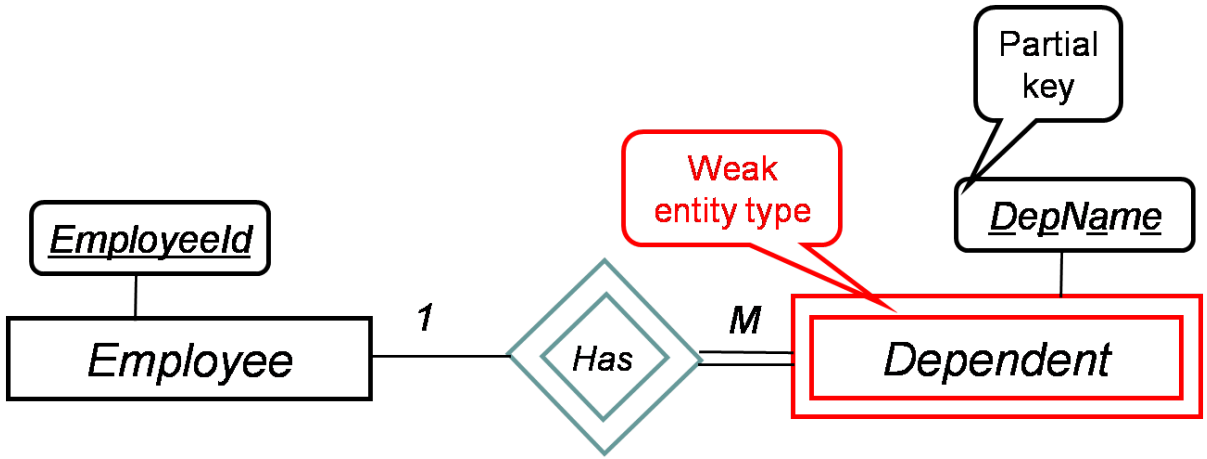
- The participation constraint specifies whether the existence of an entity depends on its association with another entity.
- A participation constraint can be
  - **total** or existence dependency
    - every instance of an entity is related to at least one instance of another entity
  - **partial**.



Weak Entity Types

Weak entities **do not have their own key attributes**.

- weak entities are related to other strong/owner entity types in combination with one of the attributes of the former entity;
- weak entities are denoted by double rectangles and their relationships with double diamond boxes;
  - these attributes form the **partial key** - **dotted underline**
- a weak entity has a **total** participation constraint
- the cardinality of a weak relationship is **1 : M**.



N-ary Relationships

In N-ary relationships,

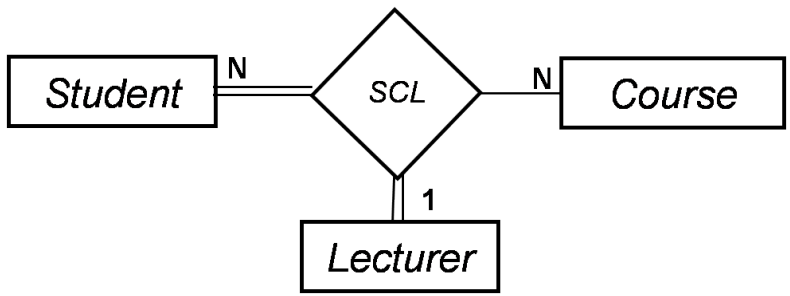
- multiple entities (N > 2) are related to each other;
- we consider associations between all the possible entity pairs.

For instance, suppose that

- each `Course` is taught by only one `Lecturer`
- each `Lecturer` teaches at least one `Course`
- each `Student` has enrolled in at least one `Course`,

then

- (`Student`, `Course`) is associated to exactly one `Lecturer`
- (`Lecturer`, `Course`) may be associated with multiple students
- (`Student`, `Lecturer`) is associated to at least one `Course`.



ER Exercise

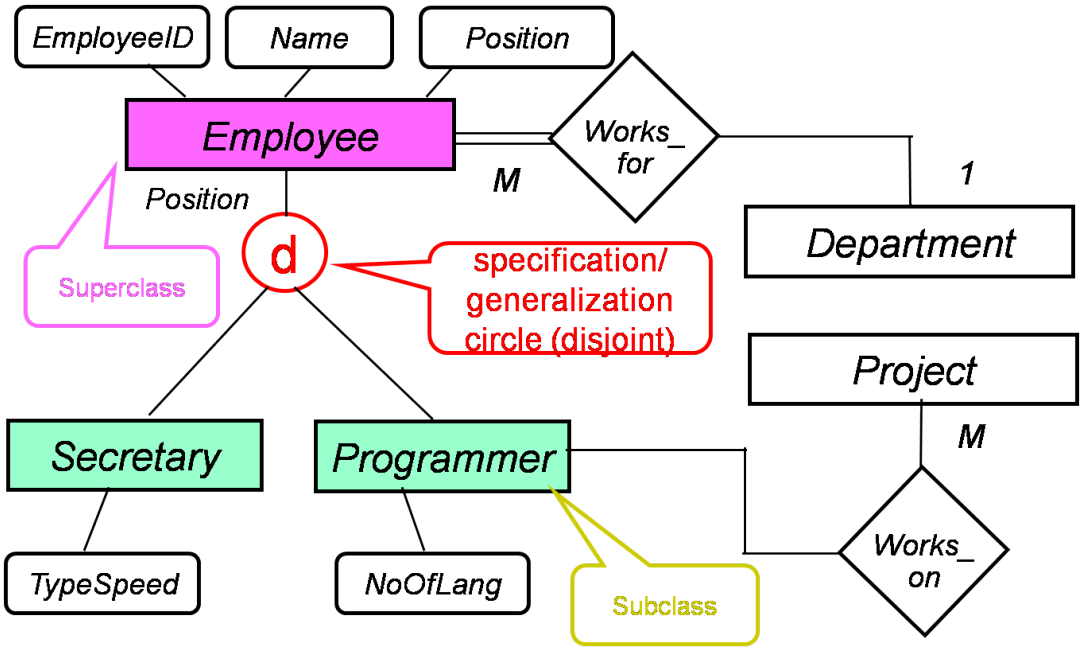
Draw an ER diagram of a database schema that meets best the following requirements.
The requirements concern a database that stores the information about students, courses, lecturers:

- each course is characterized by its id, name, and points
- some courses are divided into groups, characterized by a number
- groups with the same number may appear in many courses
- each course group is taught in a room
- a lecturer is characterized by its id and name
- each lecturer teaches only one course group
- a course group may be taught by multiple lecturers.

Enhanced Entity-Relationship Model (EER)

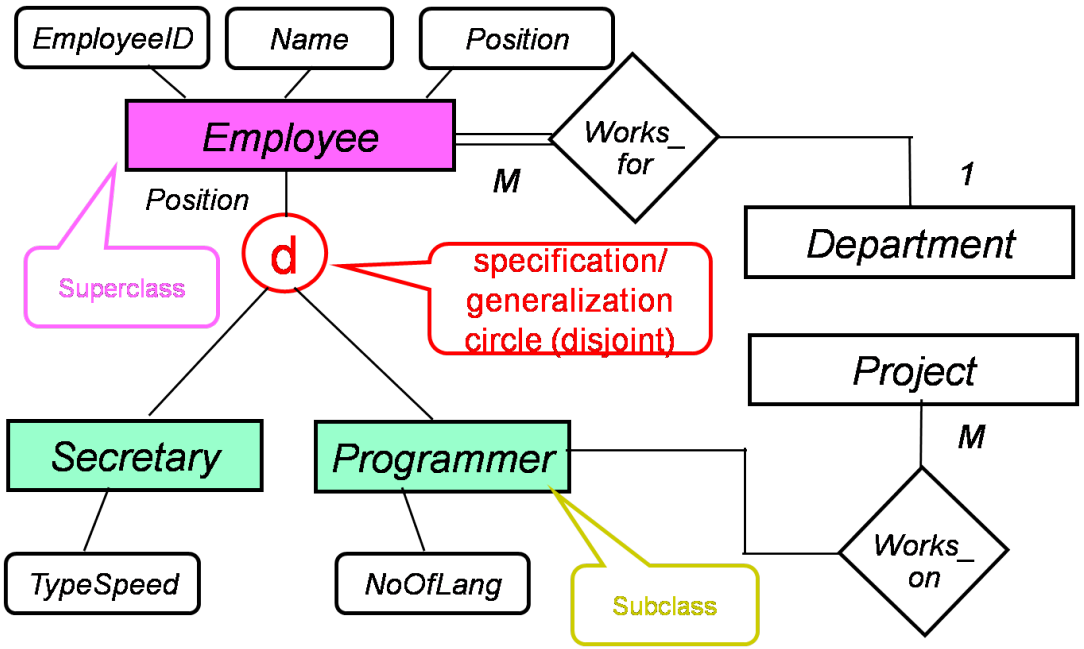
EER model further introduces the concepts of

- **is-a hierarchy**: superclass/subclass relationships
  - specialization/generalization
- attribute and relationship **inheritance**.



Enhanced Entity-Relationship Model (EER)

- An entity type may have subtypes;
- a subclass (resp. superclass) is a specialization (resp. generalization) of its superclass (resp. subclass);
- a subclass inherits all the superclass attributes and relationships.



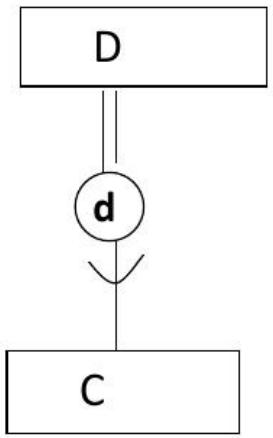
Enhanced Entity-Relationship Model (EER)

A superclass/subclass relationship is denoted by

- **d** - disjoint: an instance belongs to at most one subclass
- **o** - overlapping: an instance may belong to multiple subclasses.

The participation of a subclass may be partial and thus, we have four combinations:

- disjoint and total
- disjoint and partial
- overlapping and total
- overlapping and partial.



References

- ElMasri, Navathe, Fundamentals of Database Systems, 6th Edition, Addison Wesley.
- Hui Ma & Pavle Mogin, SWEN304 Lecture Slides, 2016
  https://ecs.victoria.ac.nz/Courses/SWEN304_2016T2/LectureSchedule