SWEN304: Mapping EER to Relational Data Model

Dr. Dionysios Athanasopoulos

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

Database Design Approach

To design a database application, a general approach includes:

- acquisition of user requirements and their analysis
- design of the EER conceptual database schema
- mapping EER into relational database schema
- normalization of the relational database schema
- design of relational external views
- defining application and database schemas (using DDL).

- Since there is no commercially available DBMS based on EER data model (they use relational data model), conceptual schema has to be **transformed** into relational data model.
- We will use for the transformation a **mapping** algorithm.

Mapping Algorithm

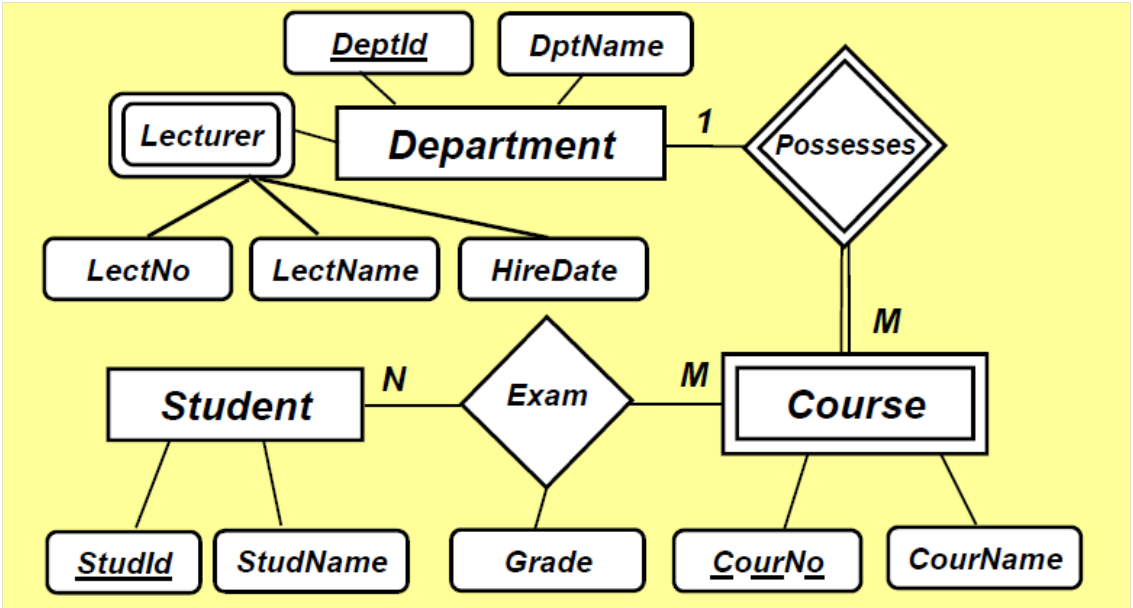
1. Mapping of regular entity types
2. Mapping of weak entity types
3. Mapping of binary 1 : 1 relationship types
4. Mapping of binary 1 : N relationship types
5. Mapping of binary M : N relationship types
6. Mapping of multivalued attributes
7. Mapping of N-ary relationship types
8. Mapping of IS-A hierarchies

General Mapping Rules

- All **entity** and **relationship** types are mapped into separate relations schemas and retain the same names.
- If entities perform different **roles**, then they are renamed.
- All **single-valued** attributes of an entity or relationship type are mapped into attributes of the corresponding relation schema and retain the same names;
  - we include only the simple attributes of a composite attribute.
- All **multi-valued** attributes are mapped into new relations.

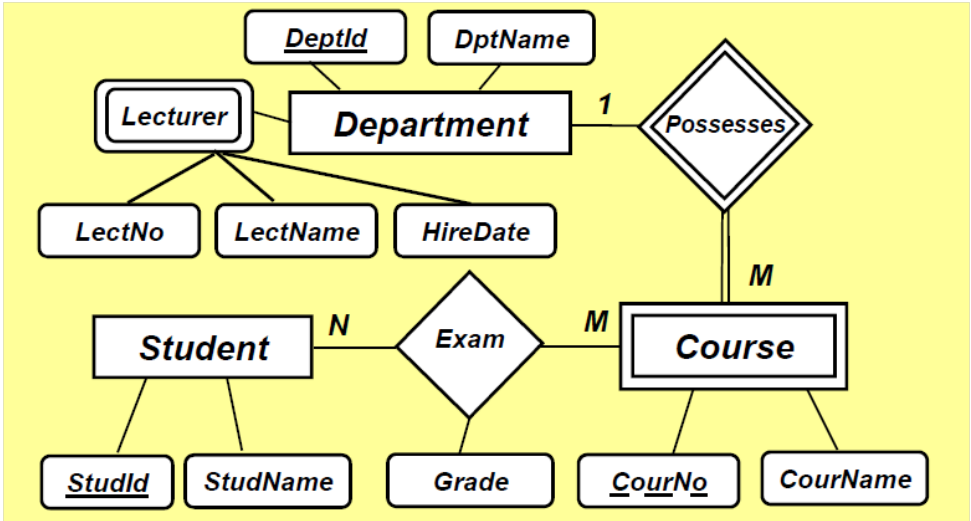
1. Mapping of Regular Entity Types

- Map each regular entity type into a relation schema
  - make the **key** of the entity type the primary key of the relation schema;
  - follow the general mapping rules for the attributes of an entity.
- Applying this step in the following EER schema, we define two relation schemas:
  - `Department (DeptId, DeptName)`
  - `Student (StudId, StudName)`



2. Mapping of Weak Entity Types

- Map each **weak** entity type W with **owner** entity types E_1, E_2, ..., E_k into a relation schema R
  - include all the attributes of W as attributes of R
  - include as foreign keys of R the primary keys of the relations that correspond to E_1, E_2, ..., E_k
  - the primary key of R is the combination of the primary keys of the owners and the partial keys of the weak entities.
- Applying this step, we define one relation schema:
  - `Course (DeptId, CourNo, CourName)`



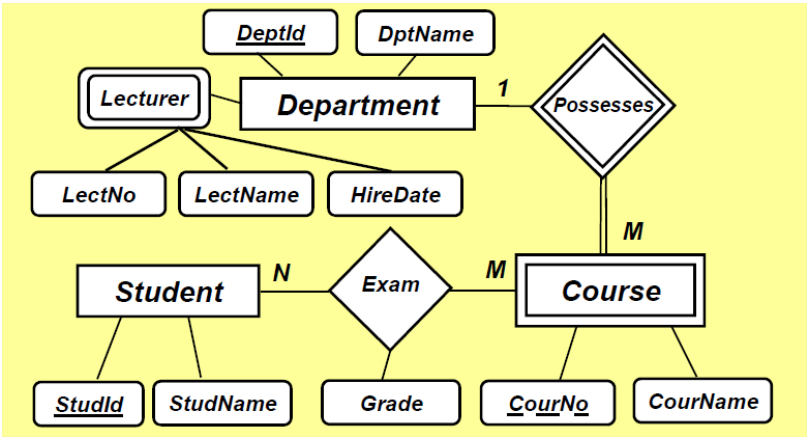
3. Mapping of Binary 1 : 1 Relationship Types

For each binary 1:1 relationship *R*, we define the relation schemas *S* and *T* that correspond to the entities associated by *R* by using the **foreign key** approach:

- choose one of the relations (e.g. *S*) and include as a foreign key in *S* the primary key of *T*
  - it is better to choose an entity type with **total participation** in *R* in the role of *S*
  - include all the simple attributes of *R* as attributes of *S*.

If **both** participations of *S* and *T* are **total**, then we use the **merged relation** approach:

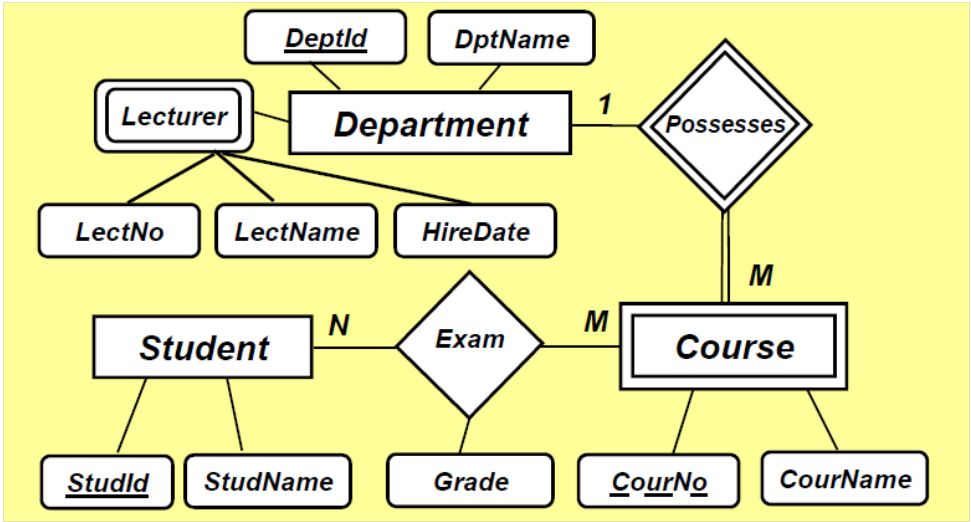
- merge the two entities and the relationship into a single relation.



4. Mapping of Binary 1 : N Relationship Types

For each **regular** binary 1 : N relationship *R*, we define the relation schemas *S* representing the entity type on the 1-side and *T* on the M-side, respectively;

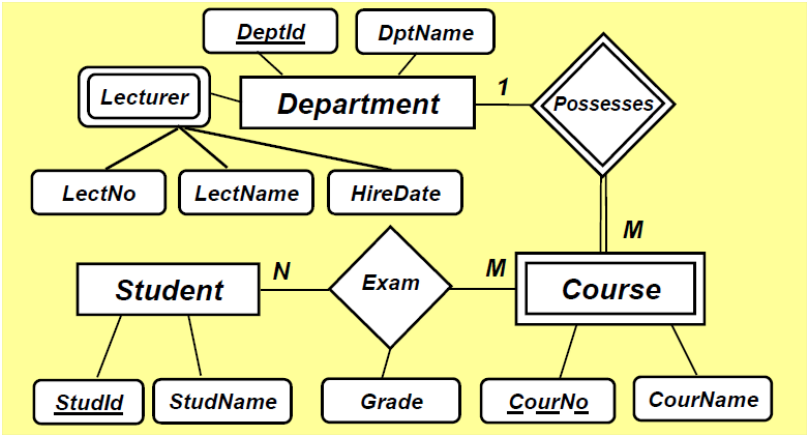
- we include the primary key of *S* as a foreign key in *T*
- we include the simple attributes of *R* as attributes of *T*
- we do not create a new entity.



5. Mapping of Binary M : N Relationship Types

For each **regular** binary M : N relationship *R*, we define the relation schema *R*;

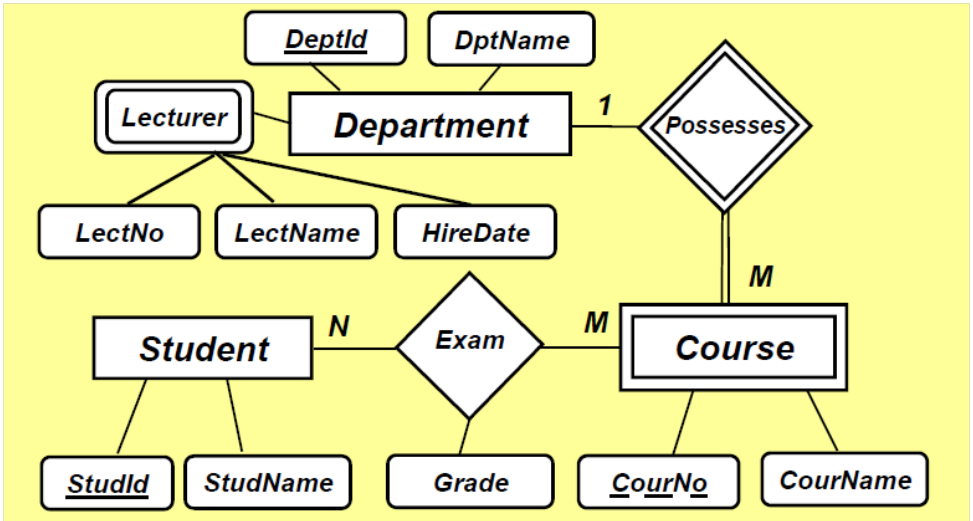
- we include as foreign keys in *R* the primary keys of the participating entities
- the primary key of *R* is composed by the primary keys of the participating entities
- we include any simple attribute of the relationship into the relation *R*.
- Applying this step, we define one relation schema:
  - `Exam(DeptId, CourNo, StudId, Grade)`



6. Mapping of Multivalued Attributes

For each multivalued attribute V of the relationship T, already represented by the relation T, we define a new relation V;

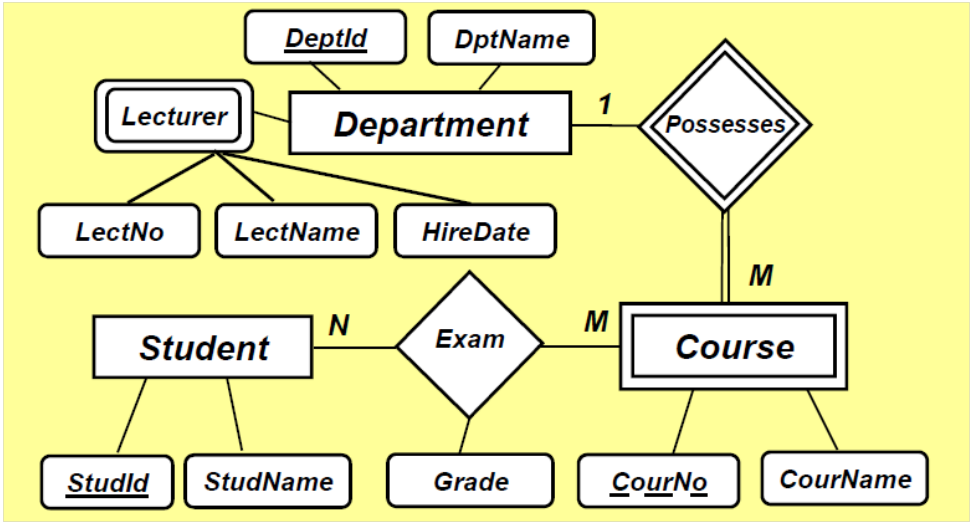
- we include all the simple and single-valued attributes of V into the relation
- we include the primary key of T into the relation
- the primary key of V is the combination of the keys of T and V.
- Applying this step, we define one relation schema:
  - `Lecturer(DeptId, LectNo, LectName, HireDate)`



7. Mapping of N-ary Relationship Types

For each N-ary relationship type R (N > 2), we define a new relation R;

- we include as foreign key in R the primary keys of the participating entities
- we include any simple attributes of R into the relation
- the primary key of R is the combination of the keys of the participating entities.



8. Mapping of IS-A Hierarchies

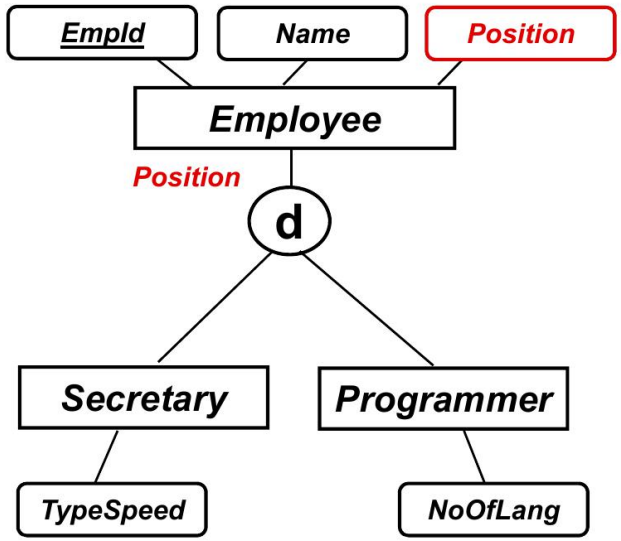
It can be done in three ways:

1. **multiple relations**: it is used in any case
   - the superclass and each subclass are mapped into separate relations
     - the superclass as a regular entity type
     - each subclass as a weak entity type
2. **multiple relations - subclasses only**: it is used for a specialization whose subclasses have **total** participation
   - each subclass is mapped into one relation schema, containing the union of the attributes of the superclass and the subclass
   - the primary key of each relation is the primary of the superclass
   - no relation schema is defined for the superclass
3. **single relation**: it is used for a specialization whose subclasses are **disjoint**
   - the superclass and all the subclasses are mapped into the same relation schema.

8. Mapping of IS-A Hierarchies: Example

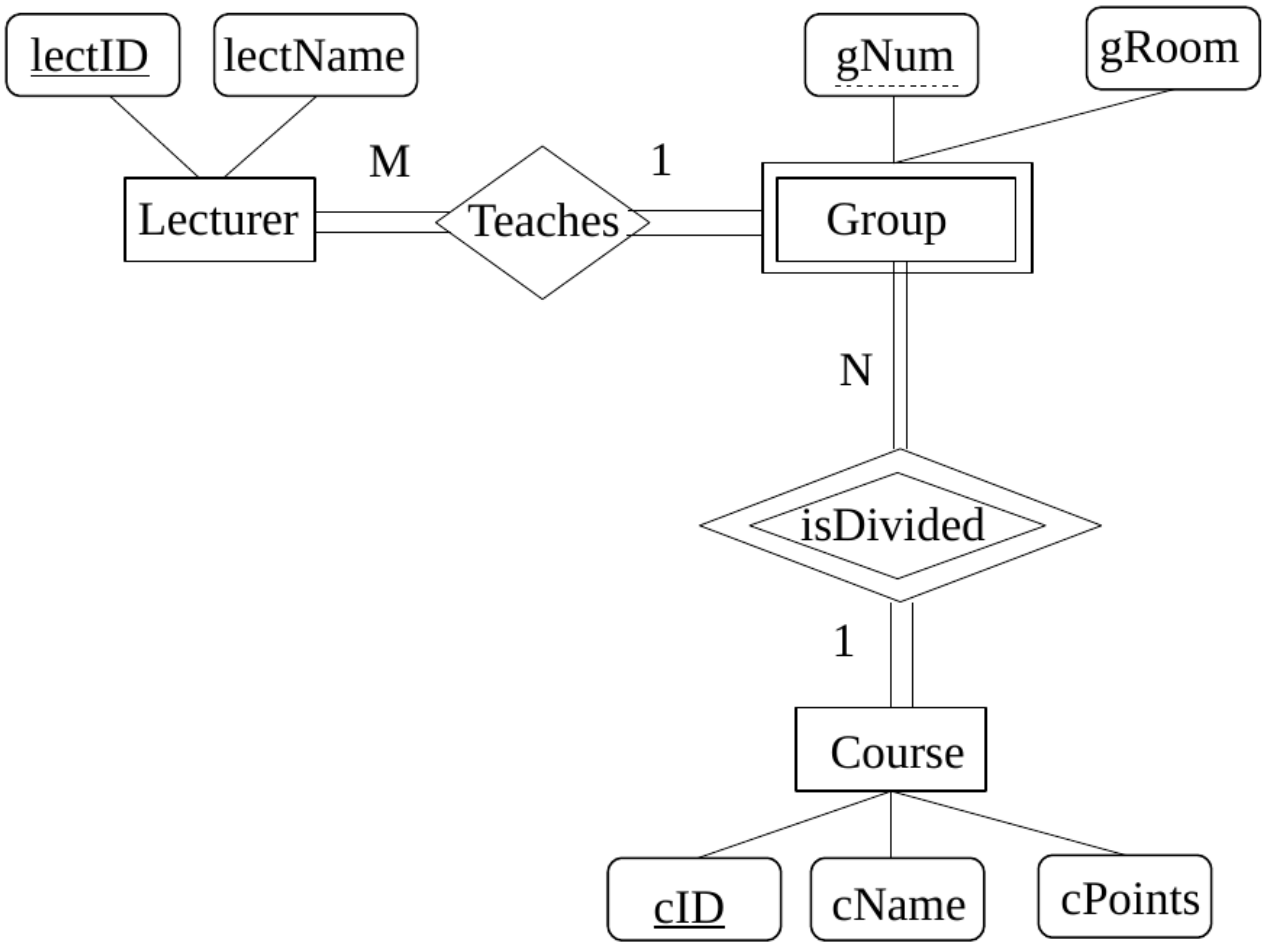
Applying this step to the following EER diagram, we have two options:

1. Multiple relations:
   - `Employee(EmpId, Name, Position)`
   - `Secretary(EmpId, TypeSpeed)`
   - `Programmer(EmpId, NoOfLang)`
2. Single relation:
   - `Employee(EmpId, Name, Position, TypeSpeed, NoOfLang)`



Mapping Algorithm: Exercise

- Transform the following ER diagram into a relational data model:



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