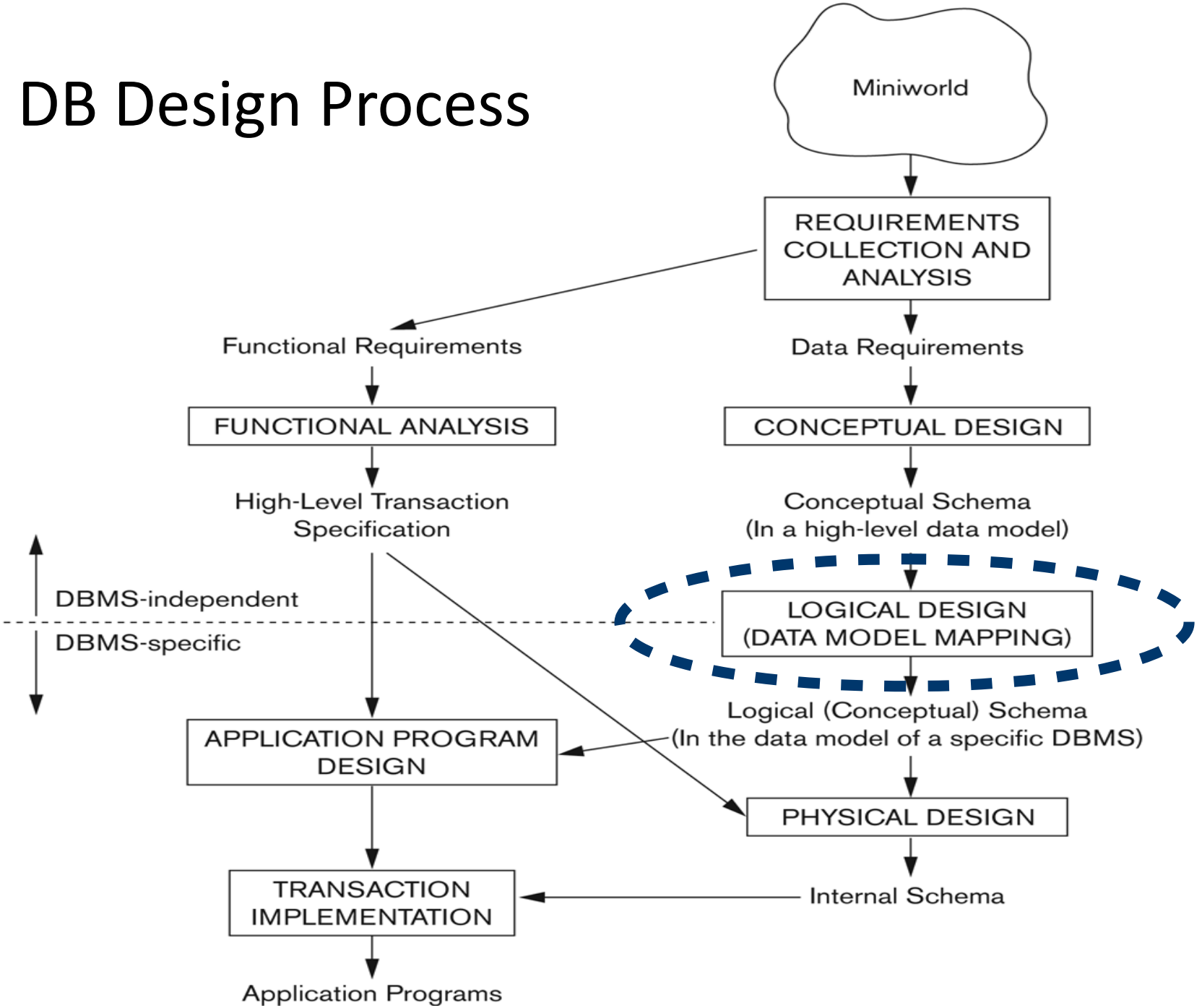
Database Technology

Topic 4:
Mapping of EER Diagrams to Relations

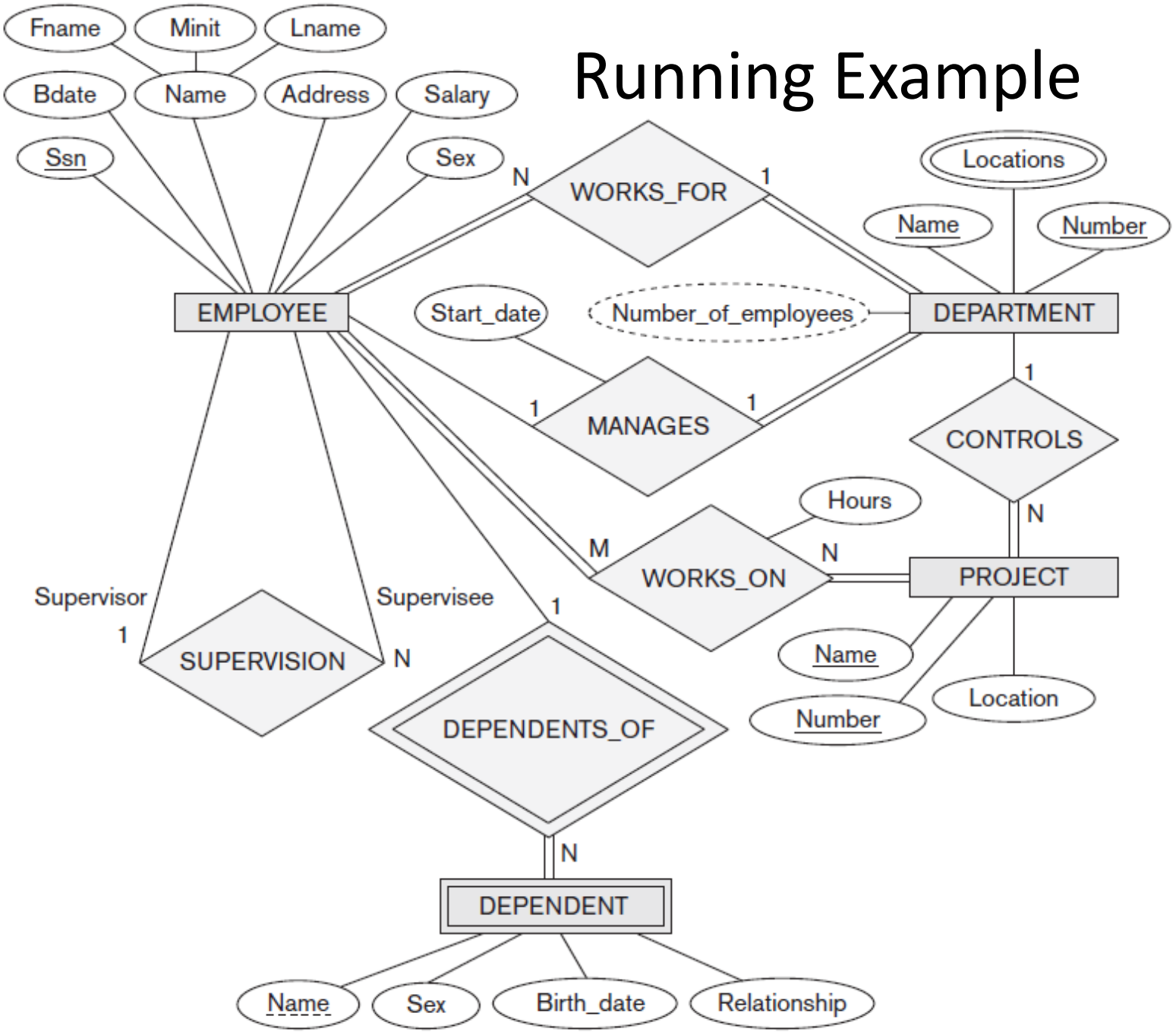
Olaf Hartig
olaf.hartig@liu.se

LINKÖPING UNIVERSITY

Recall: DB Design Process



Running Example



Algorithm for Mapping from the ER Model
to the Relational Model

Step 1: Map Regular Entity Types

- For each regular entity type, create a relation schema *R* that includes all the single-valued attributes of *E*
  - "Flatten" composite attributes (e.g., Name of Employee)
  - Example renames some attributes (e.g., Dname), but not needed
  - Pick one of the keys as primary key, declare others to be unique
  - Resulting relations are called **entity relations**
  - Each tuple represents an entity instance

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

Step 2: Map Weak Entity Types

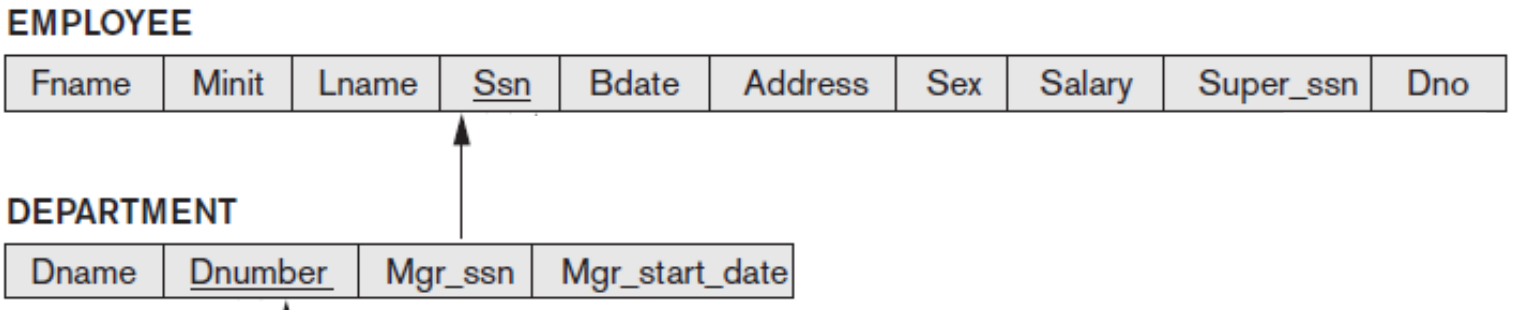
- For each weak entity type, create a relation schema *R*
  - Include all single-valued attributes of the weak entity type *and of the identifying relationship* as attributes of *R*
  - Include primary key attribute of identifying entity as foreign key attribute of *R*
  - Primary key of *R* is primary key of identifying entity together with partial key from *R*
- Omit the identifying relationship when subsequently translating (other) relationship types to relation schemas

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Step 3: Binary 1:1 Relationship Types

- For each binary 1:1 relationship type *R*, identify relation schemas that correspond to entity types participating in *R*
- Apply one of three possible approaches:
  1. **Foreign key approach**
     - Add primary key of one participating relation as foreign key attribute of the other, which will also represent *R*
       - If only one side is *total*, choose it to represent *R* *(why?)*
     - Declare foreign key attribute as unique



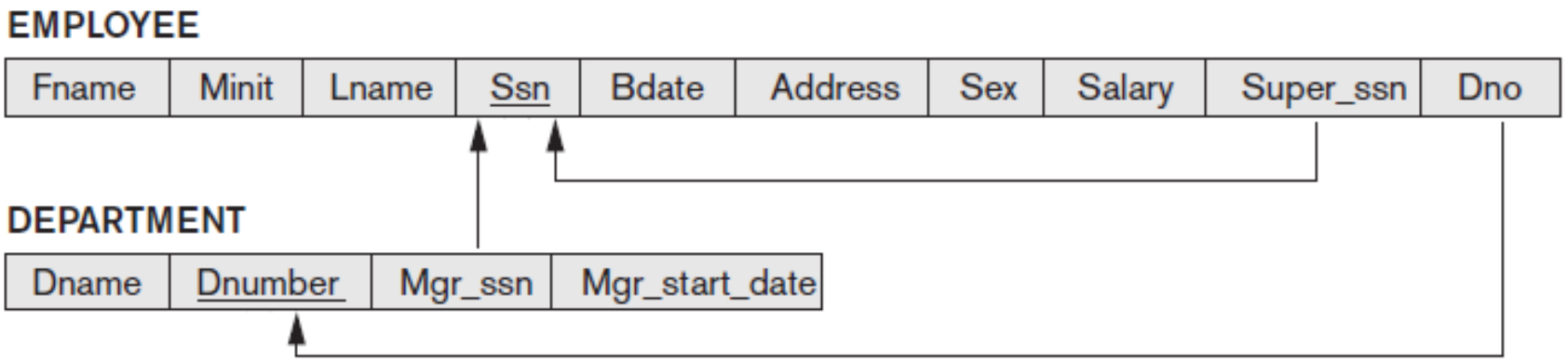
- Add single-valued attributes of relationship type as attributes of *R*

Step 3: Binary 1:1 Relationship Types

- For each binary 1:1 relationship type *R*, identify relation schemas that correspond to entity types participating in *R*
- Apply one of three possible approaches:
  2. **Merged relationship approach**
     - Possible if both participations are total
     - Combine the two relation schemas into one, which will also represent *R*
     - Make one of the primary keys unique instead
  3. **Cross-reference** or **relationship relation approach**
     - Create new relation schema for *R* with two foreign key attributes being copies of both primary keys
     - Declare one of the attributes as primary key, the other one as unique
- Add single-valued attributes of relationship type as attributes of *R*

Step 4: Binary 1:*N* Relationship Types

- **Foreign key approach**
  - Identify relation schema *S* that represents participating entity type at *N*-side of 1:*N* relationship type
  - Include primary key of other entity type (1-side) as foreign key in *S*
- **Relationship relation approach** (next slide)
- Include single-valued attributes of relationship type as attributes

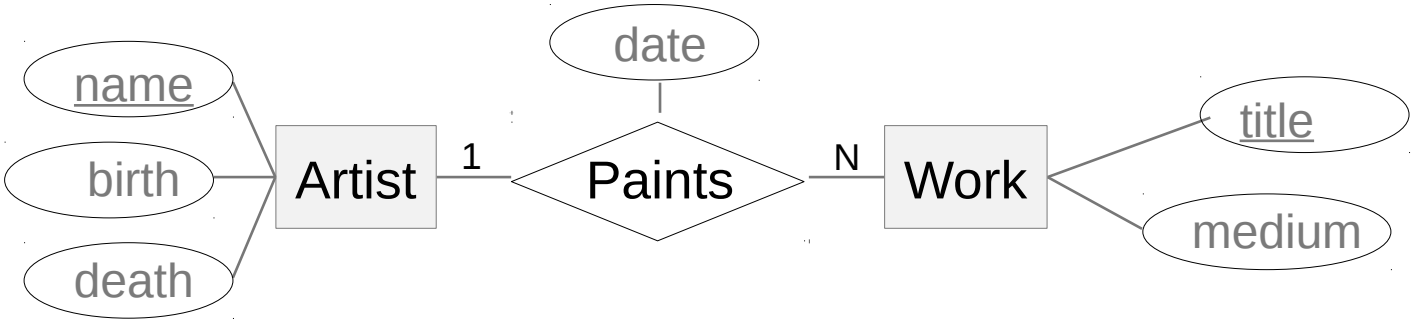


Step 4: Binary 1:*N* Relationship Types

- **Foreign key approach**
- **Relationship relation approach**
  - Create new relation schema for relationship type with two foreign key attributes being copies of both primary keys
  - Declare the foreign key attribute for the relation schema corresponding to the participating entity type *on the N-side* as primary key
- Include single-valued attributes of relationship type as attributes

Quiz

Consider the given ER diagram and relational DB schema



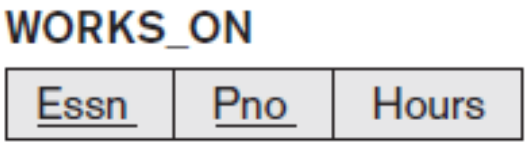
Artist(<u>name</u>,birth,death), Work(<u>title</u>,medium)

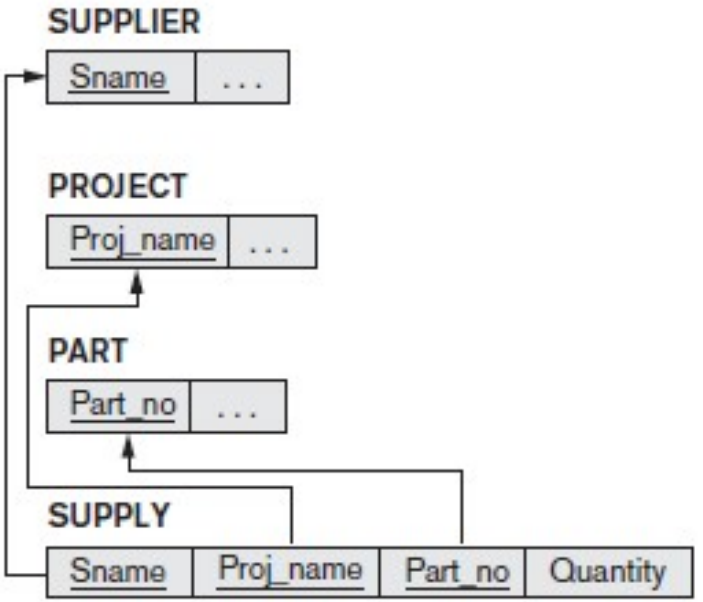
The Paints relationship can be represented by:

A. introducing a third schema: Paints(name,<u>title</u>,date)
B. extending the Work schema to be Work(<u>title</u>,medium,name,date)
C. extending the Artist schema to be Artist(<u>name</u>,birth,death,title,date)
D. either A or B above
E. any of A, B, or C above

Step 5: Binary *M*:*N* and Higher Order Relationship Types

- For each binary *M*:*N* relationship type or ternary or higher order relationship type, create a new relation *S*
  - Include primary key of participating entity types as foreign key attributes in *S*
  - Make all these attributes the primary key of *S*
  - Include any simple attributes of relationship type in *S*





Step 6: Map Multivalued Attributes

- For each multivalued attribute create new relation *R*
  - Add attribute to hold multivalued attribute values
    - If multivalued attribute is composite, include its simple components
  - Add attribute(s) for primary key of relation schema for entity type or relationship type to be foreign key for *R*
  - Primary key of *R* is the combination of *all* its attributes

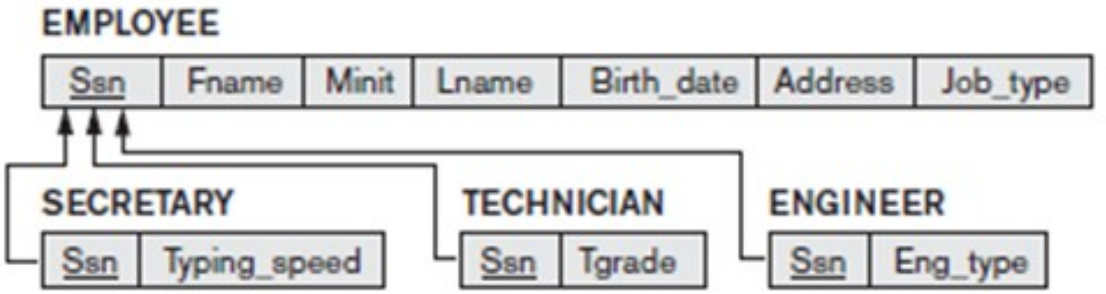
DEPT_LOCATIONS

Dnumber	Dlocation

Mapping Constructs from the EER Model

Options for Mapping Specialization/Generalization

- For *any* specialization (total or partial, disjoint or overlapping)
  1. Separate relation per supertype and subtypes



  2. Single relation with Boolean type attributes for every subtype
     - Add all attributes of all subtypes

Options for Mapping Specialization/Generalization (cont'd)

- For *total specializations* (and generalizations) *only*
  - Separate relation per subclass relations only
  - Overlapping subtypes will result in multiple tuples per entity

- For *disjoint specializations only*
  - Single relation with one type attribute
  - **Type** or **discriminating attribute** indicates subtype of tuple
  - Might require many NULL values if several specific attributes exist in subtypes

EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

Summary and Example

Summary

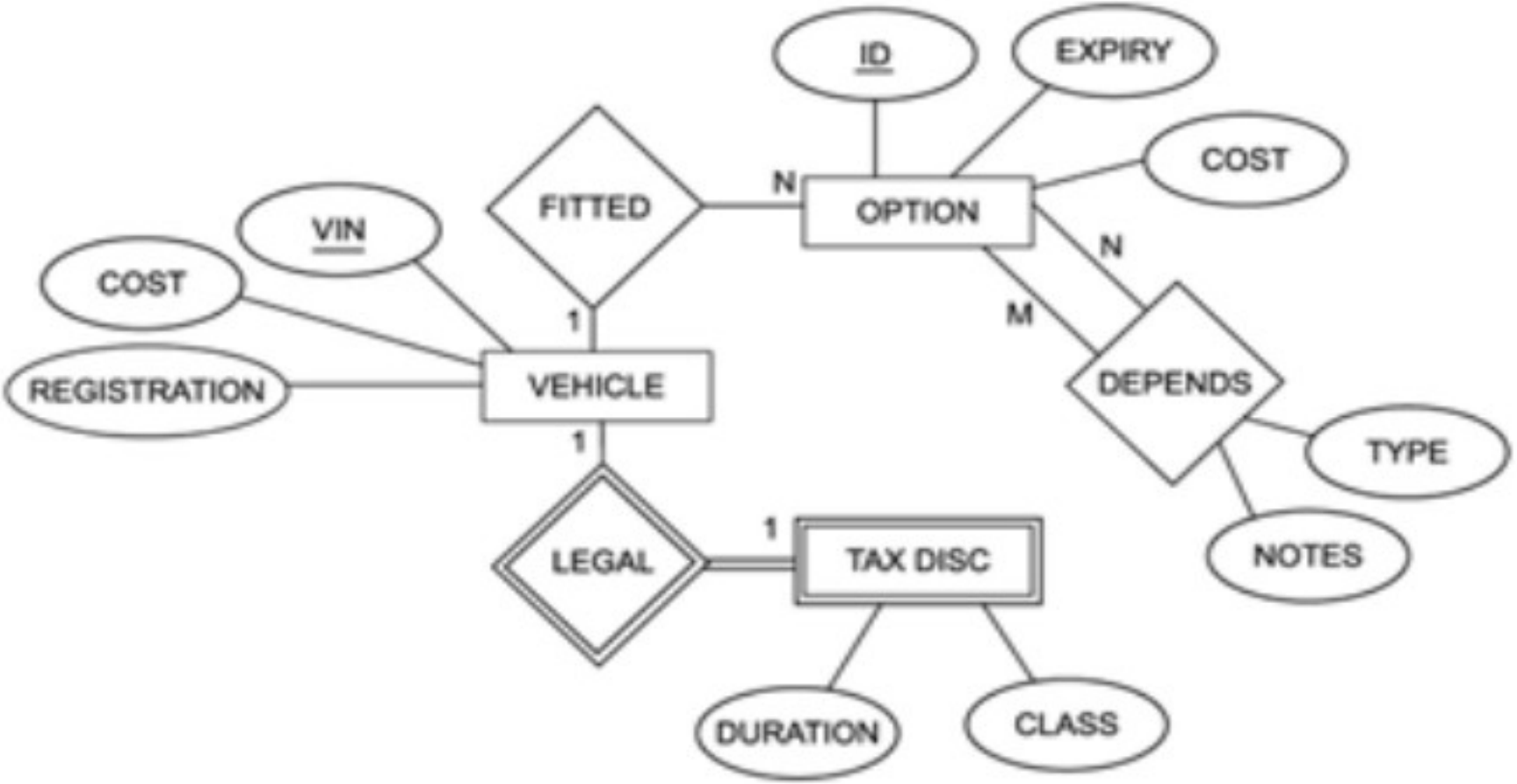
- Algorithm for ER-to-relational mapping

ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite attribute	Set of simple component attributes
Multivalued attribute	Relation and foreign key
Value set	Domain
Key attribute	Primary (or secondary) key

- Extensions for mapping constructs from EER model

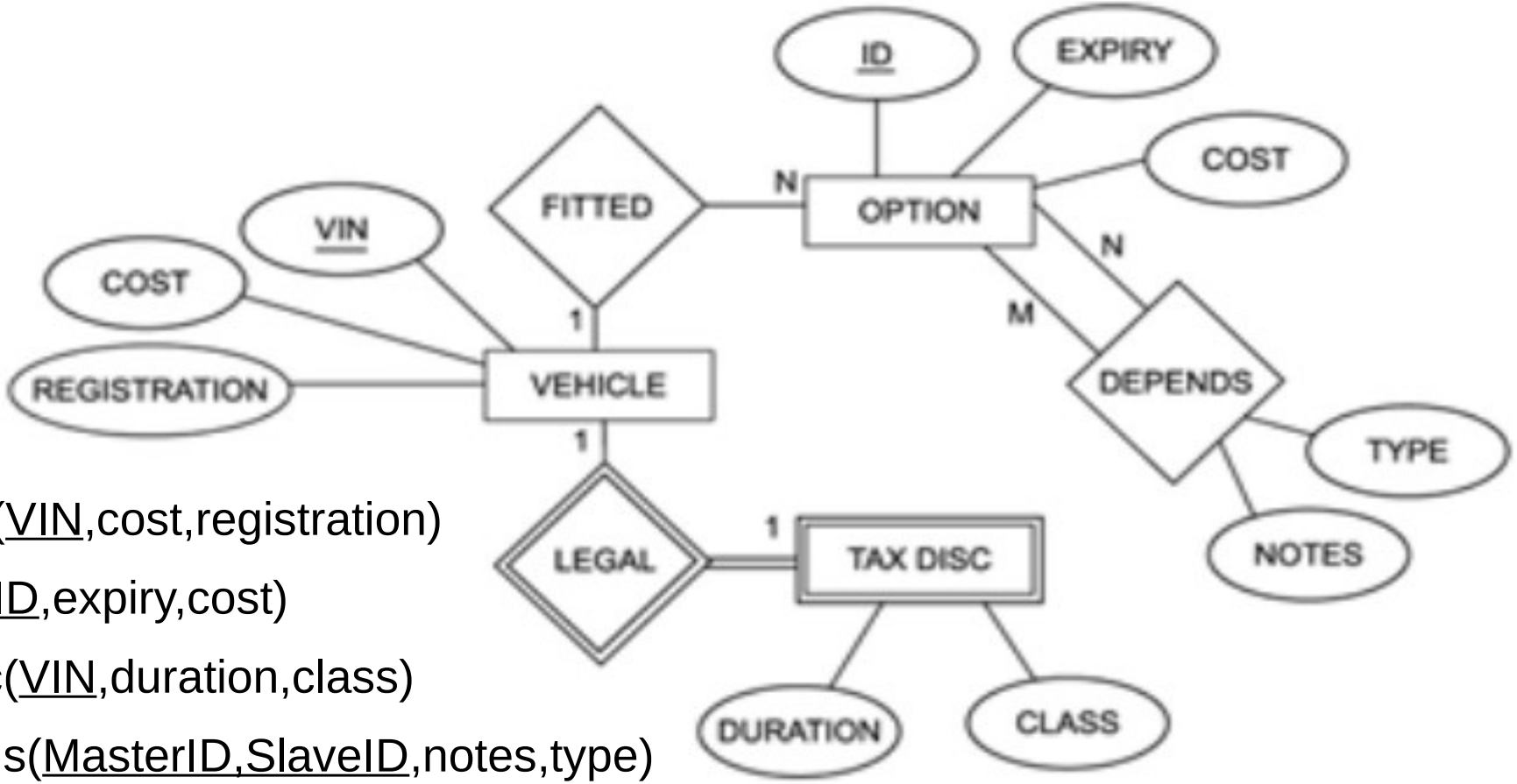
Exercise

Translate the following ER Diagram into a relational database schema.



Exercise

Translate the following ER Diagram into a relational database schema.



Vehicle(VIN,cost,registration)

Option(ID,expiry,cost)

TaxDisc(VIN,duration,class)

Depends(MasterID,SlaveID,notes,type)

Fitted(VIN,ID) *or extend* Option(…) *to be* Option(ID,expiry,cost,VIN)

www.liu.se

LINKÖPING UNIVERSITY