Database Systems

LESSON 05: ER & EER TO RELATIONAL MAPPING

September 2019

Contents

ER-to-Relational Mapping Algorithm

- Step 1: Mapping of Regular Entity Types
- Step 2: Mapping of Weak Entity Types
- Step 3: Mapping of Binary 1:1 Relation Types
- Step 4: Mapping of Binary 1:N Relationship Types.
- Step 5: Mapping of Binary M:N Relationship Types.
- Step 6: Mapping of Multivalued attributes.
- Step 7: Mapping of N-ary Relationship Types.

Mapping EER Model Constructs to Relations

- Step 8: Options for Mapping Specialization or Generalization.
- Step 9: Mapping of Union Types (Categories).

GOALS during Mapping

Preserve all information (that includes all attributes)

Maintain the constraints to the extent possible (Relational Model cannot preserve all constraints- e.g., max cardinality ratio such as 1:10 in ER; exhaustive classification into subtypes, e.g., STUDENTS are specialized into Domestic and Foreign)

Minimize null values

The mapping procedure described has been implemented in many commercial tools.

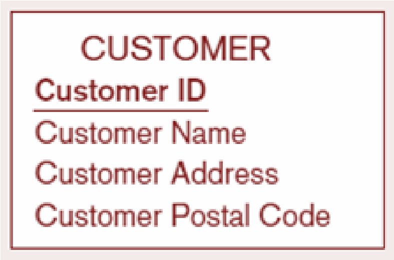
ER-to-Relational Mapping Algorithm

Step 1: Mapping of Regular Entity Types.

- For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
- Choose one of the key attributes of E as the primary key for R.
- If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Example

Regular entity type



CUSTOMER

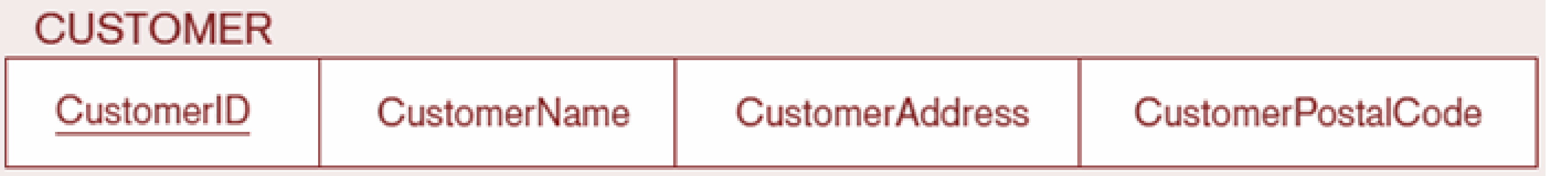
Customer ID

Customer Name

Customer Address

Customer Postal Code

Resulting relation

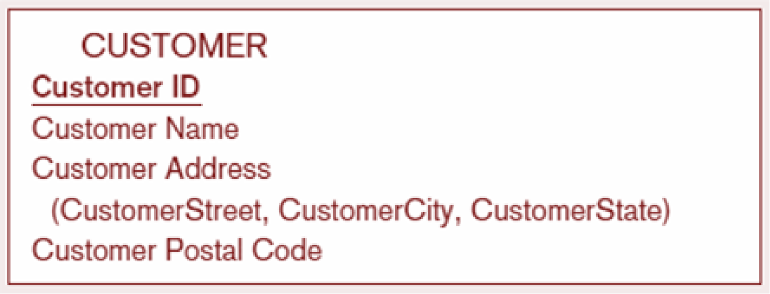


CUSTOMER

CustomerID	CustomerName	CustomerAddress	CustomerPostalCode

Example(contd.)

Regular entity type with composite attributes



CUSTOMER

Customer ID

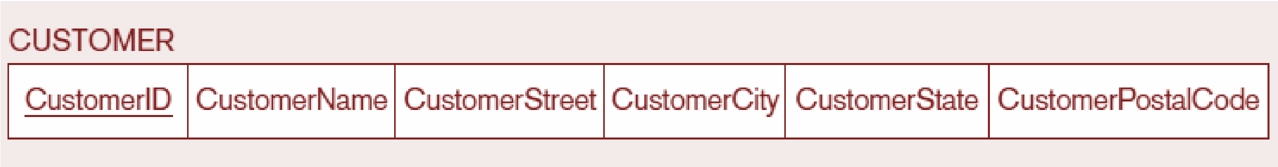
Customer Name

Customer Address

(CustomerStreet, CustomerCity, CustomerState)

Customer Postal Code

Relation



CUSTOMER

CustomerID	CustomerName	CustomerStreet	CustomerCity	CustomerState	CustomerPostalCode

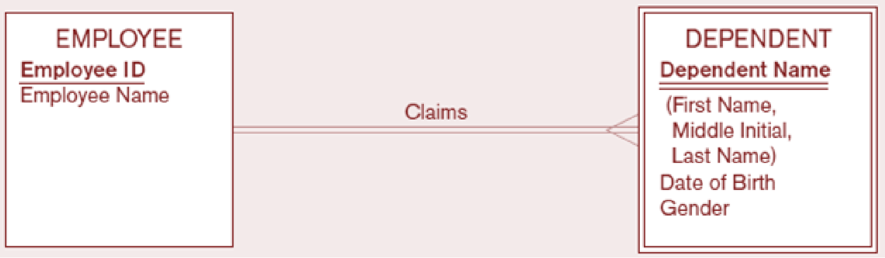
ER-to-Relational Mapping Algorithm (contd.)

Step 2: Mapping of Weak Entity Types

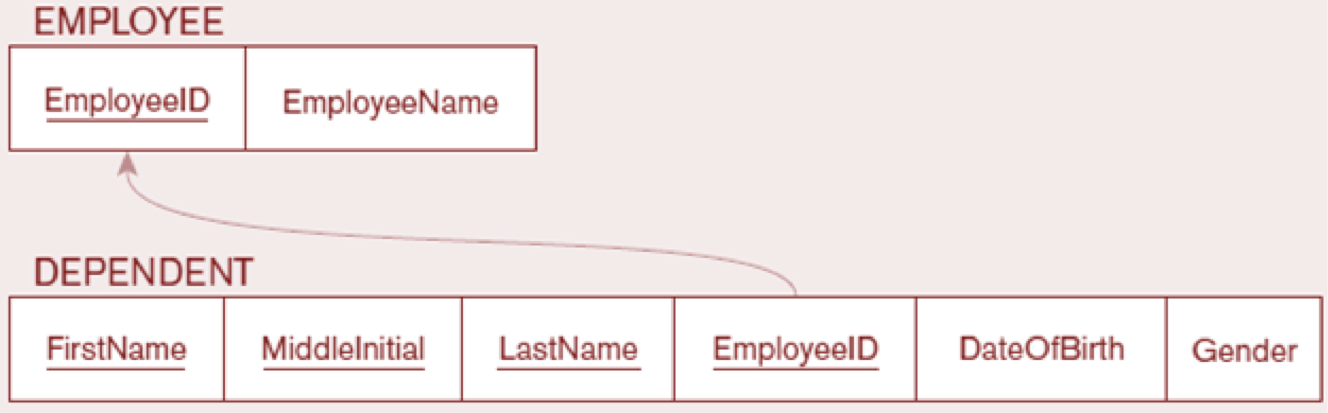
- For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
- Also, include as foreign key attributes of R the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s).
- The primary key of R is the *combination of* the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any.

Example

Weak entity:



Resulting relations:



EMPLOYEE

EmployeeID	EmployeeName

DEPENDENT

FirstName	MiddleInitial	LastName	EmployeeID	DateOfBirth	Gender

Composite primary key

ER-to-Relational Mapping Algorithm (contd.)

Step 3: Mapping of Binary 1:1 Relation Types

- For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.

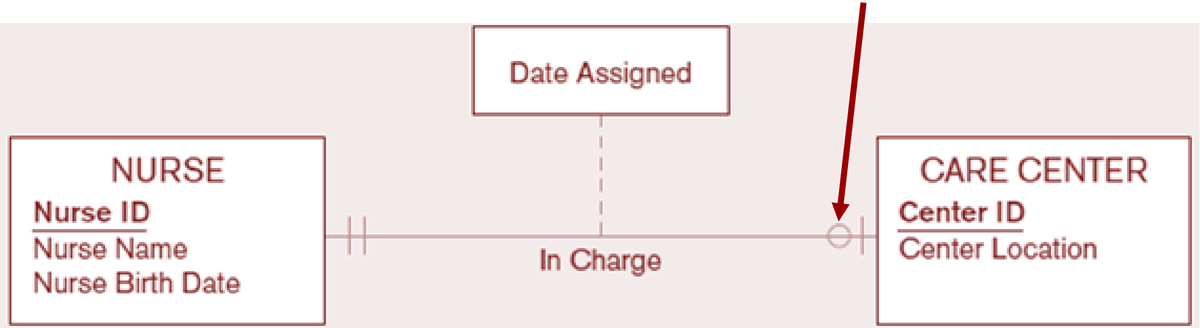
There are three possible approaches:

1. **Foreign Key (2 relations) approach:** Choose one of the relations-say S-and include 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.
2. **Merged relation (1 relation) option:** An alternate mapping of a 1:1 relationship type is possible by merging the two entity types and the relationship into a single relation. This may be appropriate when both participations are total.
3. **Cross-reference or relationship relation (3 relations) option:** The third alternative is to set up a third relation R for the purpose of cross-referencing the primary keys of the two relations S and T representing the entity types.

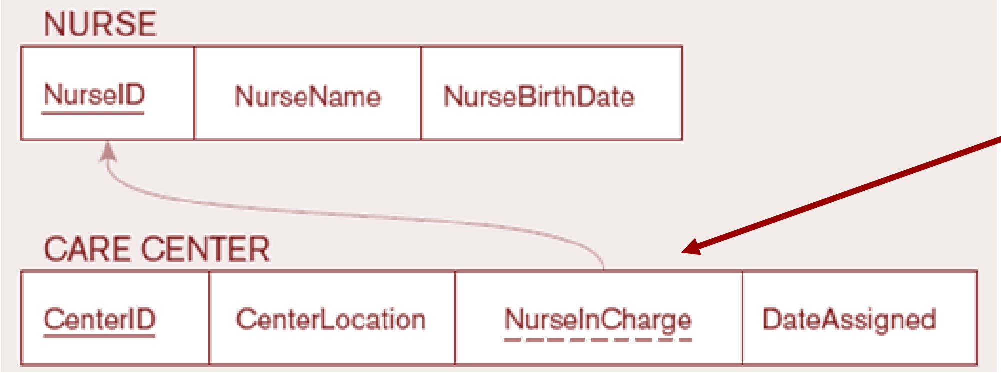
Example

1:1 optional relation

optional



Resulting relations:


NURSE

NurseID	NurseName	NurseBirthDate

CARE CENTER

CenterID	CenterLocation	NurseInCharge	DateAssigned

Not NULL to enforce total participation

ER-to-Relational Mapping Algorithm (contd.)

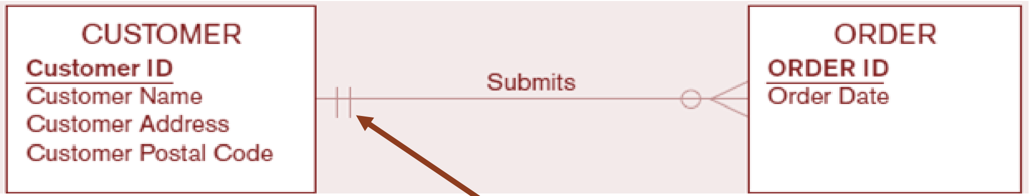
Step 4: Mapping of Binary 1:N Relationship Types.

- For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
- Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
- Include any simple attributes of the 1:N relation type as attributes of S.

An alternative approach is to use a Relationship relation (cross referencing relation) – this is rarely done.

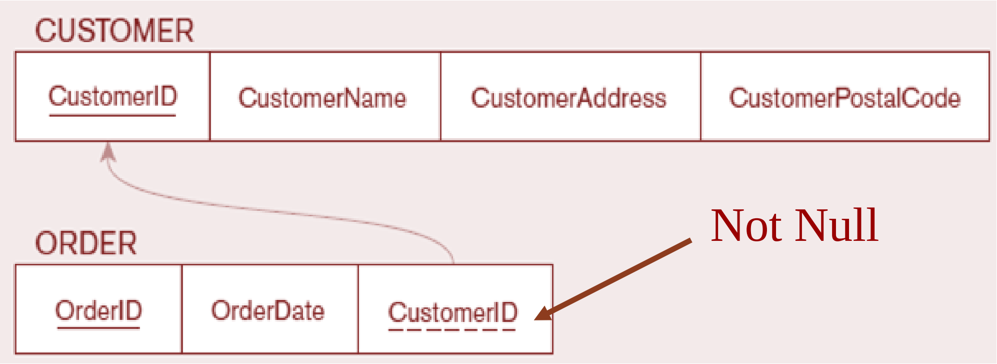
Example

1:N relationship



mandatory one

Resulting relations:


CUSTOMER

CustomerID	CustomerName	CustomerAddress	CustomerPostalCode

ORDER

OrderID	OrderDate	CustomerID

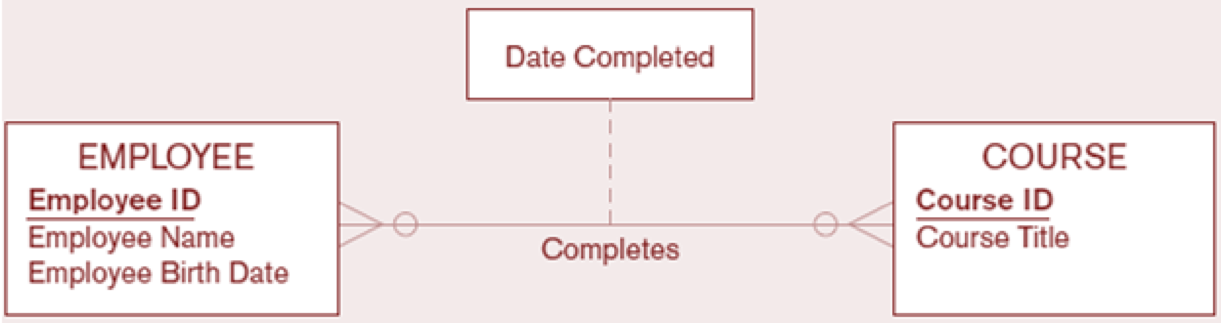
Not Null

ER-to-Relational Mapping Algorithm (contd.)

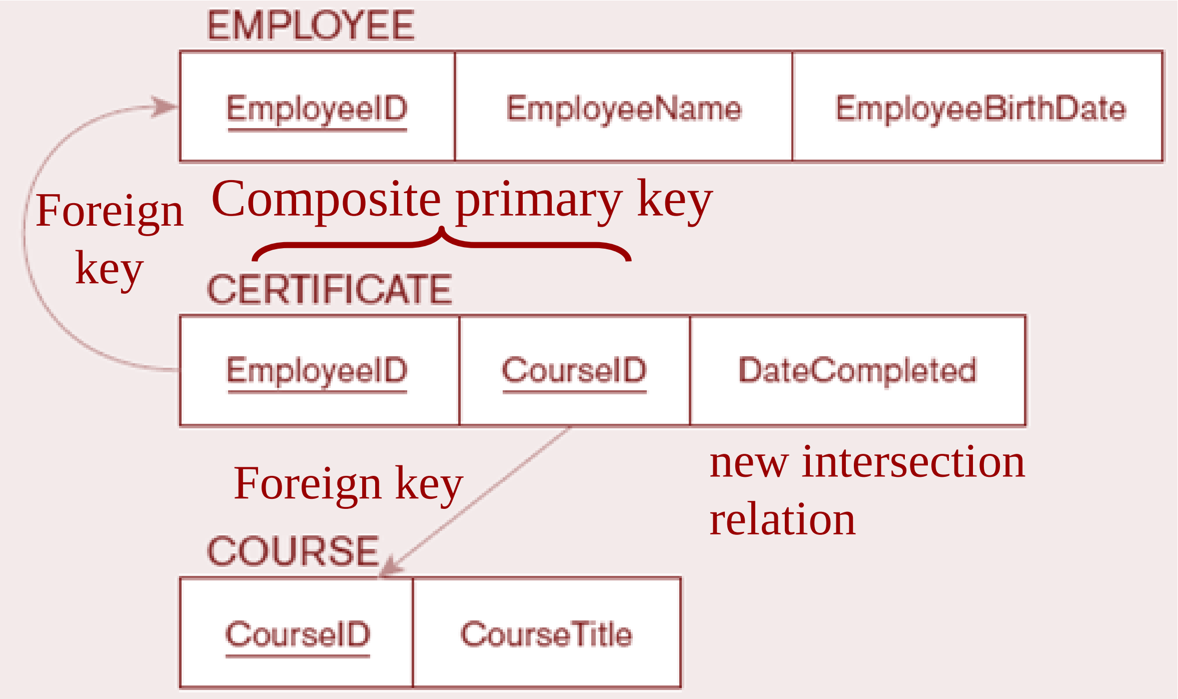
Step 5: Mapping of Binary M:N Relationship Types.

- For each regular binary M:N relationship type R, *create a new relation* S to represent R. This is a *relationship relation.*
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; *their combination will form the primary key* of S.
- Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Example



Completes relationship (M:N)



EMPLOYEE

EmployeeID	EmployeeName	EmployeeBirthDate

Foreign key

Composite primary key

CERTIFICATE

EmployeeID	CourseID	DateCompleted

Foreign key

new intersection relation

COURSE

CourseID	CourseTitle

Three resulting relations

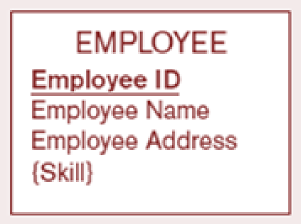
ER-to-Relational Mapping Algorithm (contd.)

Step 6: Mapping of Multivalued attributes.

- For each multivalued attribute A, create a new relation R.
- This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
- The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Example

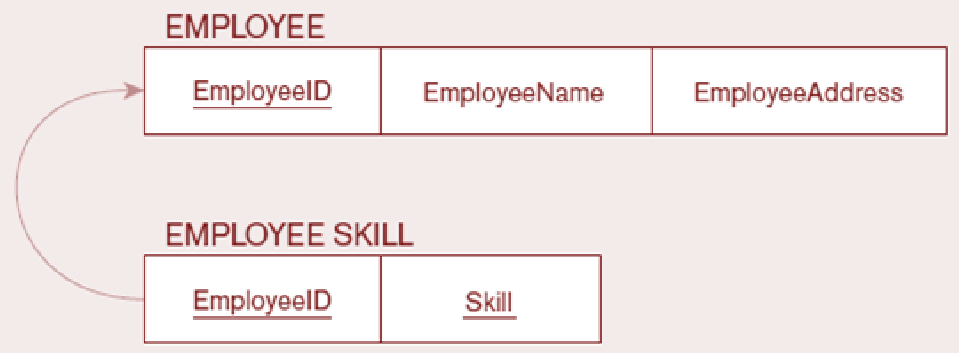
Regular entity type with multi-valued attributes



EMPLOYEE

<u>Employee ID</u>
Employee Name
Employee Address
{Skill}

Resulting relations:



EMPLOYEE

<u>EmployeeID</u>	EmployeeName	EmployeeAddress

EMPLOYEE SKILL

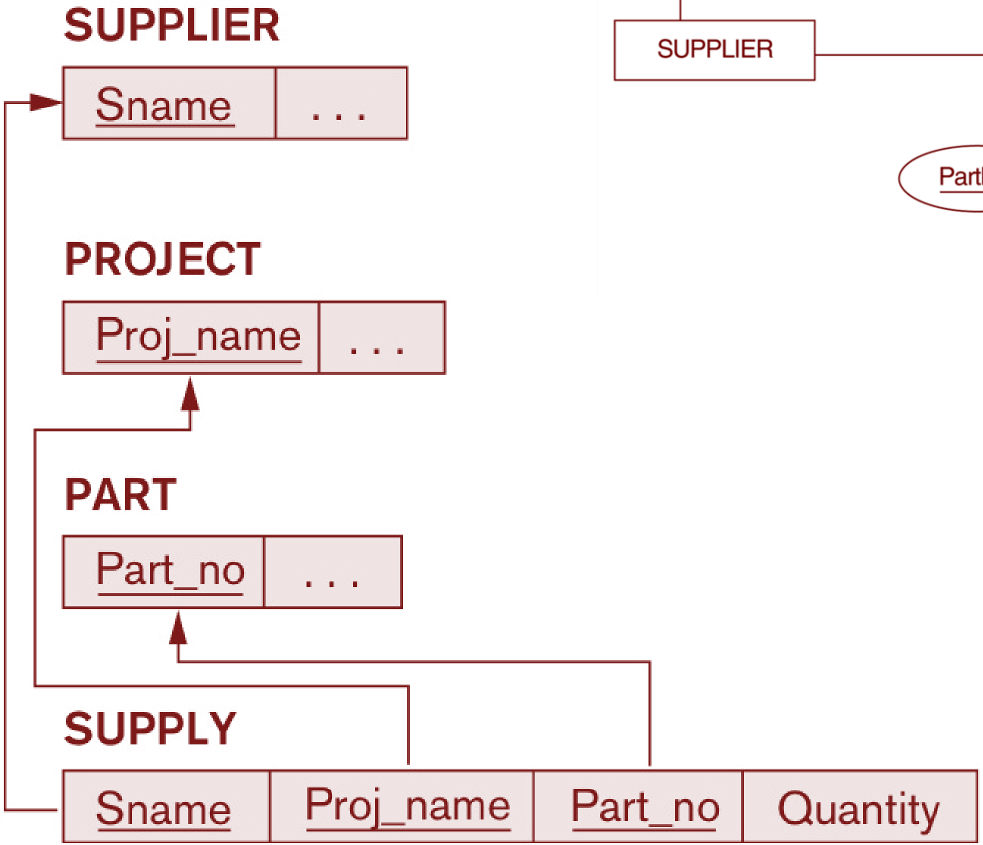
<u>EmployeeID</u>	<u>Skill</u>

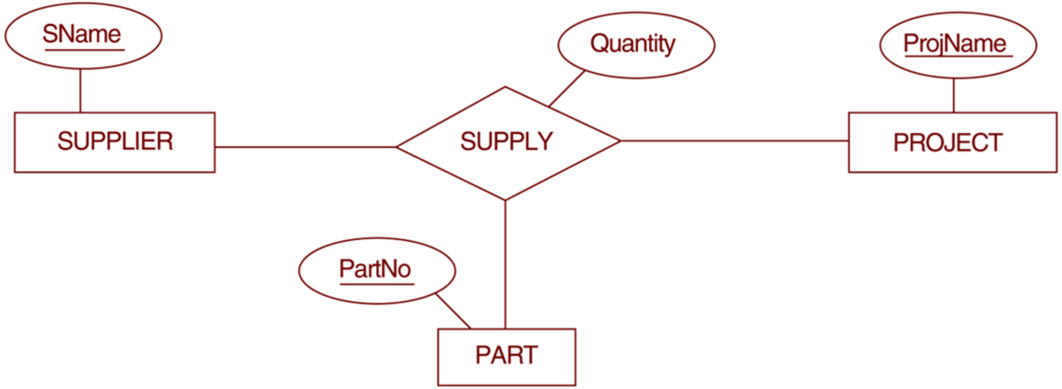
ER-to-Relational Mapping Algorithm (contd.)

Step 7: Mapping of N-ary Relationship Types.

- For each n-ary relationship type R, where n>2, create a new relationship S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
- Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

Example



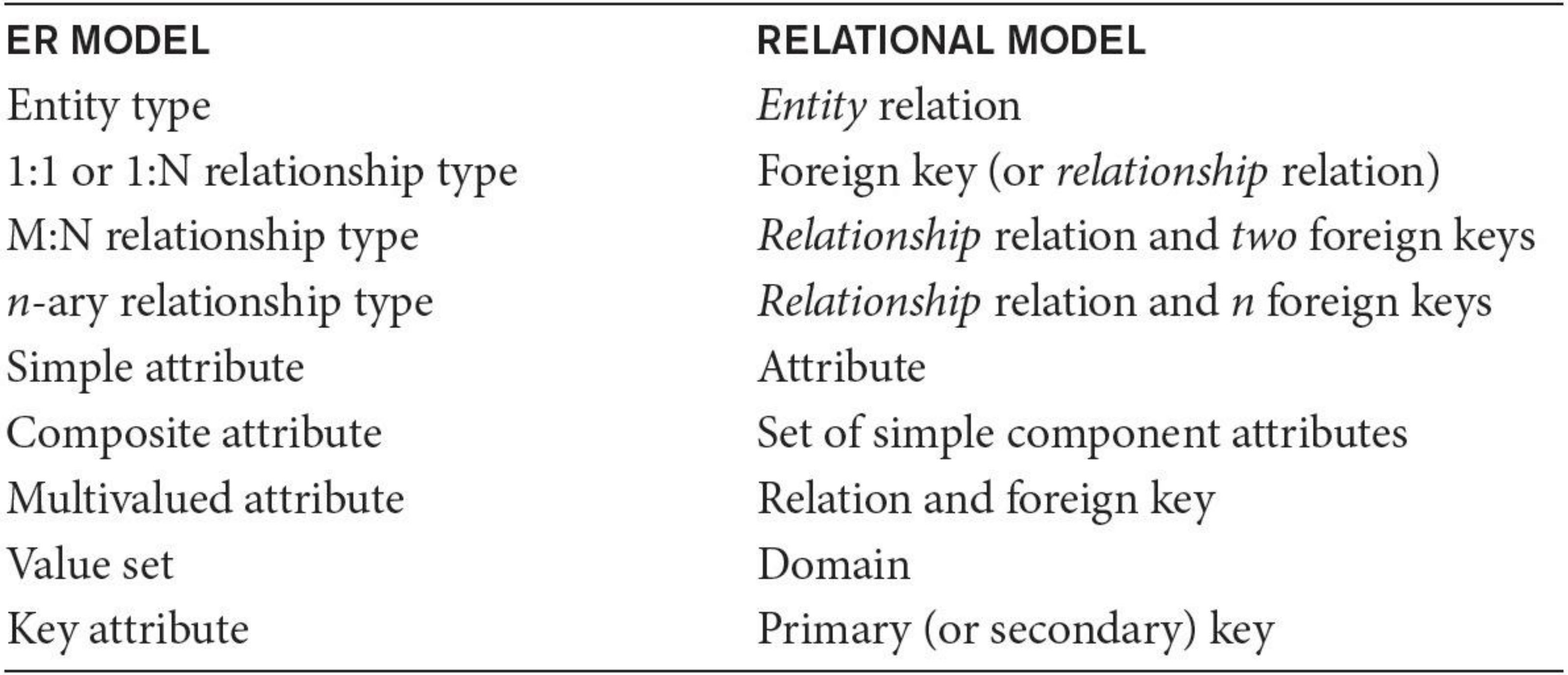


N-ary relationship type SUPPLY

Resulting SUPPLY relation

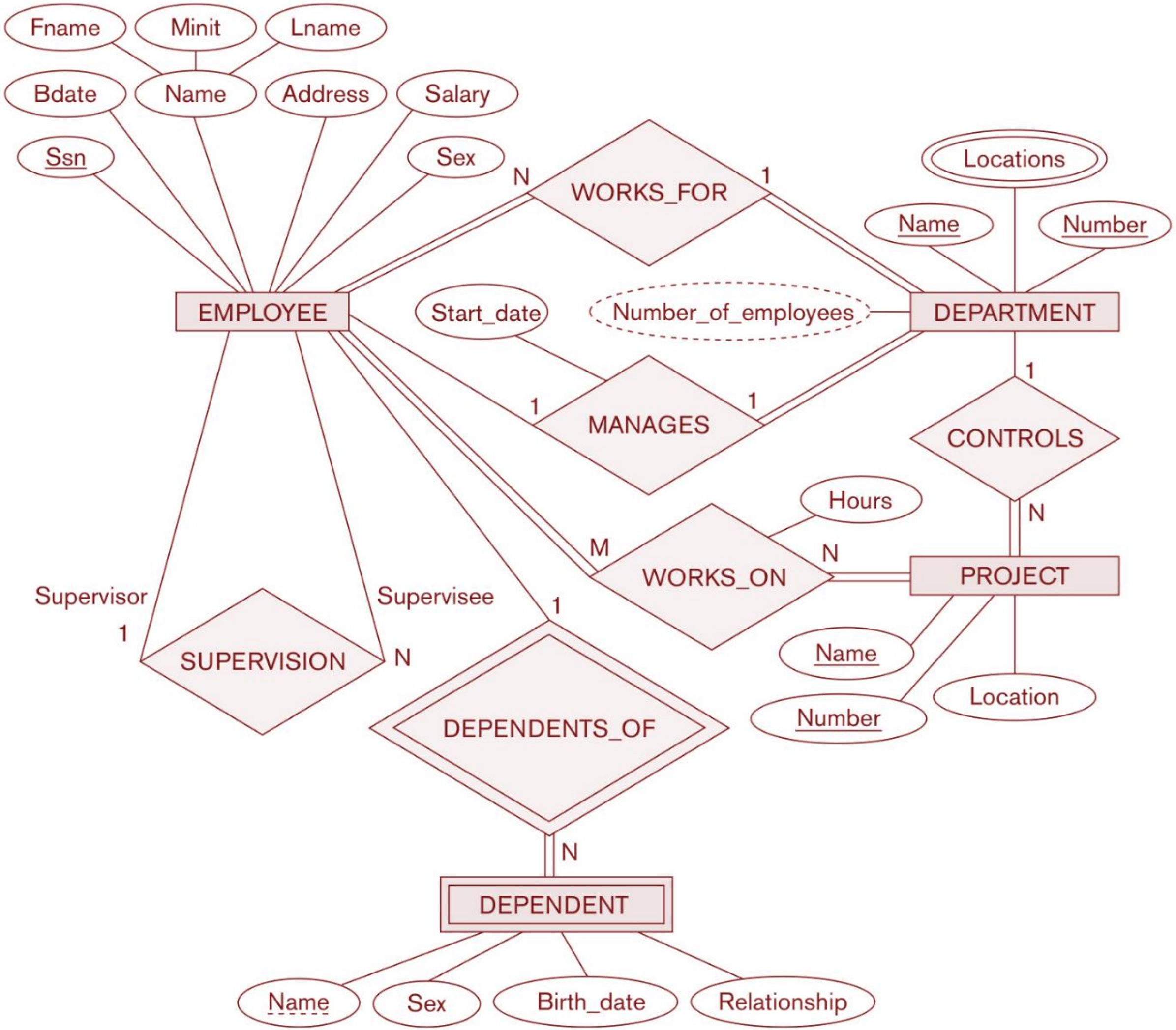
- primary key is the combination of the three foreign keys {SNAME, PARTNO, PROJNAME}

Summary of Mapping constructs and constraints

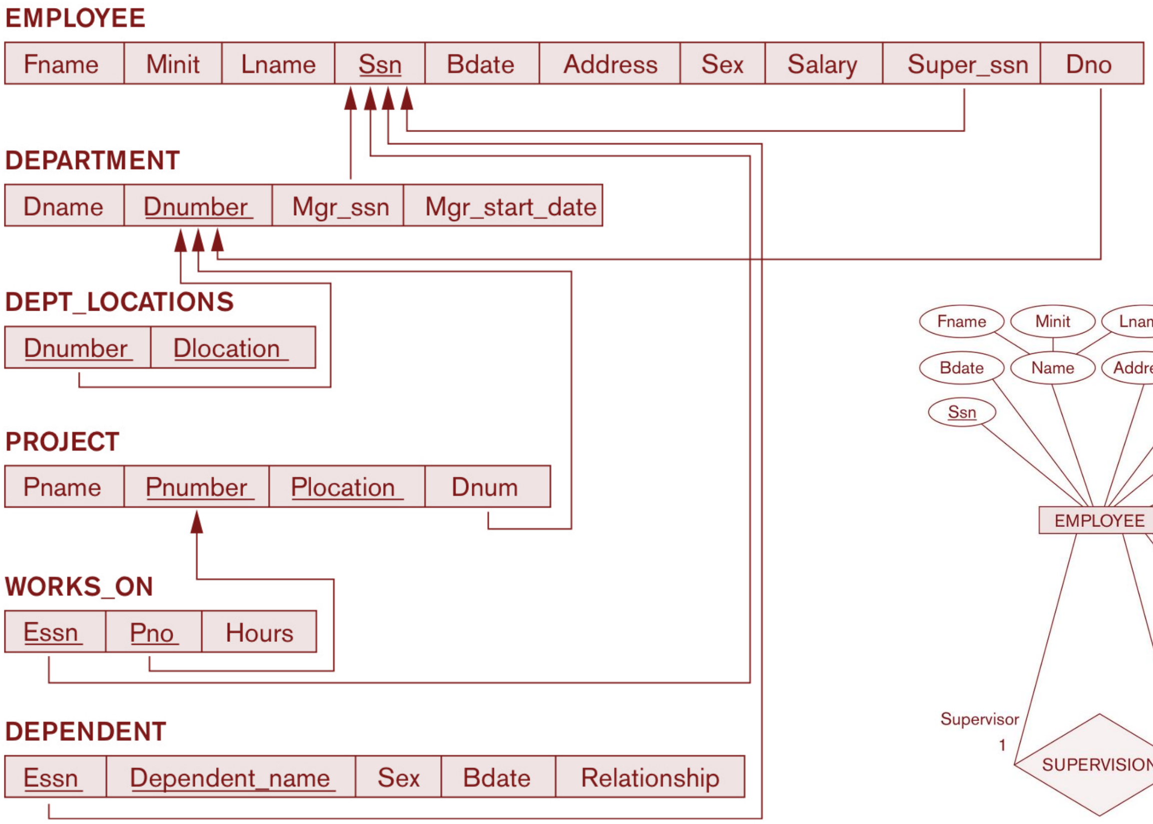


ER MODEL	RELATIONAL MODEL
Entity type	*Entity* relation
1:1 or 1:N relationship type	Foreign key (or *relationship* relation)
M:N relationship type	*Relationship* relation and *two* foreign keys
n-ary relationship type	*Relationship* relation and *n* foreign keys
Simple attribute	Attribute
Composite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

ERD for COMPANY database.



Mapping to Relations



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

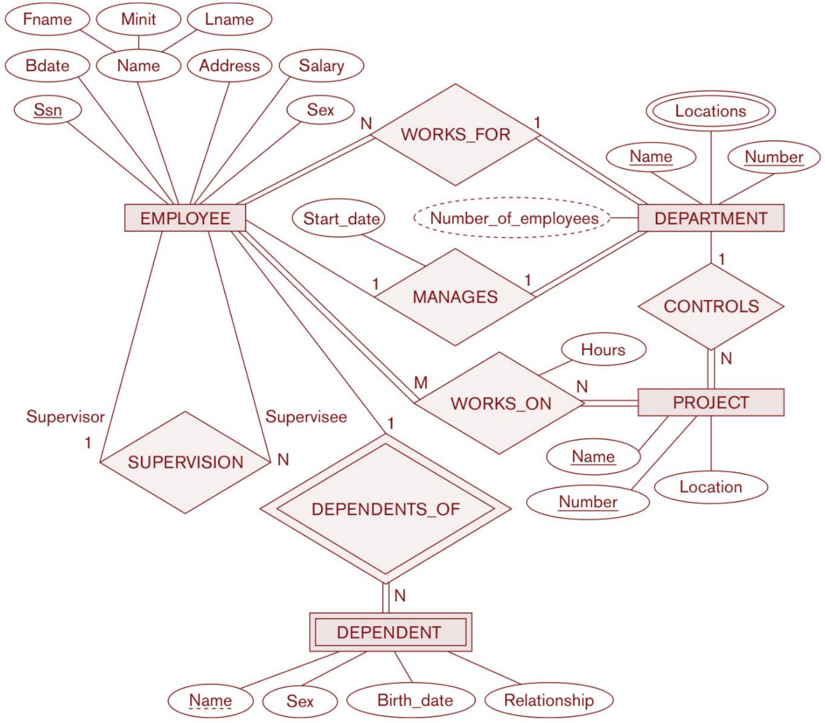
Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship



Mapping to relations (contd.)

Step 1: create the relations EMPLOYEE, DEPARTMENT, and PROJECT in the relational schema corresponding to the regular entities in the ER diagram.

- SSN, DNUMBER, and PNUMBER are the primary keys for the relations EMPLOYEE, DEPARTMENT, and PROJECT

Step 2: create the relation DEPENDENT in this step to correspond to the weak entity type DEPENDENT.

- Include the primary key SSN of the EMPLOYEE relation as a foreign key attribute of DEPENDENT (renamed to ESSN).
- The primary key of the DEPENDENT relation is the combination {ESSN, DEPENDENT_NAME} because DEPENDENT_NAME is the partial key of DEPENDENT.

Step 3: relationship type MANAGES is mapped by choosing the participating entity type DEPARTMENT to serve in the role of S, because its participation in the MANAGES relationship type is total.

Mapping to relations (contd.)

Step 4: 1:N relationship types WORKS_FOR, CONTROLS, and SUPERVISION

- For WORKS_FOR we include the primary key DNUMBER of the DEPARTMENT relation as foreign key in the EMPLOYEE relation and call it DNO.

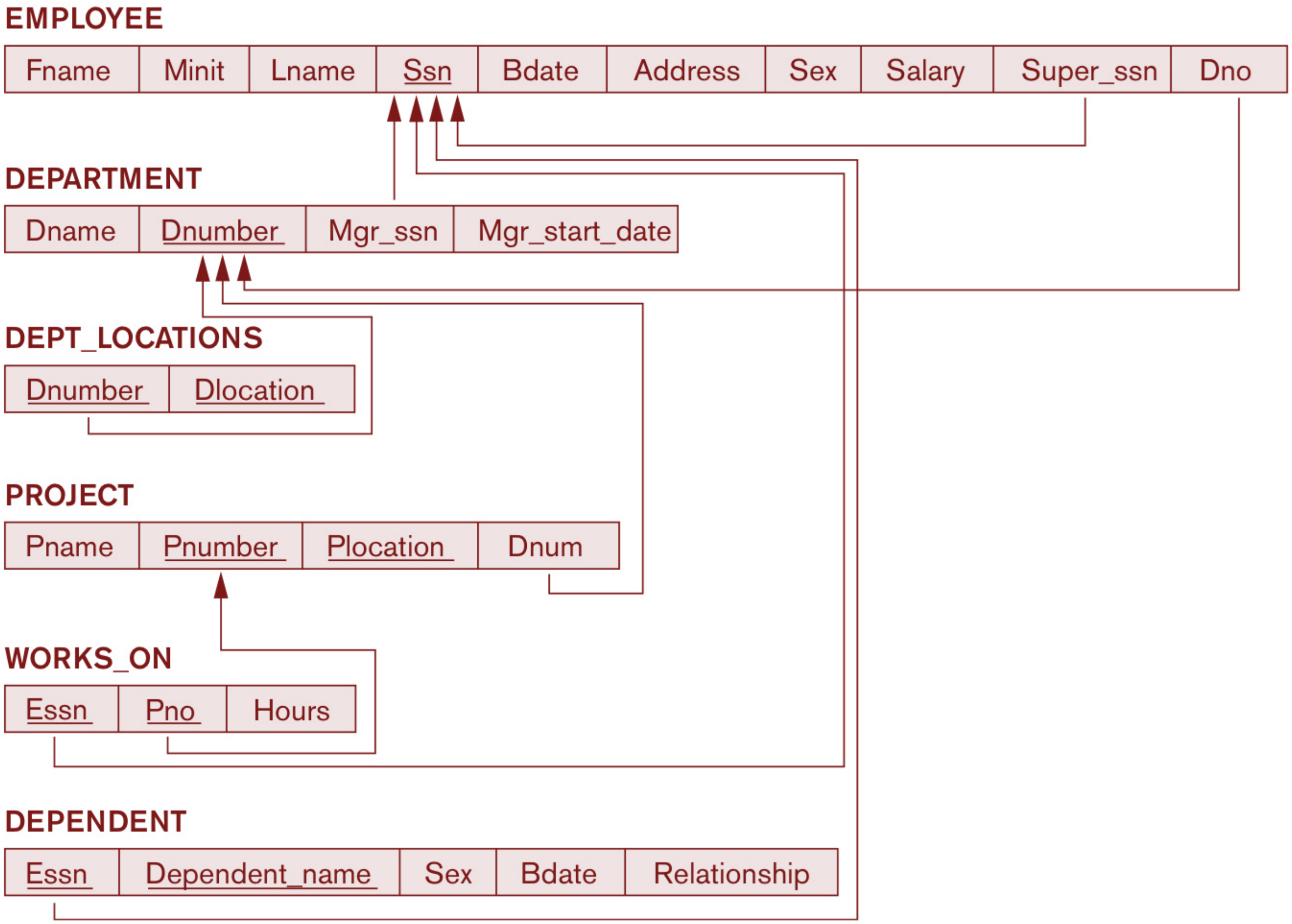
Step 5: The M:N relationship type WORKS_ON from the ER diagram is mapped by creating a relation WORKS_ON in the relational database schema.

- The primary keys of the PROJECT and EMPLOYEE relations are included as foreign keys in WORKS_ON and renamed PNO and ESSN, respectively.
- Attribute HOURS in WORKS_ON represents the HOURS attribute of the relation type. The primary key of the WORKS_ON relation is the combination of the foreign key attributes {ESSN, PNO}.

Step 6: The relation DEPT_LOCATIONS is created.

- The attribute DLOCATION represents the multivalued attribute LOCATIONS of DEPARTMENT, while DNUMBER-as foreign key-represents the primary key of the DEPARTMENT relation.
- The primary key of R is the combination of {DNUMBER, DLOCATION}.

Resulting COMPANY Database Schema



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date

DEPT_LOCATIONS

Dnumber	Dlocation

PROJECT

Pname	Pnumber	Plocation	Dnum

WORKS_ON

Essn	Pno	Hours

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship

Mapping EER Model Constructs to Relations

Step8: Options for Mapping Specialization or Generalization.

- Convert each specialization with m subclasses {S1, S2,….,Sm} and generalized superclass C, where the attributes of C are {k,a1,…an} and k is the (primary) key, into relational schemas using one of the four following options:
  - Multiple relations:
    - Option 8A: Multiple relations-Superclass and subclasses
    - Option 8B: Multiple relations-Subclass relations only
  - Single relation
    - Option 8C: Single relation with one type attribute
    - Option 8D: Single relation with multiple type attributes

Mapping EER Model Constructs to Relations

Option 8A: Multiple relations-Superclass and subclasses

- Create a relation L for C with attributes Attrs(L) = {k,a1,…an} and PK(L) = k. Create a relation Li for each subclass Si, 1 < i < m, with the attributesAttrs(Li) = {k} U {attributes of Si} and PK(Li)=k. This option works for any specialization (total or partial, disjoint of over-lapping).

Option 8B: Multiple relations-Subclass relations only

- Create a relation Li for each subclass Si, 1 < i < m, with the attributes Attr(Li) = {attributes of Si} U {k,a1…,an} and PK(Li) = k. This option only works for a specialization whose subclasses are total (every entity in the superclass must belong to (at least) one of the subclasses).

Mapping EER Model Constructs to Relations (contd.)

Option 8C: Single relation with one type attribute

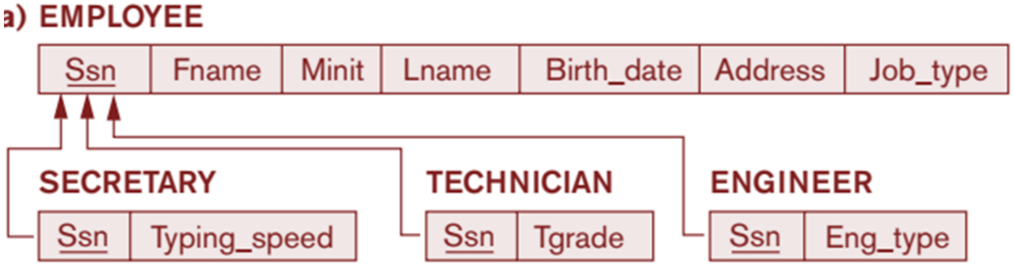
- Create a single relation L with attributes Attrs(L) = $\{k, a_1, \ldots a_n\}$ U {attributes of S_1} U…U {attributes of S_m} U {t} and PK(L) = k. The attribute t is called a type (or **discriminating**) attribute that indicates the subclass to which each tuple belongs

Option 8D: Single relation with multiple type attributes

- Create a single relation schema L with attributes Attrs(L) = $\{k, a_1, \ldots a_n\}$ U {attributes of S_1} U…U {attributes of S_m} U $\{t_1, t_2, \ldots, t_m\}$ and PK(L) = k. Each t_i, 1 < I < m, is a Boolean type attribute indicating whether a tuple belongs to the subclass S_i.

Example 1

Option 8A:



a) EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type

SECRETARY

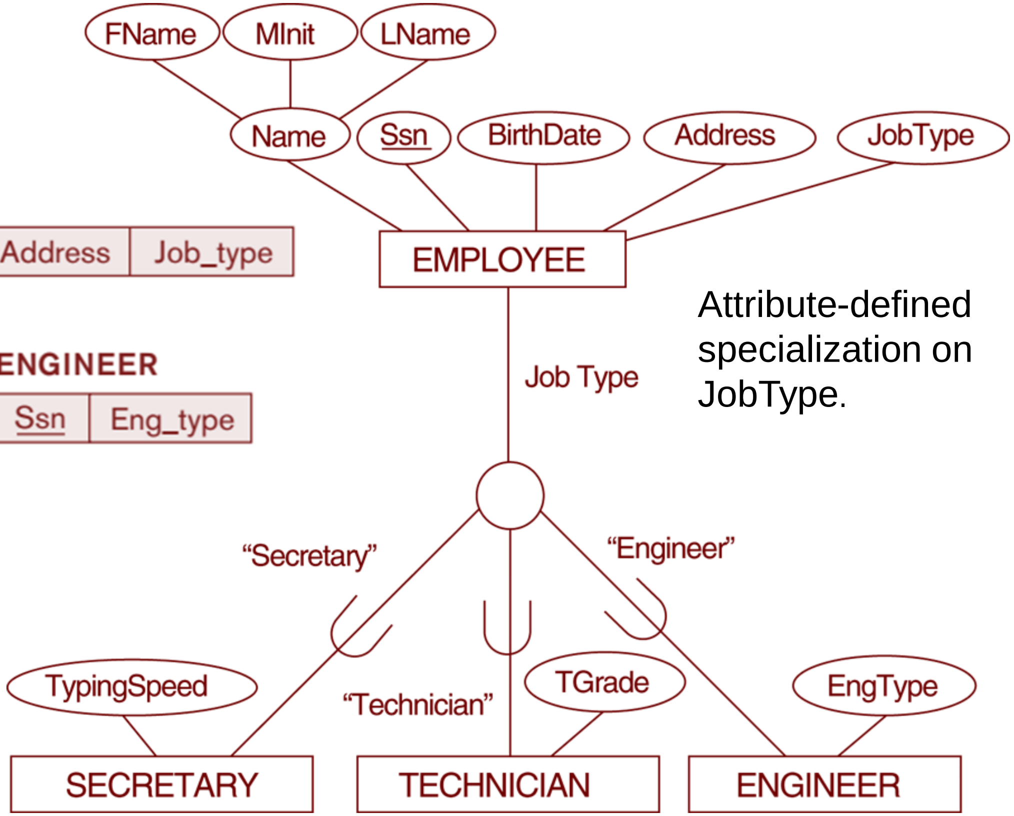
Ssn	Typing_speed

TECHNICIAN

Ssn	Tgrade

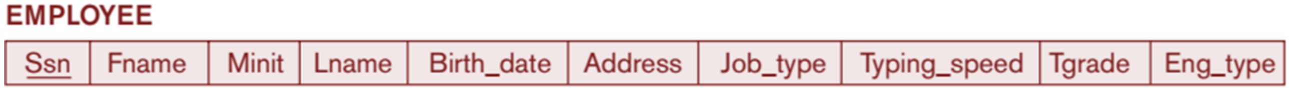
ENGINEER

Ssn	Eng_type



Attribute-defined specialization on JobType.

Option 8C:

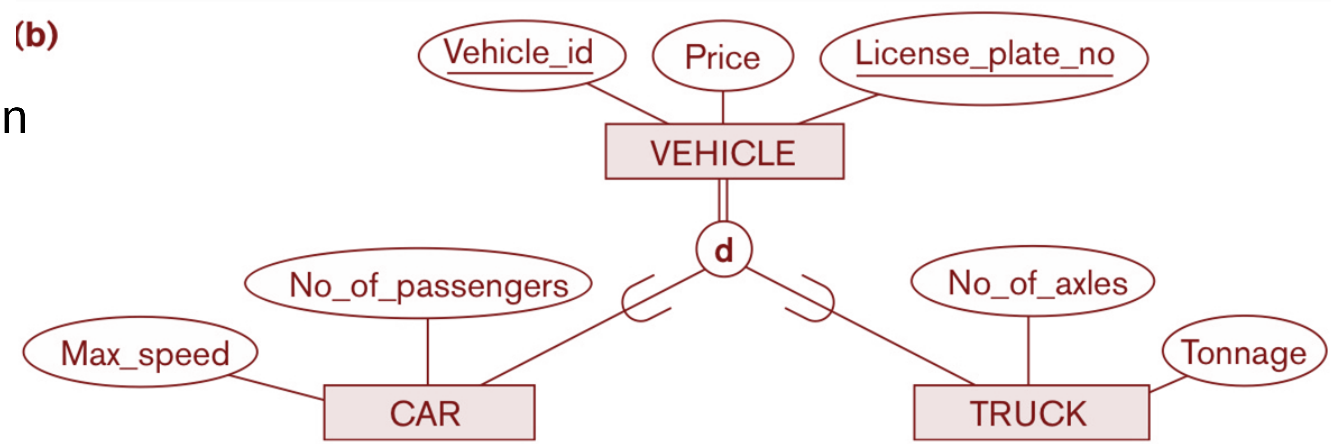


EMPLOYEE

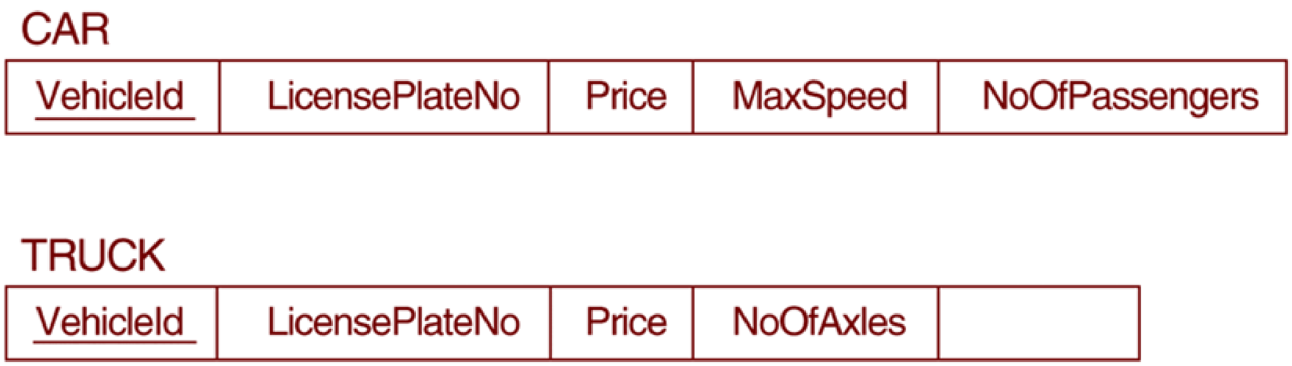
Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

Example 2

Generalization



Option 8B



CAR

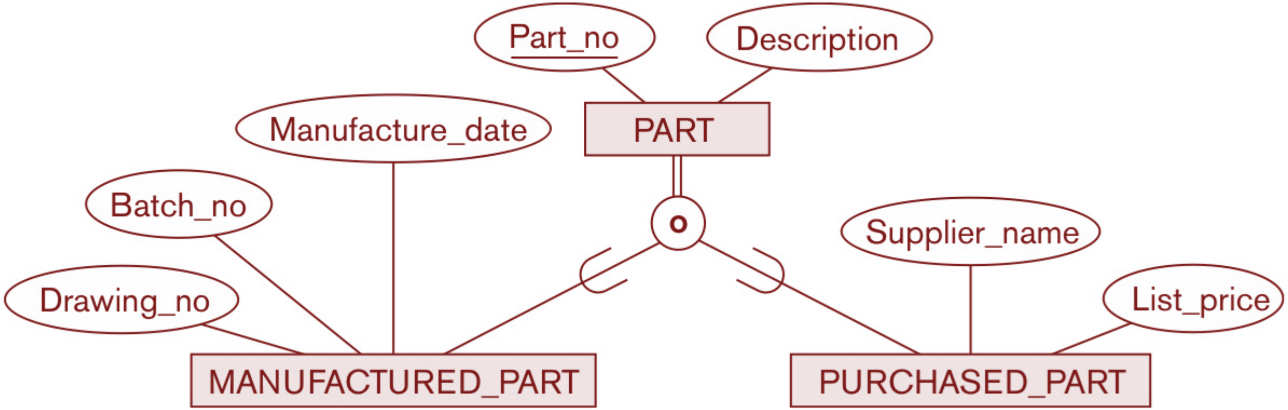
VehicleId	LicensePlateNo	Price	MaxSpeed	NoOfPassengers

TRUCK

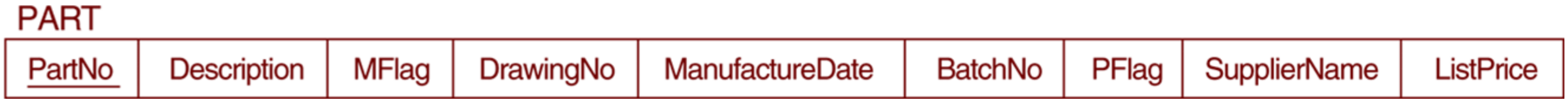
VehicleId	LicensePlateNo	Price	NoOfAxles	

Example 3

Overlapping specialization:



Option 8D with Boolean type fields Mflag and Pflag:



PART

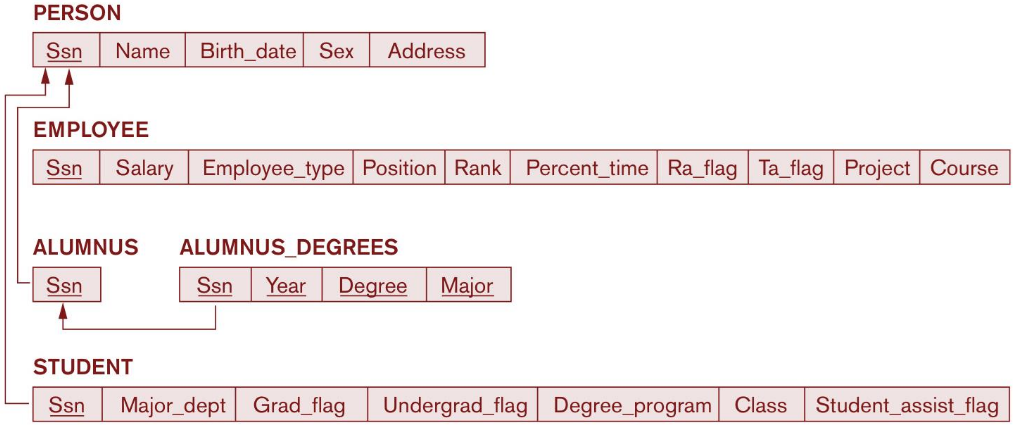
PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

Mapping EER Model Constructs to Relations (contd.)

Mapping of Shared Subclasses (Multiple Inheritance)

- A shared subclass, such as STUDENT_ASSISTANT, is a subclass of several classes, indicating multiple inheritance. These classes must all have the same key attribute; otherwise, the shared subclass would be modeled as a category.
- We can apply any of the options discussed in Step 8 to a shared subclass, subject to the restriction discussed in Step 8 of the mapping algorithm. Below both 8C and 8D are used for the shared class STUDENT_ASSISTANT.

Example



PERSON

Ssn	Name	Birth_date	Sex	Address

EMPLOYEE

Ssn	Salary	Employee_type	Position	Rank	Percent_time	Ra_flag	Ta_flag	Project	Course

ALUMNUS

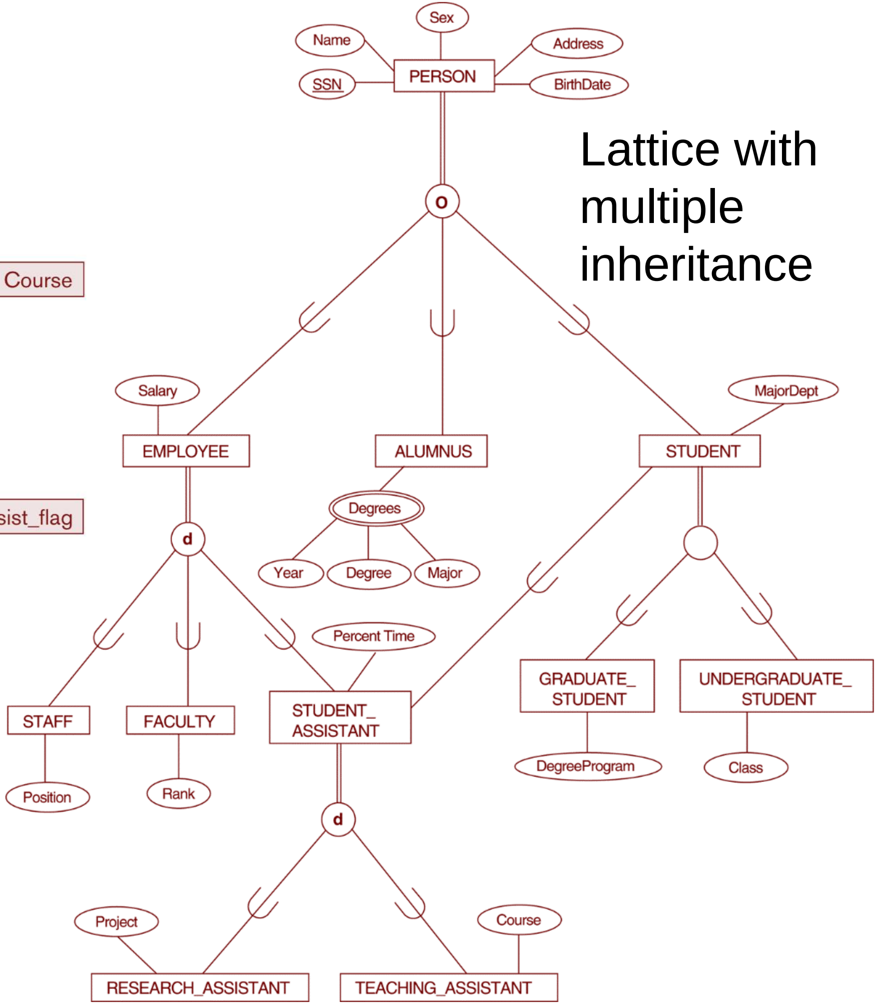
Ssn

ALUMNUS_DEGREES

Ssn	Year	Degree	Major

STUDENT

Ssn	Major_dept	Grad_flag	Undergrad_flag	Degree_program	Class	Student_assist_flag



Lattice with multiple inheritance

STUDENT_ASSISTANT (8C, 8D)

EMPLOYEE (8C)

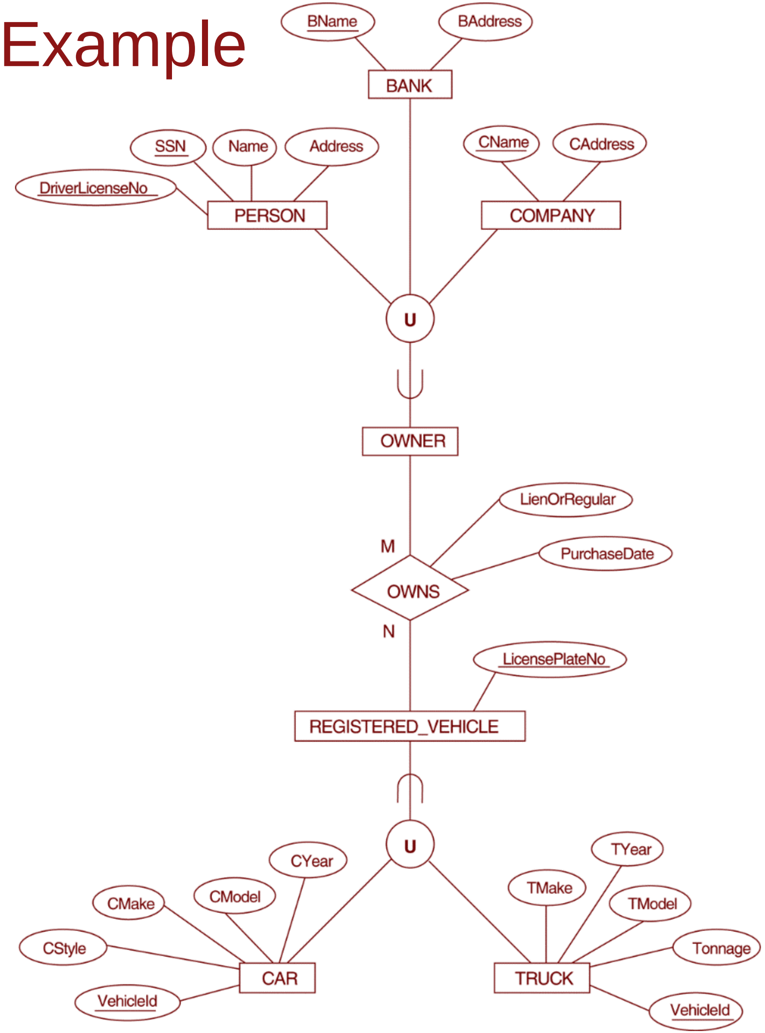
STUDENT (8D)

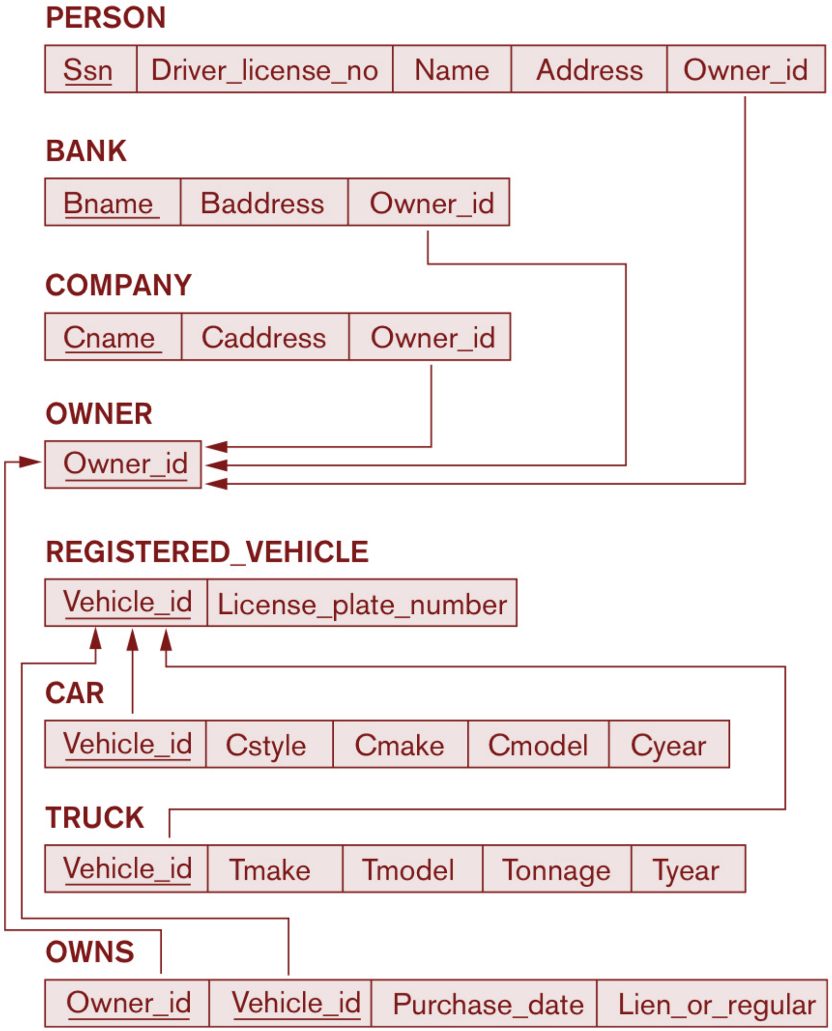
Mapping EER Model Constructs to Relations (contd.)

Step 9: Mapping of Union Types (Categories).

- For mapping a category whose defining superclass have different keys, it is customary to specify a new key attribute, called a surrogate key, when creating a relation to correspond to the category.
- In the example below we can create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called OwnerId.

Example





PERSON

Ssn	Driver_license_no	Name	Address	Owner_id

BANK

Bname	Baddress	Owner_id

COMPANY

Cname	Caddress	Owner_id

OWNER

Owner_id

REGISTERED_VEHICLE

Vehicle_id	License_plate_number

CAR

Vehicle_id	Cstyle	Cmake	Cmodel	Cyear

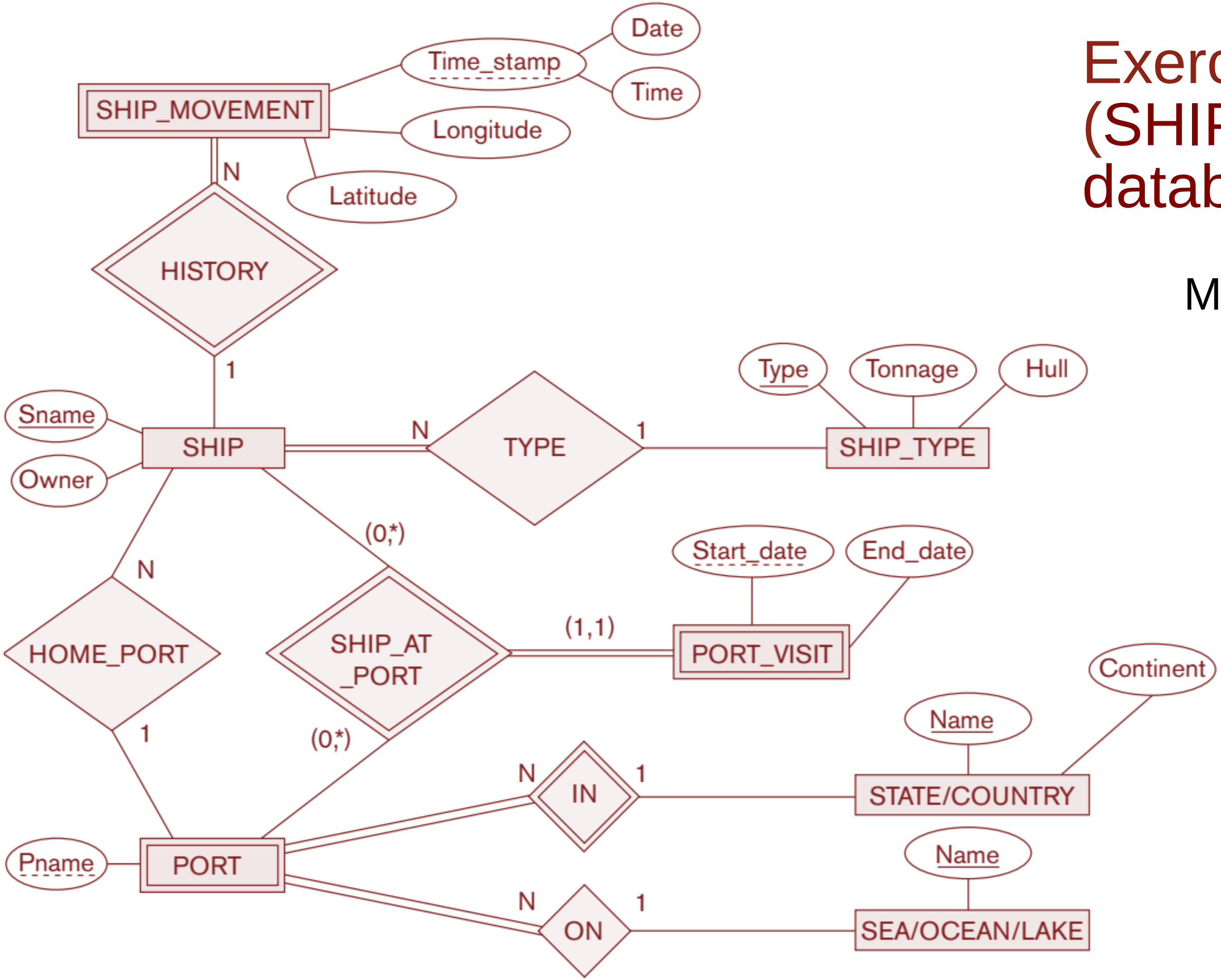
TRUCK

Vehicle_id	Tmake	Tmodel	Tonnage	Tyear

OWNS

Owner_id	Vehicle_id	Purchase_date	Lien_or_regular

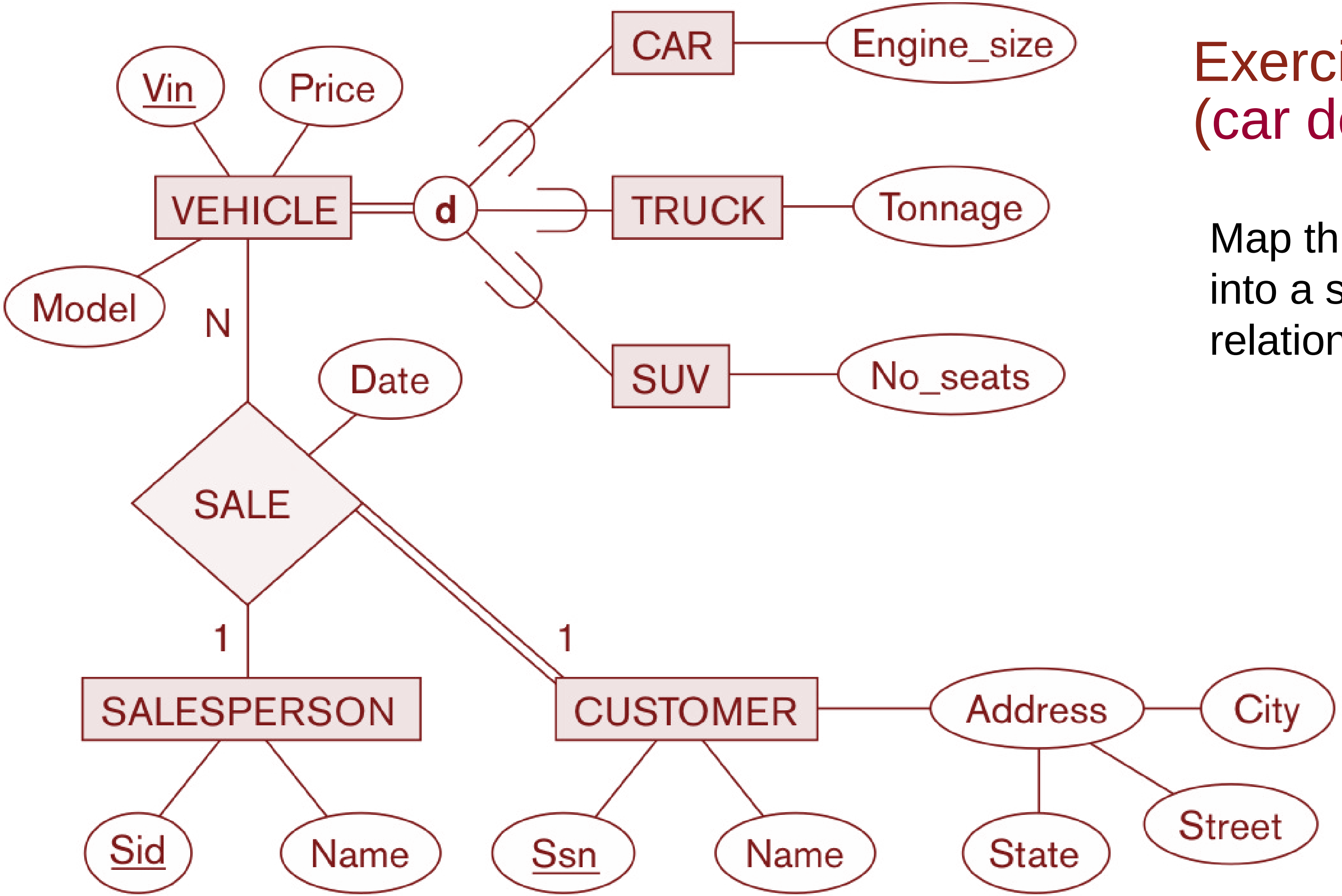
Exercise-1 (SHIP_TRACKING database)



Map this schema into

a set of relations.

Exercise-2 (car dealer)



Map this schema into a set of relations