



# ER-to-Relational Mapping



## Outline

- 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 Relationship 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

•





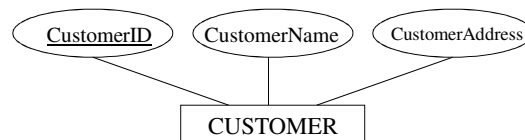
## Step 1: Mapping of Regular Entity Types

- 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.



## Step 1: Mapping of Regular Entity Types

- CUSTOMER entity type with simple attributes:



- CUSTOMER relation:

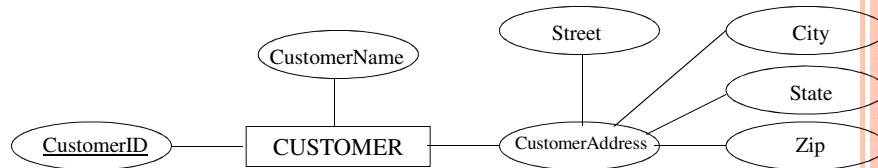
### CUSTOMER

<u>CustomerID</u>	CustomerName	CustomerAddress
-------------------	--------------	-----------------



## Step 1: Mapping of Regular Entity Types

- CUSTOMER entity type with a composite attribute:



- CUSTOMER relation with address detail:

### CUSTOMER

<u>CustomerID</u>	CustomerName	Street	City	State	Zip
-------------------	--------------	--------	------	-------	-----



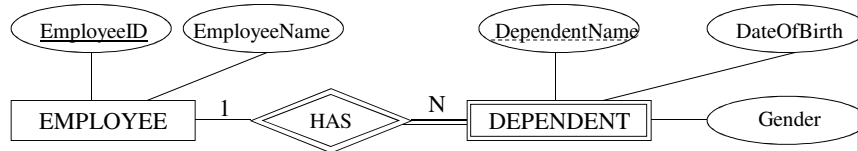
## Step 2: Mapping of Weak Entity Types

- For each weak entity type W in the ER schema with owner entity type E, create a relation R and include all simple attributes of W as attributes of R.
- 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 composed of:
  - Partial key of the weak entity type W.
  - Primary key(s) of the owner(s).

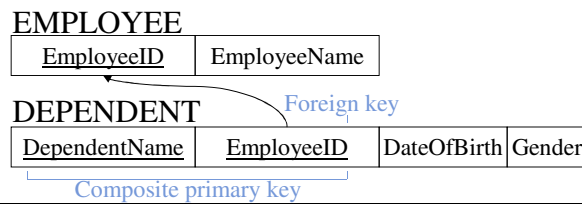


## Step 2: Mapping Weak Entities to Relations

- Weak entity DEPENDENT:



- Relations:



Chapter 6: Relational Database Design by ER- and EER-to-Relational Mapping

7



## Step 3: Mapping OF Binary 1:1 Relationship Types

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

Chapter 6: Relational Database Design by ER- and EER-to-Relational Mapping

8



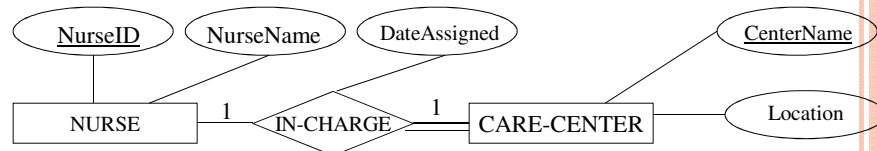
### Step 3: Mapping OF Binary 1:1 Relationship Types

- There are three possible approaches:
  - **Foreign key approach:**
    - Choose one of the relations - S, say - 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 the 1:1 relationship type R as attributes of S.
  - **Merged relation option:**
    - Merging the two entity types and the relationship into a single relation. (Used when both participations are total)
  - **Cross-reference or relationship relation option:**
    - Setting 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.

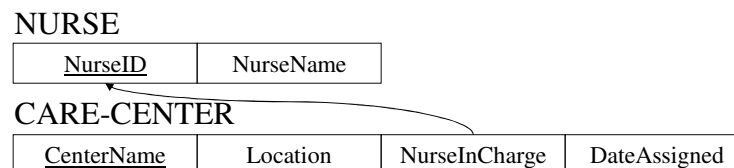


### Step 3: Mapping OF Binary 1:1 Relationship Types

- one-to-one relationship:



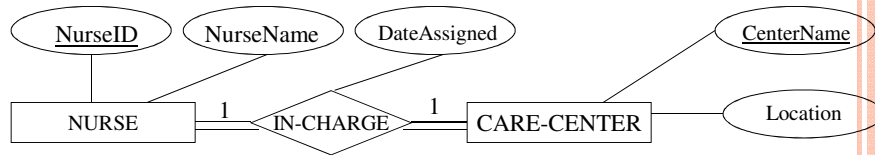
- Mapping the relationship (**foreign key approach**):





### Step 3: Mapping OF Binary 1:1 Relationship Types

- one-to-one relationship:



- Mapping the relationship (**Merged relation option**):

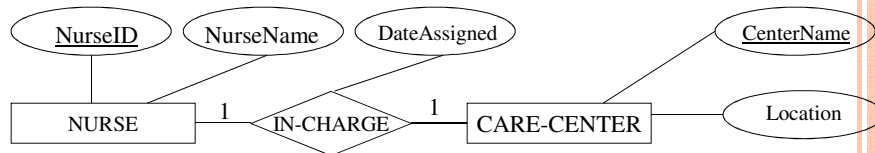
NURSE

<u>NurseID</u>	NurseName	CenterName	Location	DateAssigned
----------------	-----------	------------	----------	--------------



### Step 3: Mapping OF Binary 1:1 Relationship Types

- one-to-one relationship:



- Mapping the relationship (**Cross-reference option**):

NURSE

<u>NurseID</u>	NurseName
----------------	-----------

CARE-CENTER

<u>CenterName</u>	Location
-------------------	----------

NURSE-CARE-CENTER

<u>NurseID</u>	<u>CenterName</u>	DateAssigned
----------------	-------------------	--------------



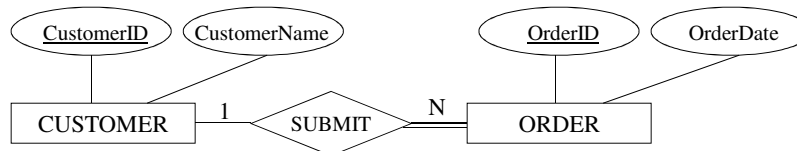
## Step 4: Mapping of Binary 1:N Relationship Types

- For each regular binary 1:N relationship type R, identify the relation S that represents 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 attribute of the 1:N relationship type as attributes of S.

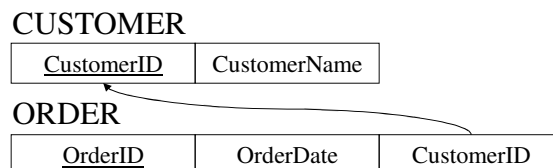


## Step 4: Mapping of Binary 1:N Relationship Types

- One-to-many relationship:



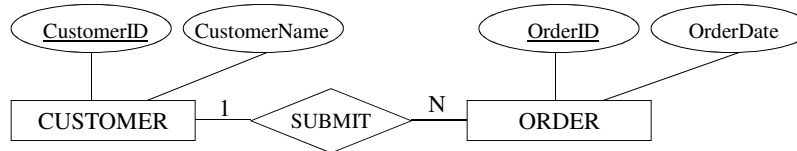
- Mapping the relationship (**foreign key approach**):



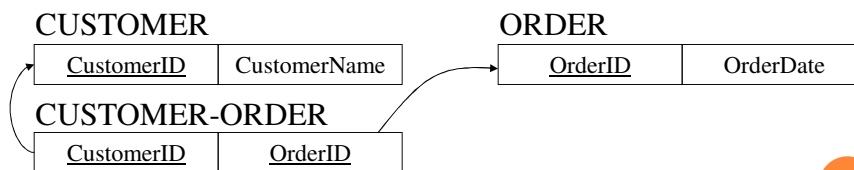


## Step 4: Mapping of Binary 1:N Relationship Types

- One-to-many relationship:



- Mapping the relationship (**Cross-reference option**):



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

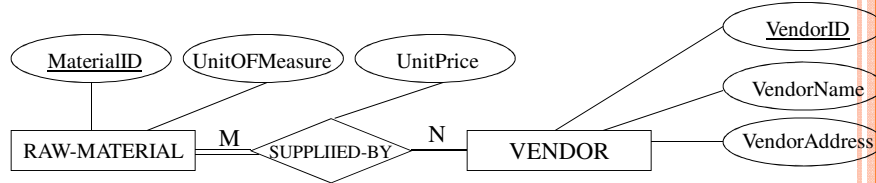
- For each binary M:N relationship type R, create a new relation S to represent R.
- Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
- Include any simple attributes of the M:N relationship type as attributes of S.



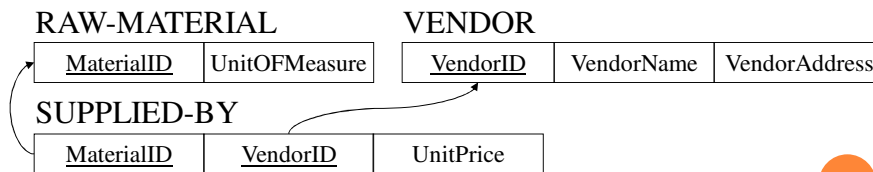


## Step 5: Mapping of Binary M:N Relationship Types

- Many-to-many relationship:



- Mapping the relationship:



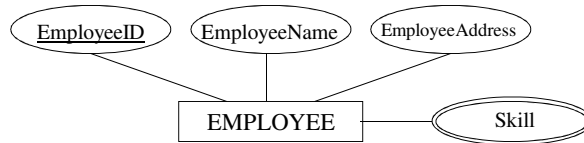
## Step 6: Mapping of Multivalued Attributes

- 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 or 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, include its simple components.

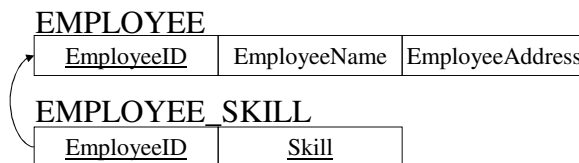


## Step 6: Mapping of Multivalued Attributes

- EMPLOYEE entity type with a multivalued attribute:



- Multivalued attribute becomes a separate relation with a foreign key:



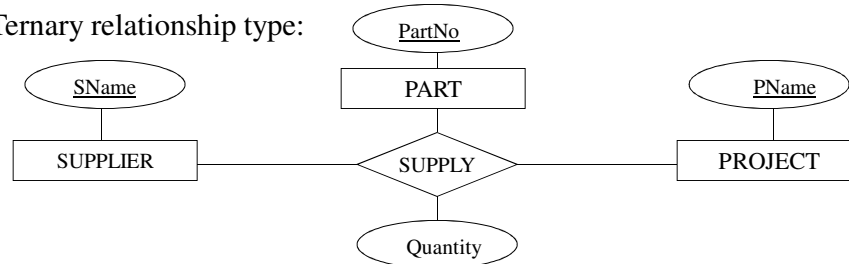
## Step 7: Mapping of N-ary Relationship Types

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

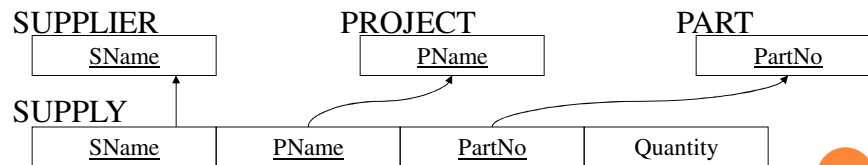


## Step 7: Mapping of N-ary Relationship Types

- Ternary relationship type:

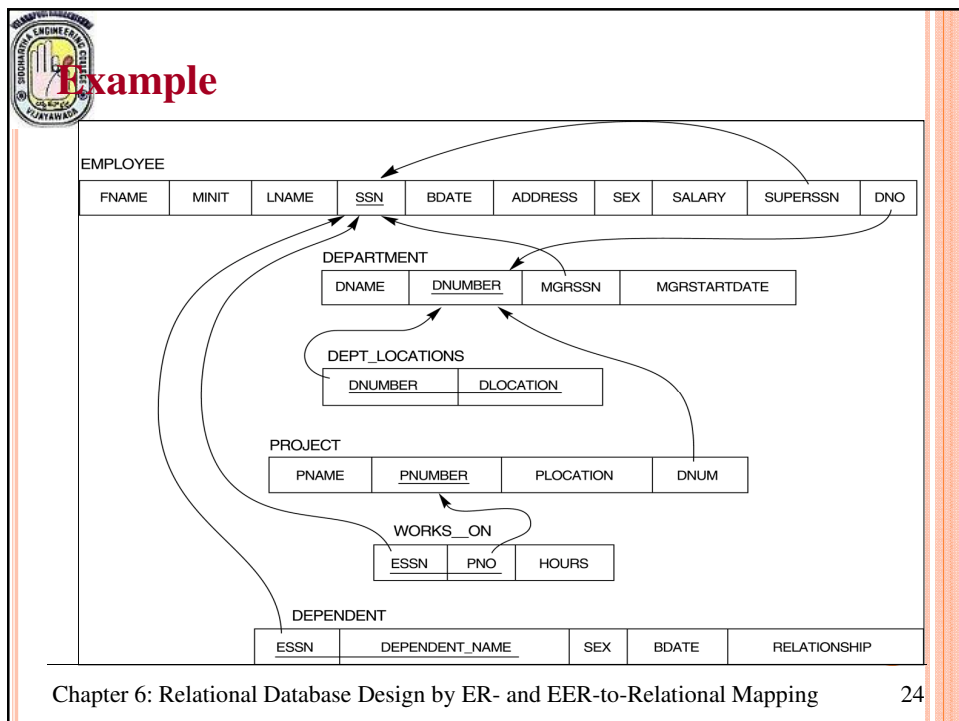
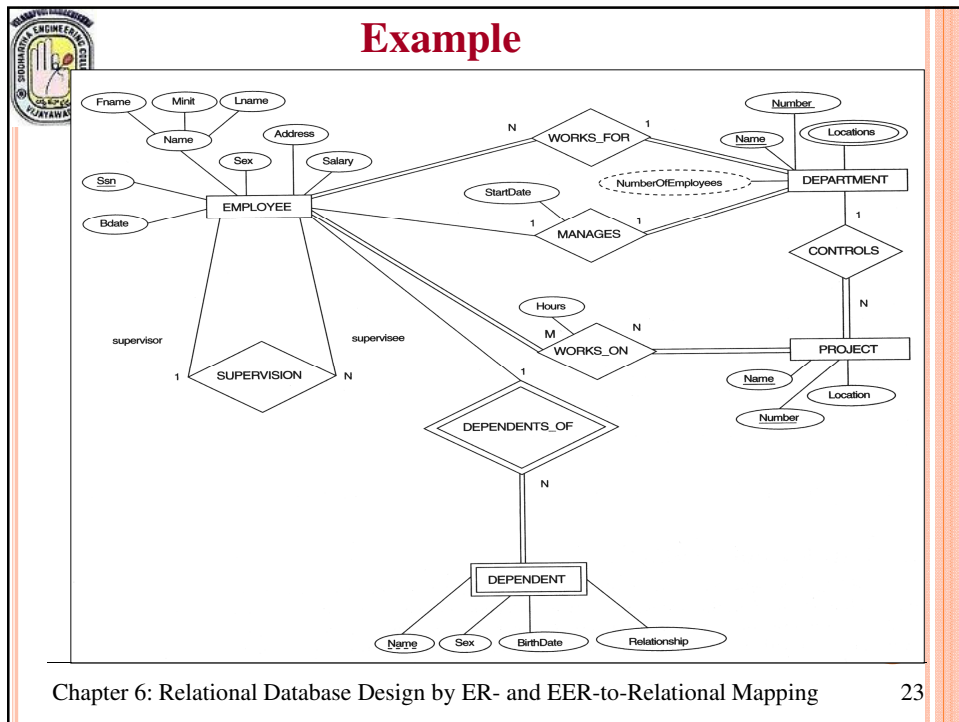


- Mapping the n-ary relationship type SUPPLY:



## Correspondence Between ER and Relational Models

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





## Step 8: Options for Mapping Specialization or Generalization

### Option 8A: Multiple Relations - Superclass and Subclasses

This option works for any specialization.

#### SECRETARY

<u>SSN</u>	TypingSpeed
------------	-------------

#### TECHNICIAN

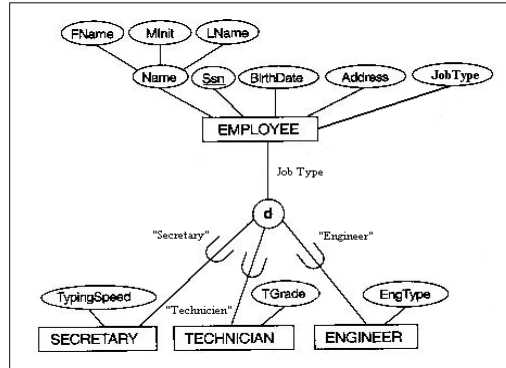
<u>SSN</u>	TGrade
------------	--------

#### ENGINEER

<u>SSN</u>	EngType
------------	---------

#### EMPLOYEE

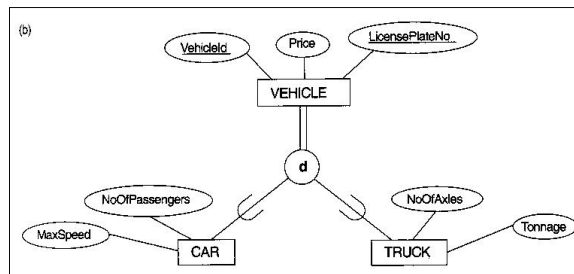
<u>SSN</u>	FName	Minit	LName	BirthDate	Address	JobType
------------	-------	-------	-------	-----------	---------	---------



## Step 8: Options for Mapping Specialization or Generalization

### Option 8B: Multiple Relations - Subclass Relations Only

This option only works for a specialization whose subclasses are total.



#### CAR

<u>VehicleID</u>	LicensePlateNo	Price	MaxSpeed	NoOfPassengers
------------------	----------------	-------	----------	----------------

#### TRUCK

<u>VehicleID</u>	LicensePlateNo	Price	NoOfAxes	Tonnage
------------------	----------------	-------	----------	---------

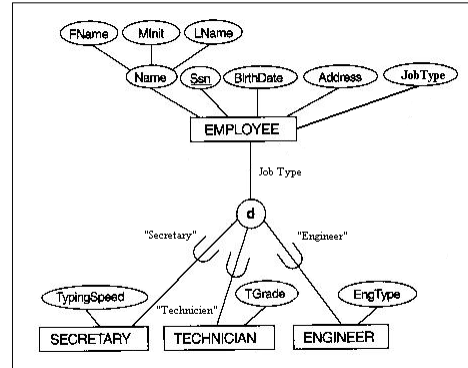


## Step 8: Options for Mapping Specialization or Generalization

### Option 8C: Single Relation With One Type Attribute

This option:

- Works only for a specialization whose subclasses are disjoint.
- Has the potential for generating many null values if many specific attributes exist in the subclasses.



EMPLOYEE

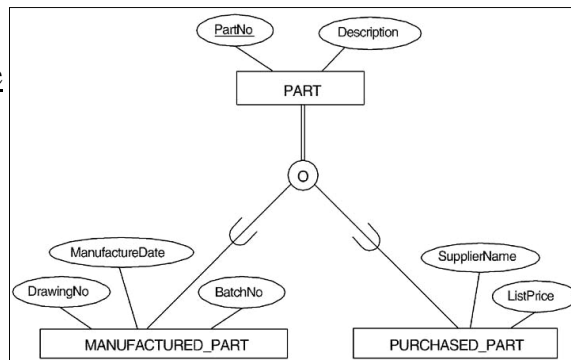
<u>SSN</u>	FName	Minit	LName	BirthDate	Address	JobType	TypingSpeed	TGrade	EngType
------------	-------	-------	-------	-----------	---------	---------	-------------	--------	---------



## Step 8: Options for Mapping Specialization or Generalization

### Option 8D: Single Relation With Multiple Type Attributes

This option works for a specialization whose subclasses are overlapping (It also works for a disjoint specialization).



PART

<u>PartNo</u>	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice
---------------	-------------	-------	-----------	-----------------	---------	-------	--------------	-----------



## Step 8: Options for Mapping Specialization or Generalization

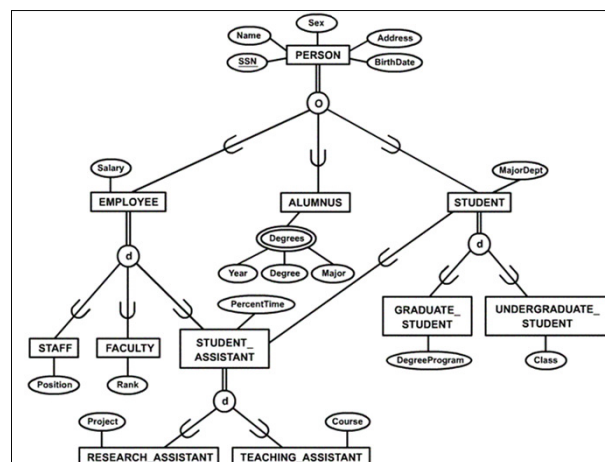
### Mapping of Shared Subclasses (Multiple Inheritance)

- Shared subclasses must have the same key attribute; otherwise the shared subclass would be modeled as a category.
- Any of the options discussed in step 8 can be applied to a shared subclass.



## Step 8: Options for Mapping Specialization or Generalization

### Mapping of Shared Subclasses (Multiple Inheritance)





## Step 8: Options for Mapping Specialization or Generalization

### Mapping of Shared Subclasses (Multiple Inheritance)

#### PERSON

<u>SSN</u>	Name	BirthDate	Sex	Address
------------	------	-----------	-----	---------

#### EMPLOYEE

<u>SSN</u>	Salary	EmployeeType	Position	Rank	PercentTime	RAFlag	TAFlag	Project	Course
------------	--------	--------------	----------	------	-------------	--------	--------	---------	--------

#### STUDENT

<u>SSN</u>	MajorDept	GradFlag	UndergradFlag	DegreeProgram	Class	StudAssistFlag
------------	-----------	----------	---------------	---------------	-------	----------------

#### ALUMNUS

<u>SSN</u>
------------

#### ALUMNUS-DEGREE

<u>SSN</u>	<u>Year</u>	<u>Degree</u>	<u>Major</u>
------------	-------------	---------------	--------------



## Step 9: Mapping of Union Types (Categories)

- For mapping a category whose defining subclasses 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.





## Step 9: Mapping of Union Types (Categories)



## Step 9: Mapping of Union Types (Categories)

### PERSON

<u>SSN</u>	DriverLicenseNo	Name	Address	OwnerID
------------	-----------------	------	---------	---------

### BANK

<u>BName</u>	BAddress	OwnerID
--------------	----------	---------

### COMPANY

<u>CName</u>	CAddress	OwnerID
--------------	----------	---------

### OWNER

<u>OwnerID</u>
----------------

### REGISTERED-VEHICLE

<u>VehicleID</u>	LicensePlateNumber
------------------	--------------------

### CAR

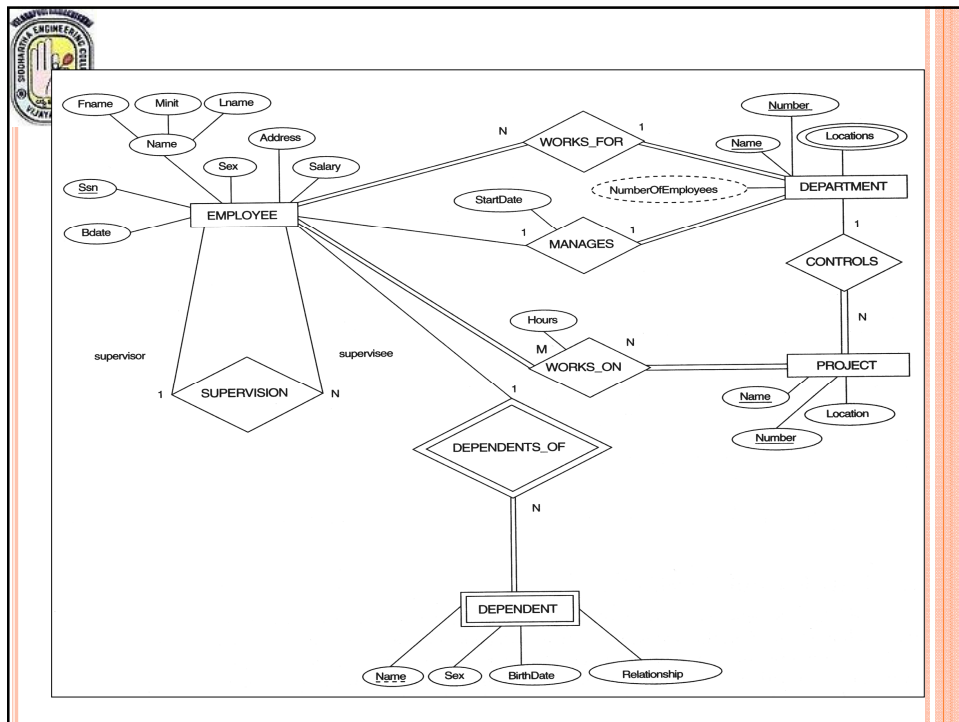
<u>VehicleID</u>	CStyle	CMake	CModel	CYear
------------------	--------	-------	--------	-------

### TRUCK

<u>VehicleID</u>	TStyle	TMake	Tonnage	TYear
------------------	--------	-------	---------	-------

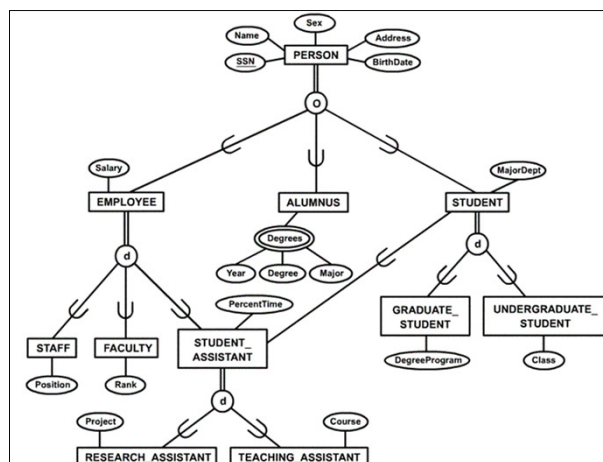
### OWNS

<u>OwnerID</u>	<u>VehicleID</u>	PurchaseDate	LienOrRegular
----------------	------------------	--------------	---------------



## Step 8: Options for Mapping Specialization or Generalization

### Mapping of Shared Subclasses (Multiple Inheritance)





## Step 9: Mapping of Union Types (Categories)

