

Projeto Lógico

Definição e Passos

- Consiste do mapeamento do Diagrama ER (ou EER) para o modelo relacional
- Existem alguns passos que podem ser utilizados no mapeamento
 - Mapeamento de Entidades Regulares
 - Mapeamento de Entidades Fracas
 - Mapeamento de atributos multivalorados
 - Mapeamento de relações binárias 1:1
 - Mapeamento de relações binárias 1:N
 - Mapeamento de relações binárias M:N
 - Mapeamento de relações N-árias

Projeto Lógico

Mapeamento Entidades Regulares

- ❑ Para cada entidade E regular (forte) em um diagrama ER uma relação R é criada
- ❑ Relação contém todos os atributos simples de E
- ❑ Para atributos compostos, inclua cada atributo como um atributo da relação R
- ❑ Escolha um dos atributos chave de E como a chave primária
- ❑ Caso a chave primária seja composta mais de um atributo de R fará parte da chave primária

Projeto Lógico

Mapeamento Entidades Fracas

- ❑ Para cada entidade fraca W presente no diagrama ER que possua como proprietário uma entidade regular E , crie uma relação RW
- ❑ Esta relação conterá todos os atributos simples e compostos de W
- ❑ Inclua em RW uma chave estrangeira que conterá como atributos a chave primária de R
- ❑ A chave primária de RW será a combinação da chave primária de R e possíveis chaves parciais presentes em W

Projeto Lógico

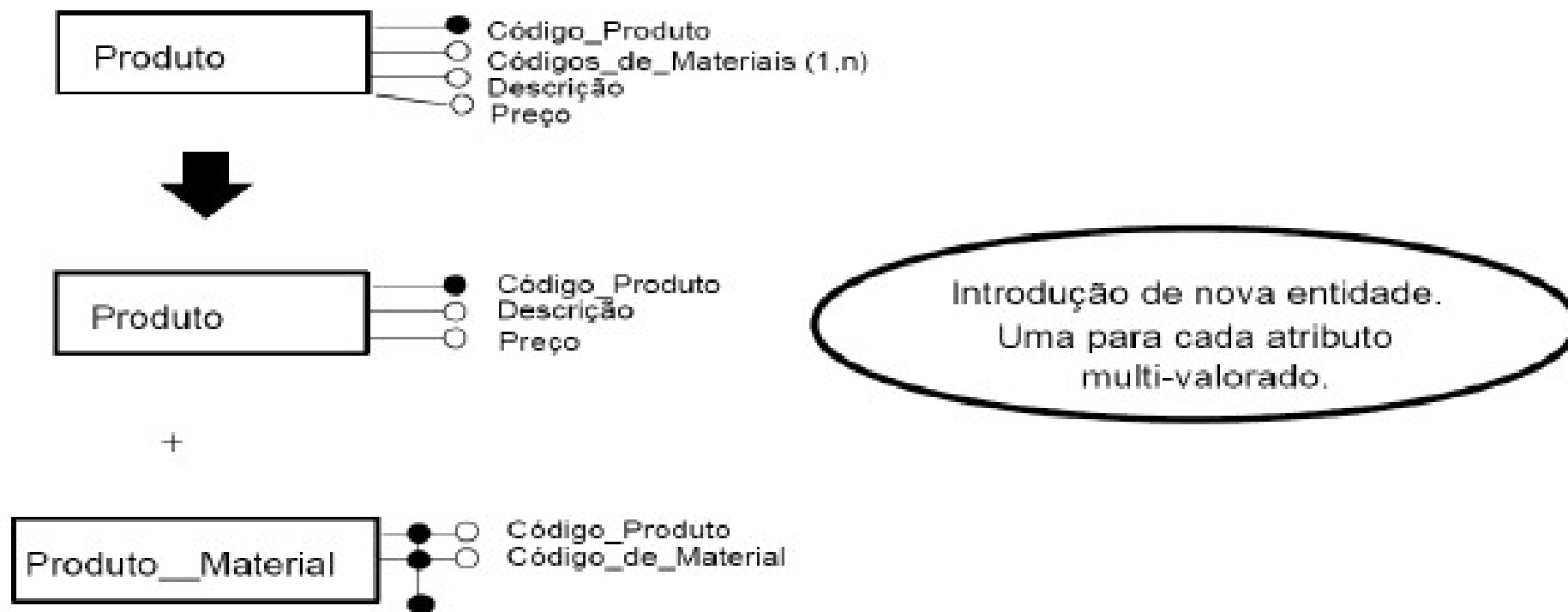
Mapeamento Atributos Multivalorados

- ❑ Para cada atributo multivalorado A , crie uma nova relação R
- ❑ A relação incluirá todos os atributos de A , mais uma chave estrangeira K , que contém a chave primária da entidade que contém o atributo multivalorado
- ❑ A chave primária de R será a combinação de A e K

Projeto Lógico

Mapeamento Atributos Multivalorados

- Cada atributo como uma entidade



Projeto Lógico

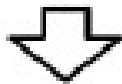
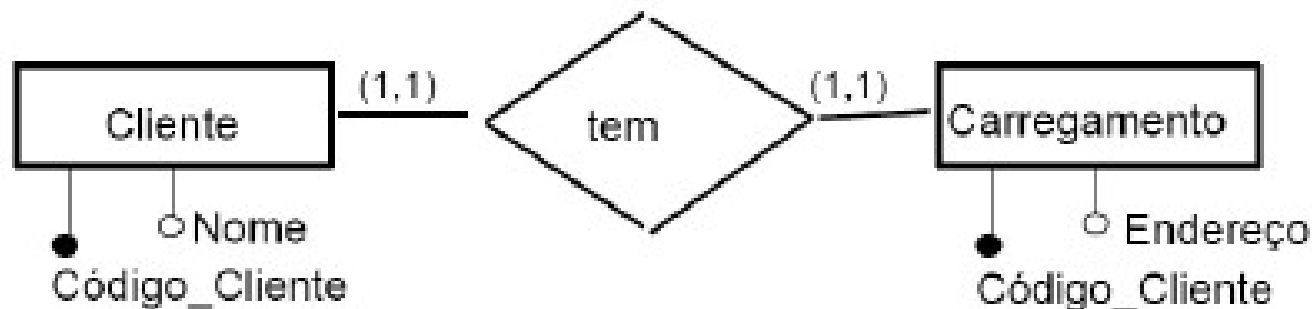
Mapeamento de Relações Binárias 1:1

- ❑ Para cada relação binária 1:1 existente no diagrama ER identifique as relações S e T que correspondem às entidades participantes do relacionamento
- ❑ Escolha uma relação, por exemplo, S e inclua uma chave primária de T
 - O ideal é escolher uma entidade com participação total no relacionamento
- ❑ Inclua atributos simples e/ou compostos na relação

Projeto Lógico

Mapeamento de Relações Binárias 1:1

- Tradição de relacionamentos 1:1 com mesma chave



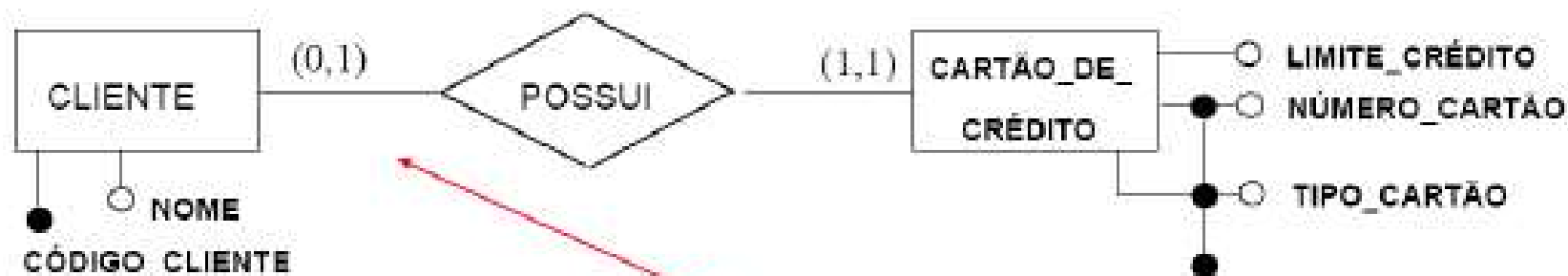
Agrupando numa mesma relação

CLIENTE_CARREGAMENTO (Código_Cliente, Nome, Endereço)

Projeto Lógico

Mapeamento de Relações Binárias 1:1

- Tradição de relacionamentos 1:1 com chave diferente



Agrupando em relações distintas

CLIENTE (CÓDIGO CLIENTE, NOME)

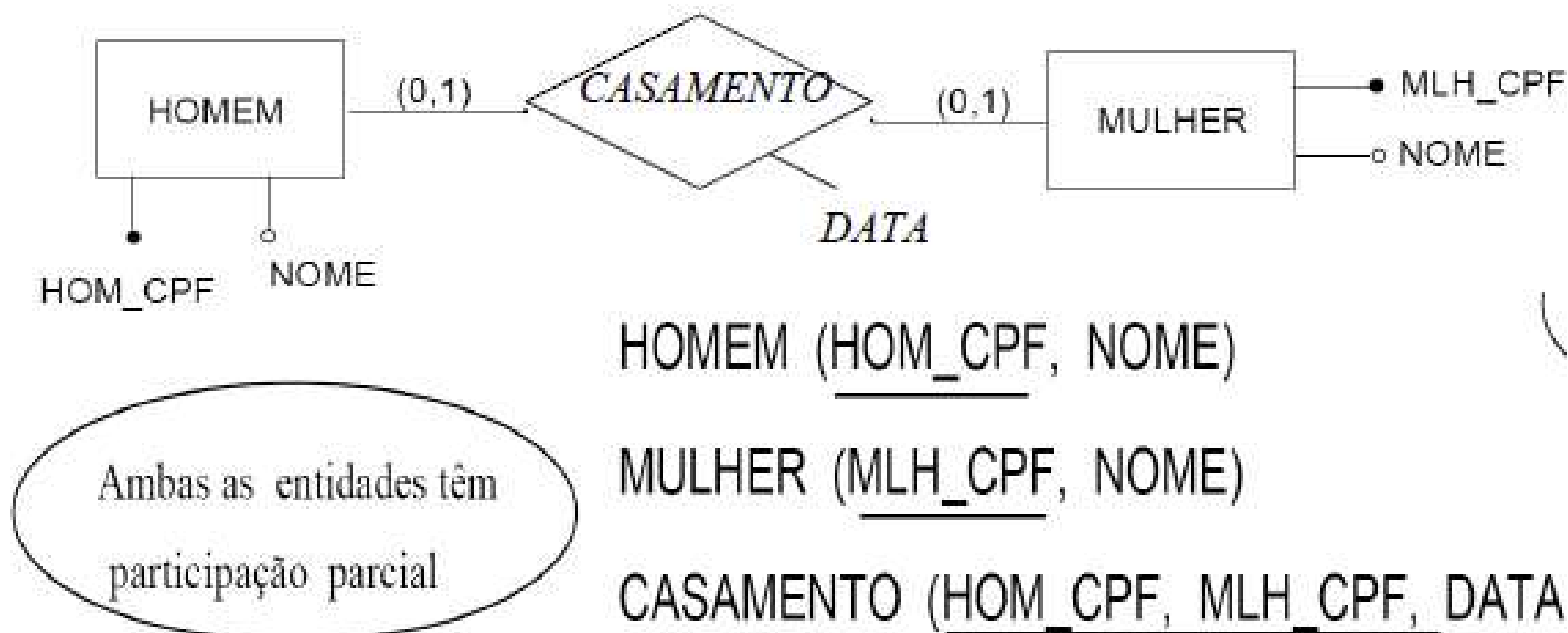
CARTAO_DE_CREDITO (TIPO_CARTAO, NUMERO_CARTAO,
CODIGO_CLIENTE, LIMITE)

Uma das entidades tem
participação parcial

Projeto Lógico

Mapeamento de Relações Binárias 1:1

- Tradição de relacionamentos 1:1 com chave diferente



Projeto Lógico

Mapeamento de Relações Binárias 1:N

- ❑ Para cada relação binária 1:N R, identifique a relação S que participa no lado N da relação
- ❑ Inclua como chave estrangeira em S a chave primária da relação T, que representa a outra entidade participante de R
- ❑ Inclua atributos simples da relação como atributos de S

Projeto Lógico

Mapeamento de Relações Binárias 1:N

- Tradição de relacionamentos 1:N



CIDADE (NOME_CIDADE, NOME_ESTADO, POPULAÇÃO)

ESTADO (NOME_ESTADO, GOVERNADOR, POPULAÇÃO)

Projeto Lógico

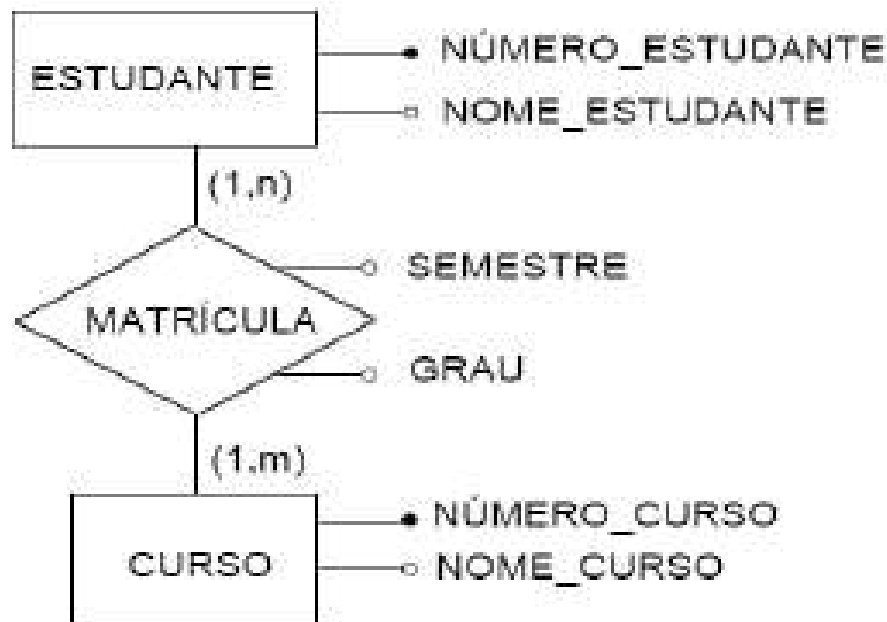
Mapeamento de Relações Binárias M:N

- ❑ Para cada relacionamento binário M:N, será criada uma nova relação S
- ❑ A relação S conterá como chave estrangeiras as chaves primárias das entidades participantes da relação
- ❑ Sua combinação será a chave primária de S
- ❑ Todos os atributos do relacionamento serão atributos de S

Projeto Lógico

Mapeamento de Relações Binárias M:N

Relações Binária M:N



Independente
das
cardinalidades

ESTUDANTE (NÚMERO_ESTUDANTE, NOME_ESTUDANTE)

CURSO (NÚMERO_CURSO, NOME_CURSO)

MATRÍCULA (NÚMERO_ESTUDANTE, NÚMERO_CURSO, SEMESTRE, GRAU)

Projeto Lógico

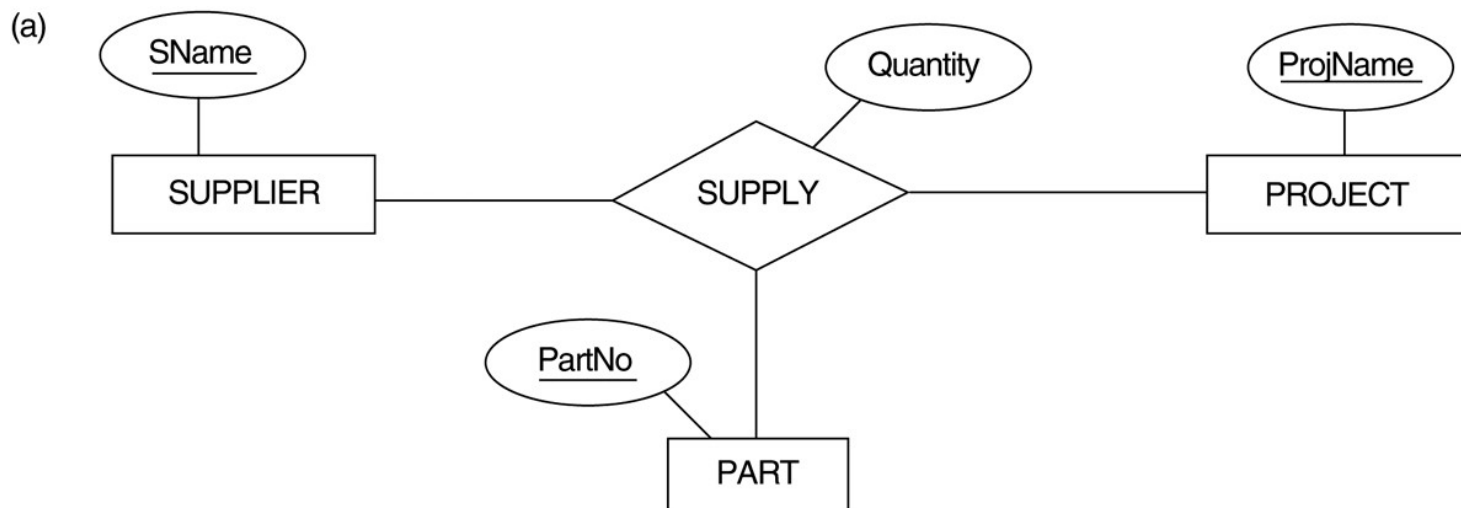
Mapeamento de relações N-árias

- ❑ Para cada relação n-ária R , onde $n > 2$, será criada uma nova relação S que representa R
- ❑ Inclua como chave estrangeira em S as chaves primárias das relações que representam as entidades participantes
- ❑ Inclua atributos simples do relacionamento como atributos de S

Projeto Lógico

Mapeamento de relações N-árias

Exemplo



SUPPLIER

<u>SNAME</u>	...
--------------	-----

PROJECT

<u>PROJNAME</u>	...
-----------------	-----

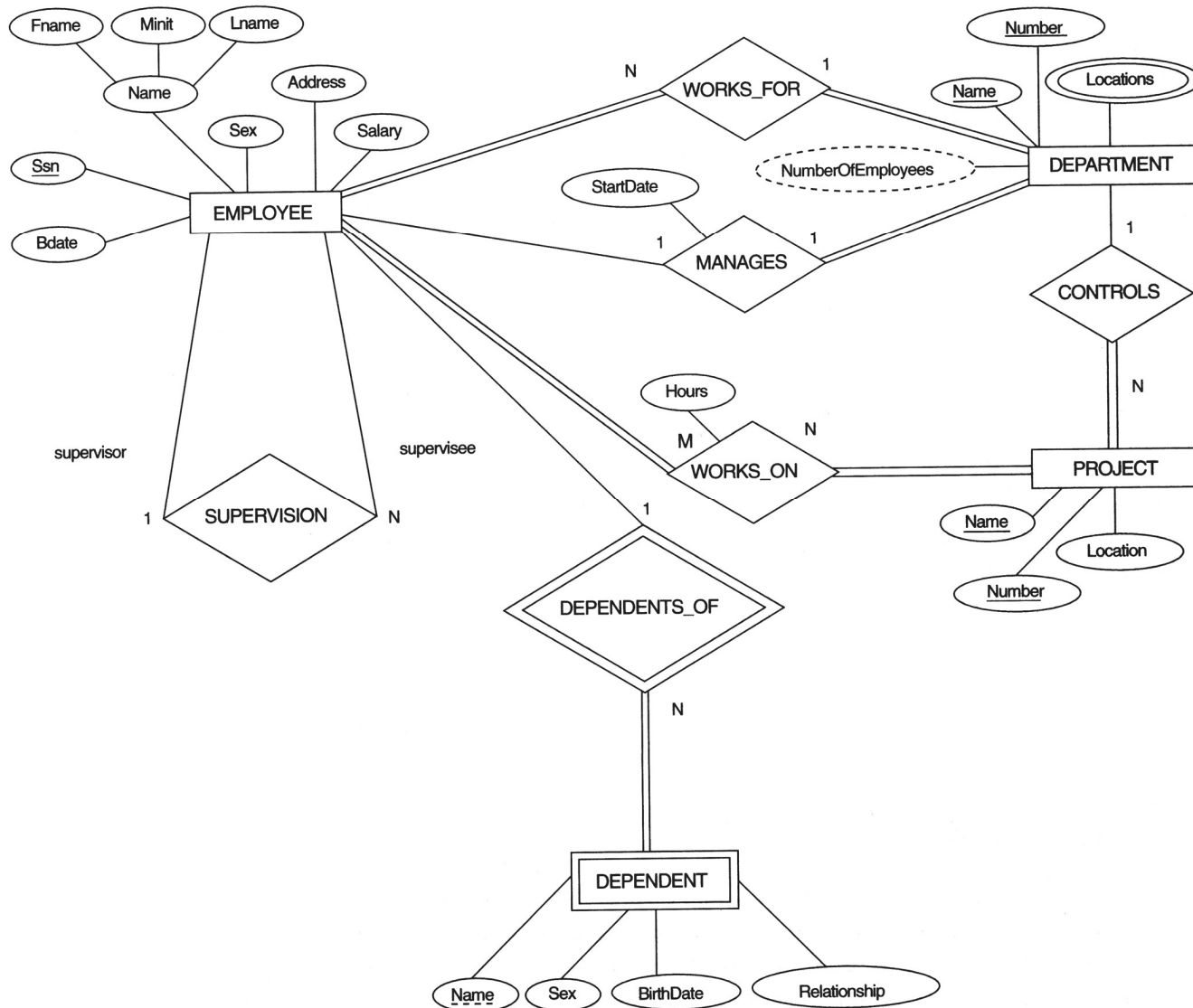
PART

<u>PARTNO</u>	...
---------------	-----

SUPPLY

<u>SNAME</u>	PROJNAME	<u>PARTNO</u>	QUANTITY
--------------	----------	---------------	----------

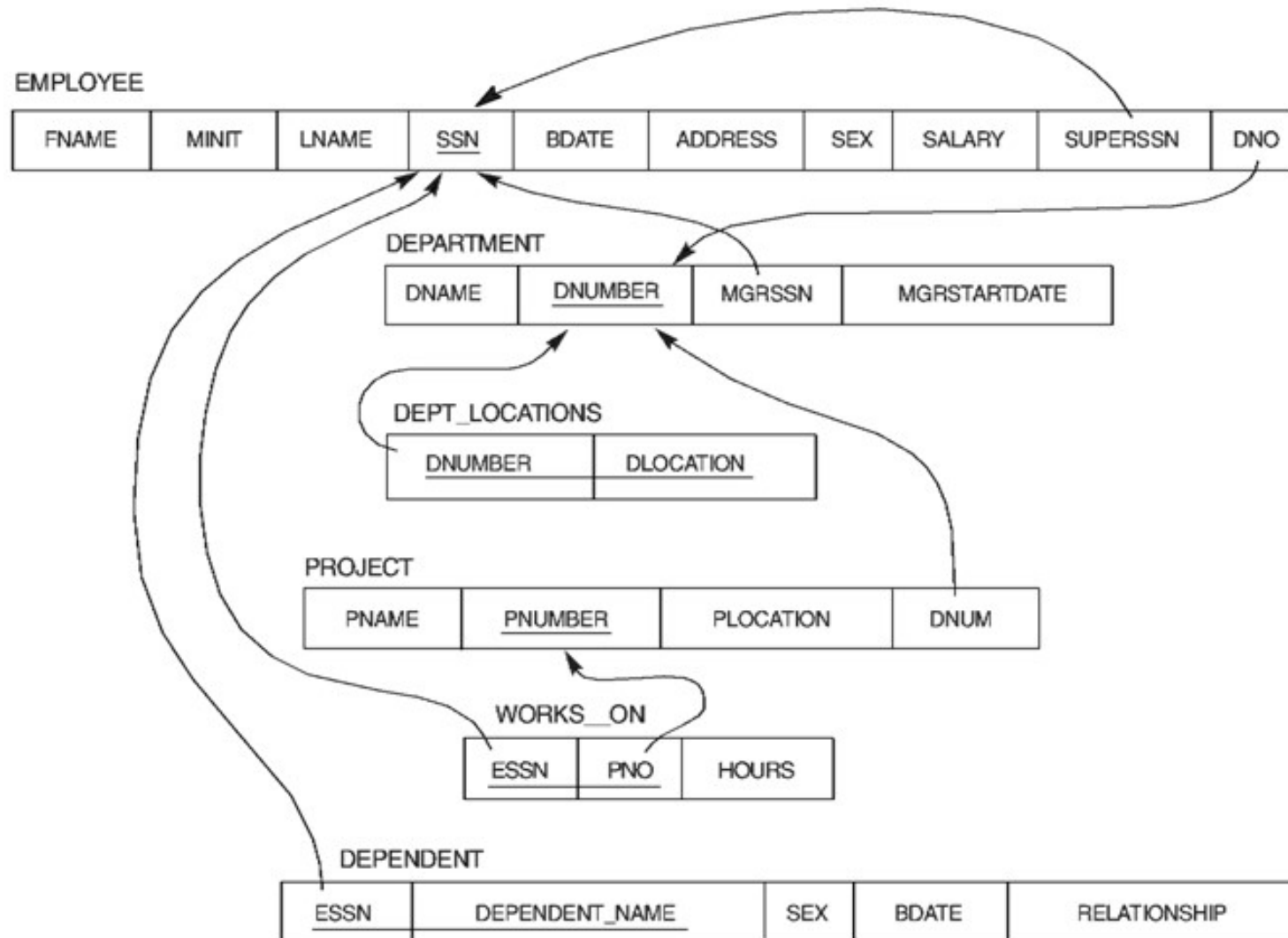
Projeto Lógico ER Conceitual



Projeto Lógico

Tabelas

Relações e Atributos



Projeto Lógico

Mapeamento EER (Herança Simples)

- Mapeamento de Especialização e Generalização (Herança Simples)
 - Sendo C uma superclasse definida como:
 - $C = \{K_1, A_1, \dots, A_n\}$, sendo K sua chave primária
 - Que possui m sub-classes (S_1, S_2, \dots, S_m)
 - Conversão para o modelo relacional pode ser feita utilizando uma das seguintes opções:
 - A - Múltiplas relações (incluindo super classes e subclasses)
 - B- Múltiplas relações (incluindo somente sub-classes)
 - C - Relação Única com um atributo “tipo”
 - D - Relação Única com múltiplos atributos “tipo”

Projeto Lógico

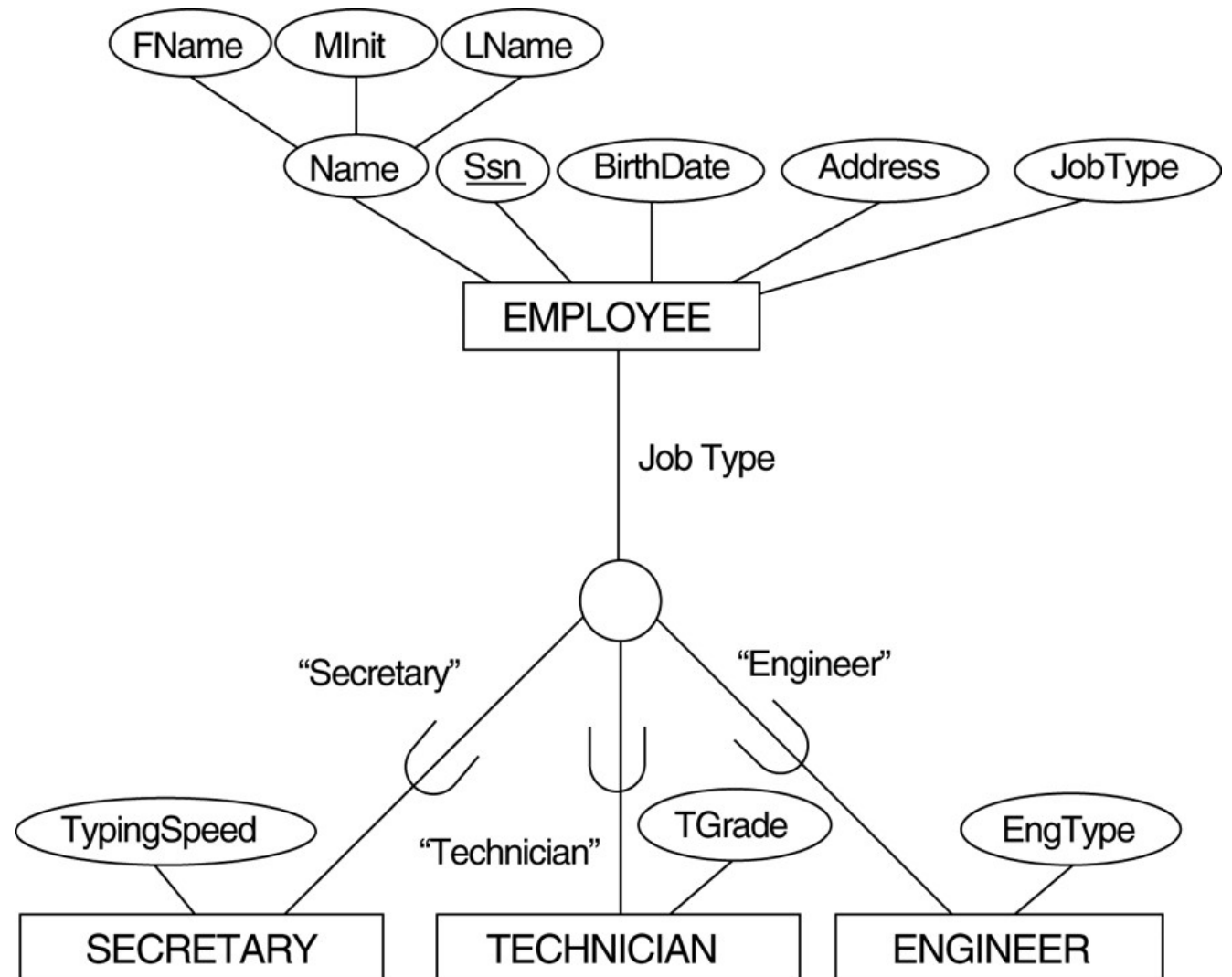
Mapeamento EER (Herança Simples) - A

- Múltiplas relações (incluindo super classes e subclasses)
 - A partir de $C=\{k, a_1, a_2, \dots, a_n\}$ e $\{S_1, S_2, \dots, S_m\}$
 - São produzidas múltiplas relações L (super classe) e L_i (subclasses)
 - $L=\{k, a_1, a_2, \dots, a_n\}$ e
 - $L_i=\{k\} \cup \{\text{atributos específicos de } S_i\} \mid 1 \leq i \leq m$
 - $PK(L_i) = k$
 - Esta opção é válida para qualquer tipo de especialização e generalização (total ou parcial, disjunta ou com sobreposição)

Projeto Lógico

Mapeamento EER (Herança Simples) - A

- ❑ Múltiplas relações (incluindo super classes e subclasses)
- ❑ Exemplo
 - Diagrama EER



Projeto Lógico

Mapeamento EER (Herança Simples) - A

- Múltiplas relações (incluindo super classes e subclasses)
- Exemplo
 - Projeto Lógico

(a) EMPLOYEE

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

SECRETARY

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

TECHNICIAN

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

ENGINEER

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

Projeto Lógico

Mapeamento EER (Herança Simples) - B

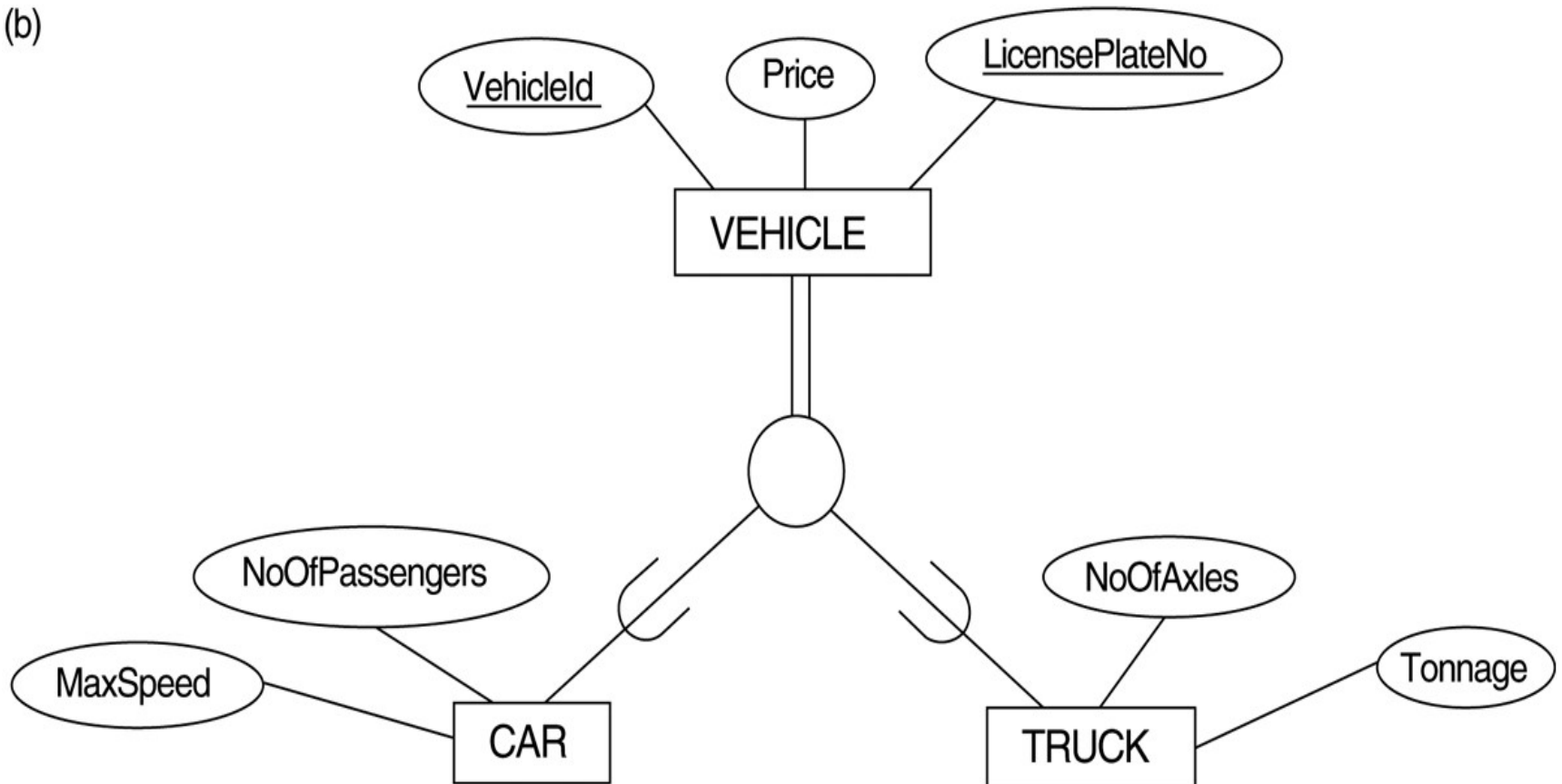
- Múltiplas relações (incluindo somente sub-classes)
 - A partir de $C = \{k, a_1, a_2, \dots, a_n\}$ e $\{S_1, S_2, \dots, S_m\}$
 - Para cada sub-classes são produzidas múltiplas relações (L_i)
 - $L_i = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos específicos de } S_i\} \mid 1 \leq i \leq m$
 - $PK(L_i) = k$
 - Esta opção é válida relações de especialização e generalização onde a participação é total

Projeto Lógico

Mapeamento EER (Herança Simples) - B

- ❑ Múltiplas relações (incluindo somente sub-classes)
- ❑ Exemplo
 - EER

(b)



Projeto Lógico

Mapeamento EER (Herança Simples) - B

- Múltiplas relações (incluindo somente sub-classes)
- Exemplo
 - Projeto Lógico

(b) CAR

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

TRUCK

<u>VehicleId</u>	LicensePlateNo	Price	NoOfAxles	
------------------	----------------	-------	-----------	--

Projeto Lógico

Mapeamento EER (Herança Simples) - C

- Relação Única com um atributo “tipo”
 - A partir de $C = \{k, a_1, a_2, \dots, a_n\}$ e $\{S_1, S_2, \dots, S_m\}$
 - Produzir uma única relação (L) que contenha
 - $L = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos específicos de } S_1\} \cup \dots \cup \{\text{atributos específicos de } S_m\} \cup \{t\}$
 - $PK(L) = k$
 - Atributo “t” é chamado tipo ou discriminador e indica em qual relação a tupla participa
 - Esta opção é válida relações de especialização e generalização onde existe a disjunção

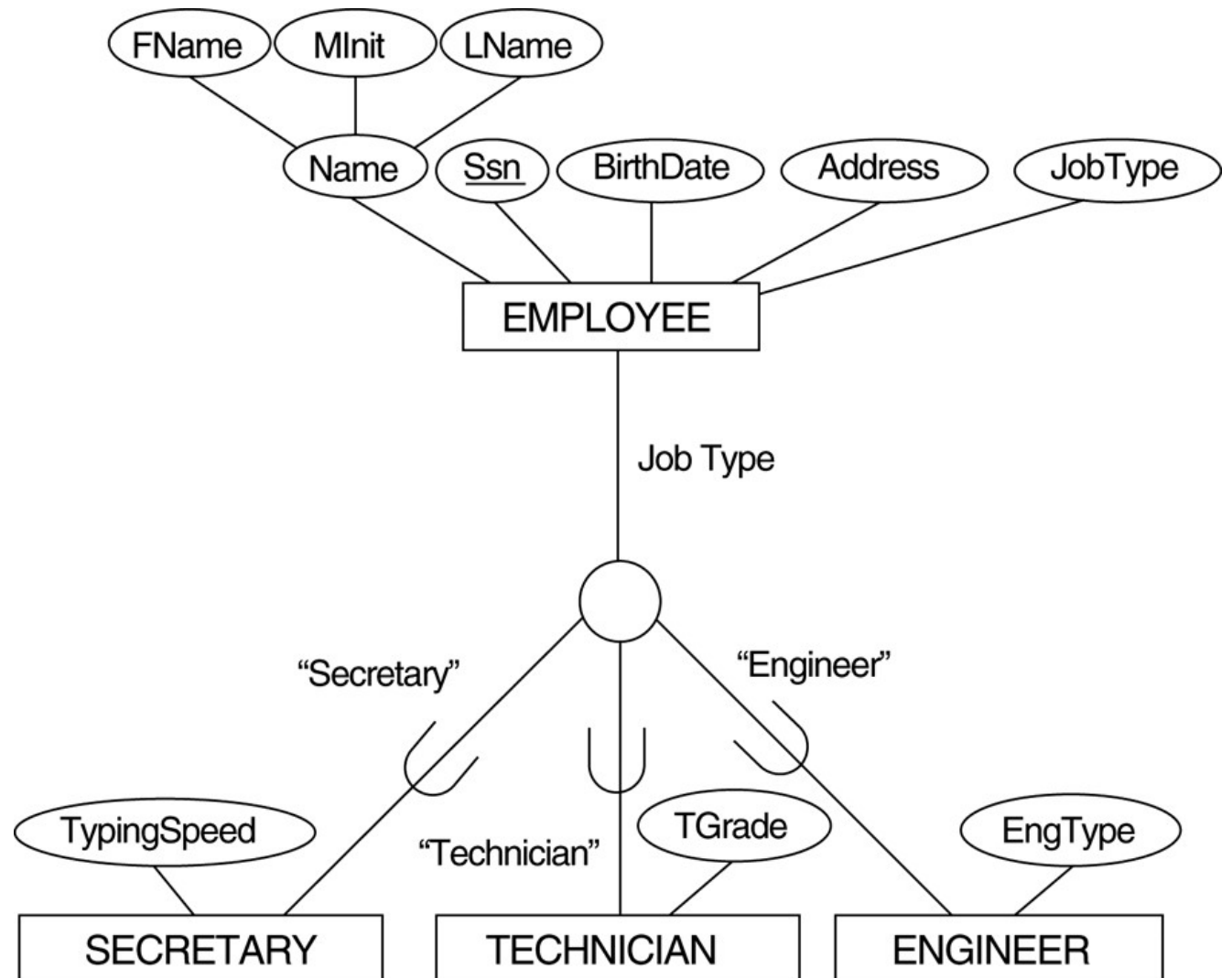
Projeto Lógico

Mapeamento EER (Herança Simples) - C

□ Relação Única com um atributo “tipo”

□ Exemplo

■ EER



Projeto Lógico

Mapeamento EER (Herança Simples) - C

- Relação Única com um atributo “tipo”
- Exemplo
 - Projeto Lógico

(c) EMPLOYEE

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

Projeto Lógico

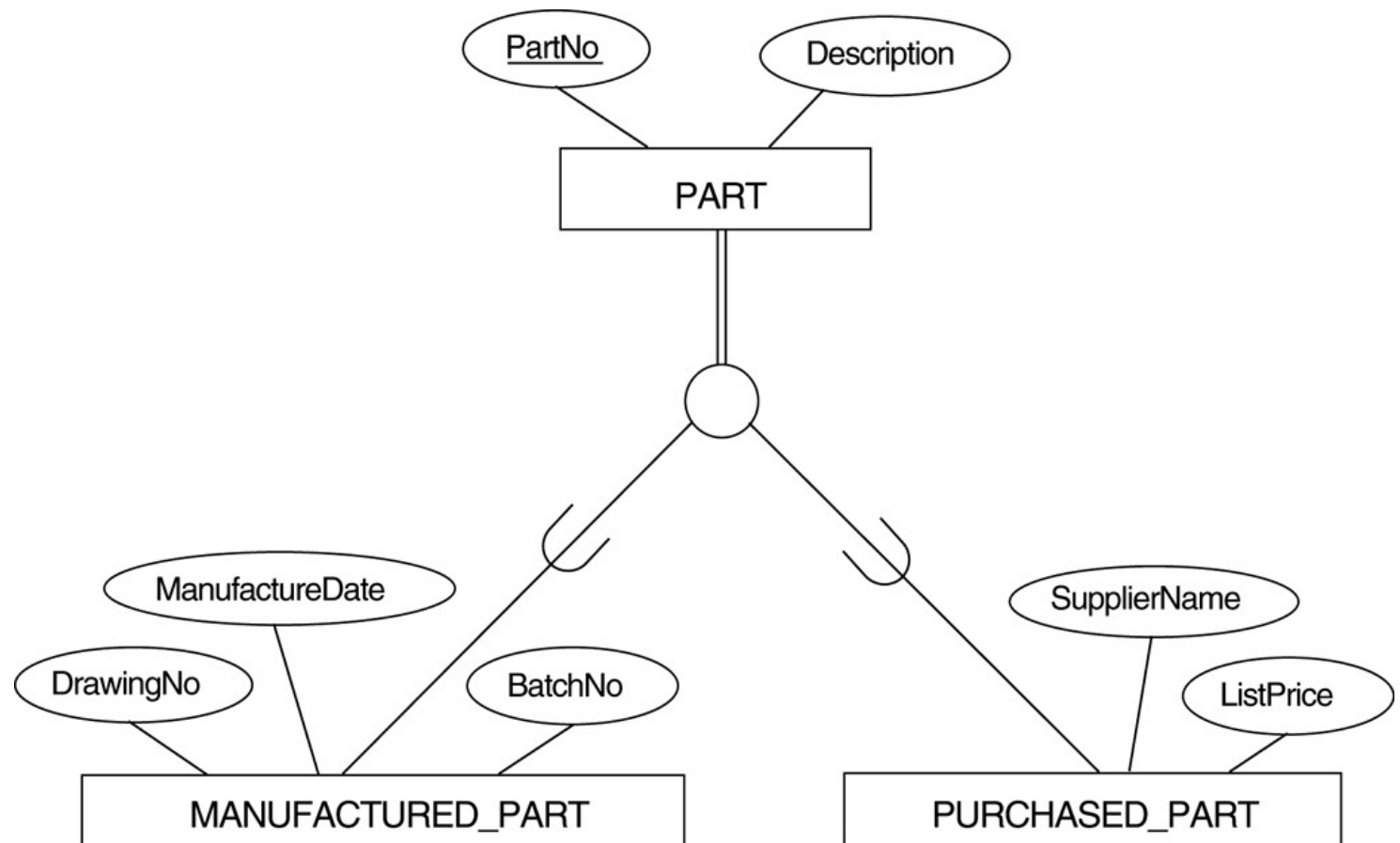
Mapeamento EER (Herança Simples) - C

- Relação Única com múltiplos atributo “tipo”
 - A partir de $C = \{k, a_1, a_2, \dots, a_n\}$ e $\{S_1, S_2, \dots, S_m\}$
 - Produzir uma única relação (L) que contenha
 - $L = \{k, a_1, a_2, \dots, a_n\} \cup \{\text{atributos específicos de } S_1\} \cup \dots \cup \{\text{atributos específicos de } S_m\} \cup \{t_1, t_2, \dots, t_m\}$
 - $PK(L) = k$
 - Atributos “ t_i ” são booleanos e chamados tipo ou discriminador, indicando em quais relações a tupla participa
 - Esta opção é válida relações de especialização e generalização onde existe a sobreposição

Projeto Lógico

Mapeamento EER (Herança Simples) - C

- Relação Única com múltiplos atributo “tipo”
- Exemplo
 - EER



Projeto Lógico

Mapeamento EER (Herança Simples) - C

- Relação Única com múltiplos atributo “tipo”
- Exemplo
 - Projeto Lógico

(d)

PART

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

Projeto Lógico

Mapeamento EER (Herança Múltipla)

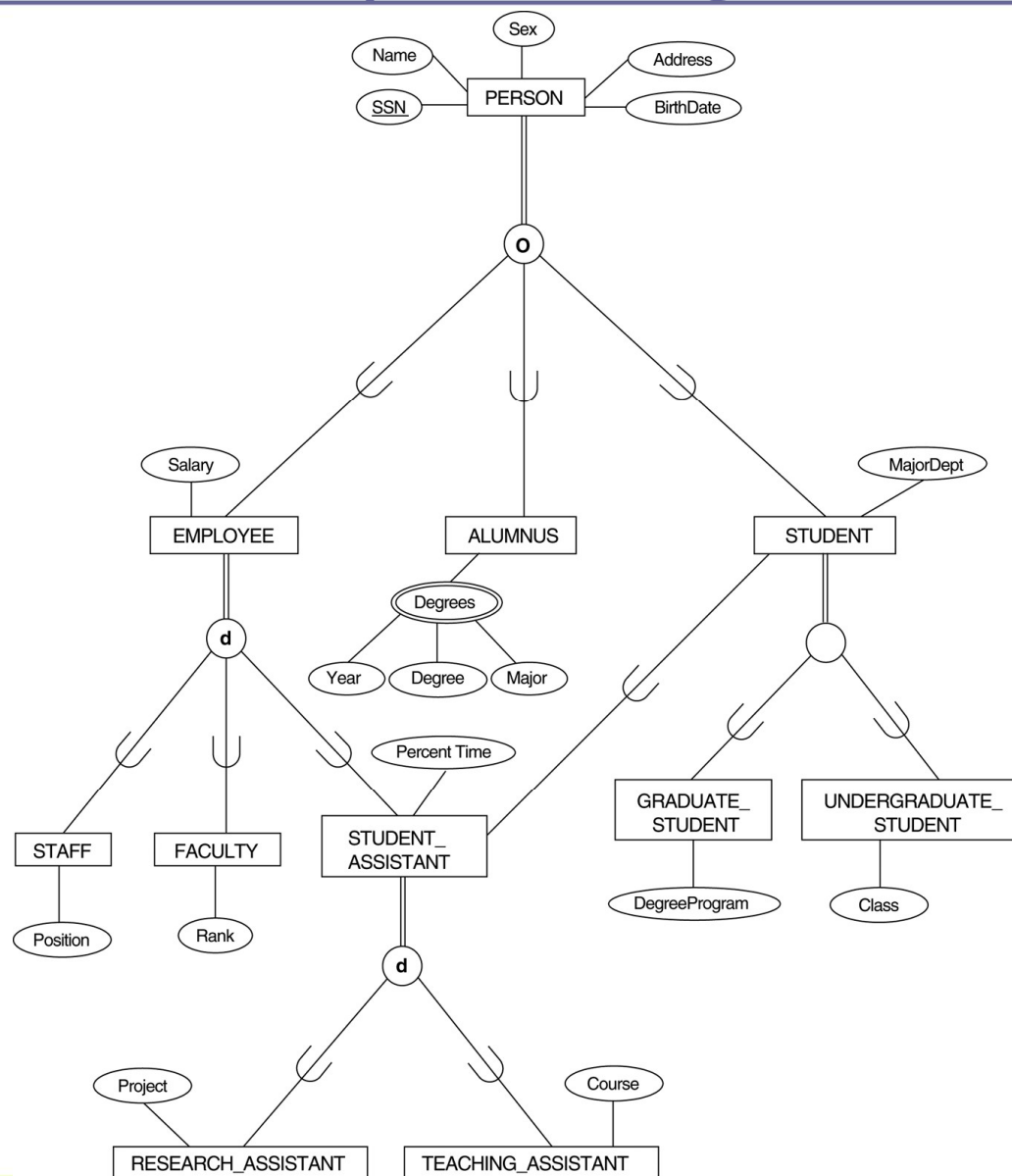
- ❑ Na herança múltipla uma mesma classe é subclasse de várias superclasses
- ❑ Neste caso todas as classes devem possuir o mesmo atributo chave
- ❑ Opções C ou D apresentadas anteriormente podem ser utilizadas no mapeamento do EER para o modelo relacional

Projeto Lógico

Mapeamento EER (Herança Múltipla)

Exemplo

EER



Projeto Lógico

Mapeamento EER (Herança Múltipla)

□ Exemplo

■ Projeto Lógico

PERSON

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

EMPLOYEE

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

ALUMNUS

<u>SSN</u>

ALUMNUS_DEGREES

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

STUDENT

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

Projeto Lógico

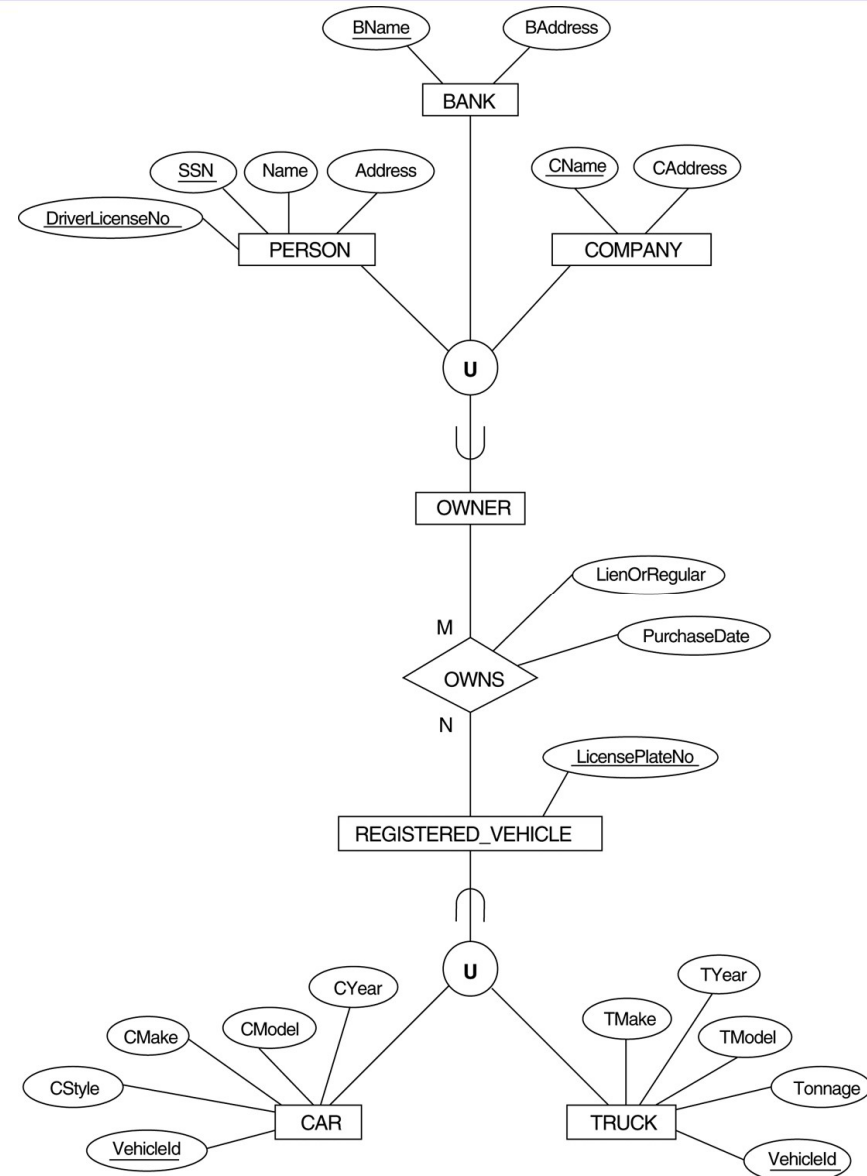
Mapeamento EER (Categorias)

- ❑ O mapeamento de categorias, onde suas chaves sejam diferentes, consiste na criação de um novo atributo chave
- ❑ Este atributo, chamado chave substituta (surrogate key) é utilizado na relação que representa a categoria
- ❑ Atributo é migrado para relações envolvidas na categoria

Projeto Lógico

Mapeamento EER (Categorias)

- Exemplo
 - EER



Projeto Lógico

Mapeamento EER (Categorias)

□ Exemplo

■ Projeto Lógico

- Foi criada uma relação chamada OWNER que corresponde à categoria
- Chave primária da relação “OwnerId”
- Chave é migrada para relações que participam da categoria

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	
------------------	--------	-------	--------	--

TRUCK

<u>VehicleId</u>	TMake	TModel	Tonnage	TYear
------------------	-------	--------	---------	-------

OWNS

<u>OwnerId</u>	<u>VehicleId</u>	PurchaseDate	LienOrRegular
----------------	------------------	--------------	---------------

Projeto Lógico

Mapeamento EER → Relacional

Exercício

