Lecture 5
ER & EER to Relational Mapping

Step 1: Mapping of Regular Entity Sets

- For each regular (strong) entity set E, create a relation R that includes all the simple attributes of E
  - Include only the simple component attributes of a composite attribute
  - 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 2: Mapping of Weak Entity Sets

- For each weak entity set W with owner entity set E, create a relation R and include all simple attributes (or simple components of composite attributes) of W as attributes of R
  - Include the primary key of the owner entity set as a foreign key of R
  - The primary key of R is the *combination of* the primary key of the owner entity set E and the partial key of the weak entity set W, if any

Step 3: Mapping of Binary 1:1 Relationship Sets

- For each binary 1:1 relationship set R, identify the relations S and T that correspond to the entity sets participating in R

1. **Foreign key approach**: Choose one of the relations S with total participation and include the primary key of T as a foreign key in S. Include all the simple attributes (or simple components of composite attributes) of R as attributes of S
2. **Merged relation approach**: When *both participations are total*, merge the two entity sets and the relationship set into a single relation
3. **Cross-reference or relationship relation approach**: 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 sets when none of the participation is total

Step 4: Mapping of Binary 1:N Relationship Sets

- For each binary 1:N relationship set R, identify the relation S that represent the participating entity set at the N-side of the relationship set
  - Include the primary key of the relation T that represents the other entity set participating in R as foreign key in S
  - Include any simple attributes (or simple components of composite attributes) of R as attributes of S

Step 5: Mapping of Binary M:N Relationship Sets

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

Step 6: Mapping of Multivalued attributes

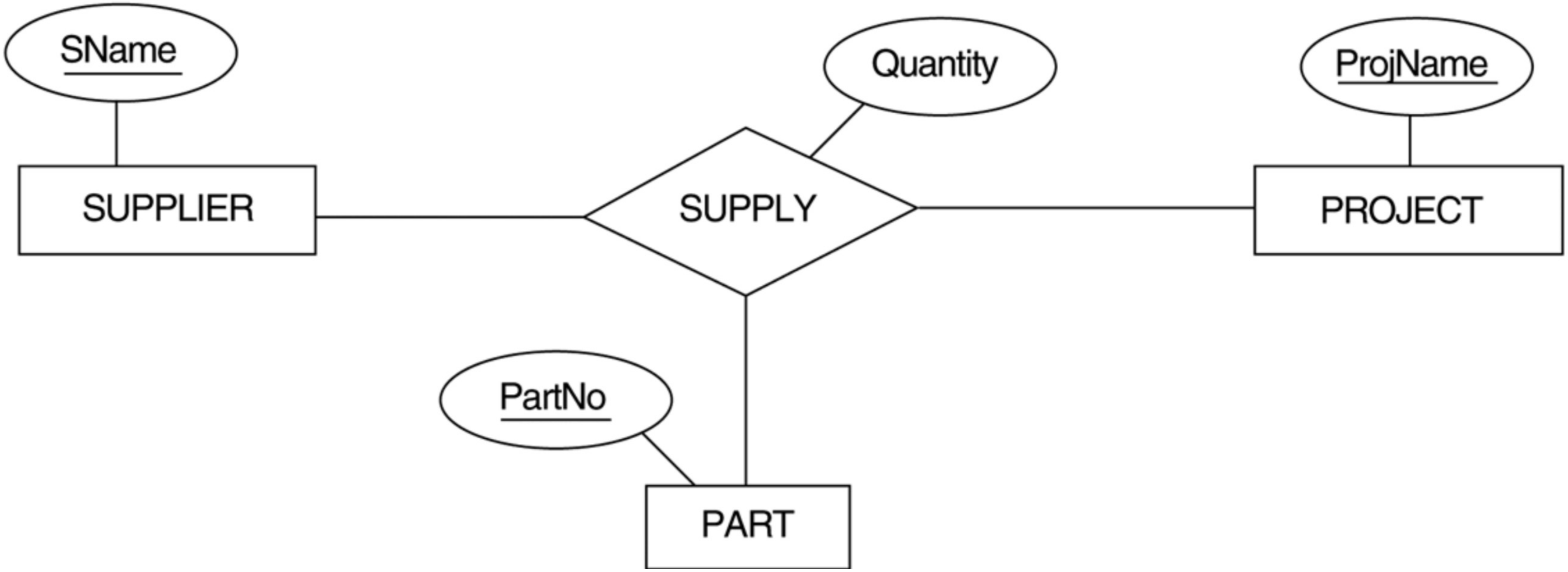
- For each multivalued attribute A, create a new relation R
  - Include an attribute corresponding to A, plus the primary key attribute K as a foreign key in R
  - The primary key of R is the combination of A and K. If the multivalued attribute is composite, include its simple components

Step 7: Mapping of N-ary Relationship Sets

- For each n-ary relationship set R (n>2), create a new relation S to represent R
  - Include the primary keys of the relations that represent the participating entity sets as foreign key attributes in S. Foreign keys form the primary key of S
  - Include any simple attributes (or simple components of composite attributes) of R as attributes of S

Example

Ternary relationship set: the SUPPLY relationship



Example (Cont.)

SUPPLIER

SNAME	. . .

PROJECT

PROJNAME	. . .

PART

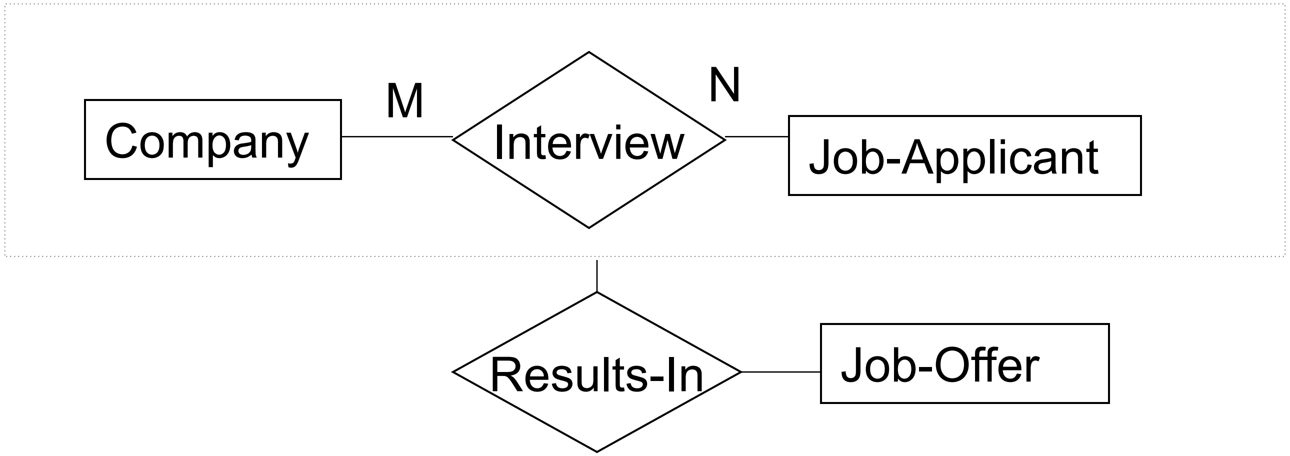
PARTNO	. . .

SUPPLY

SNAME	PROJNAME	PARTNO	QUANTITY

Mapping Aggregation

- Relationship will be mapped to a table; extended with primary keys of participating entity sets, plus its own attributes, if any
- Example:



Company

M

Interview

N

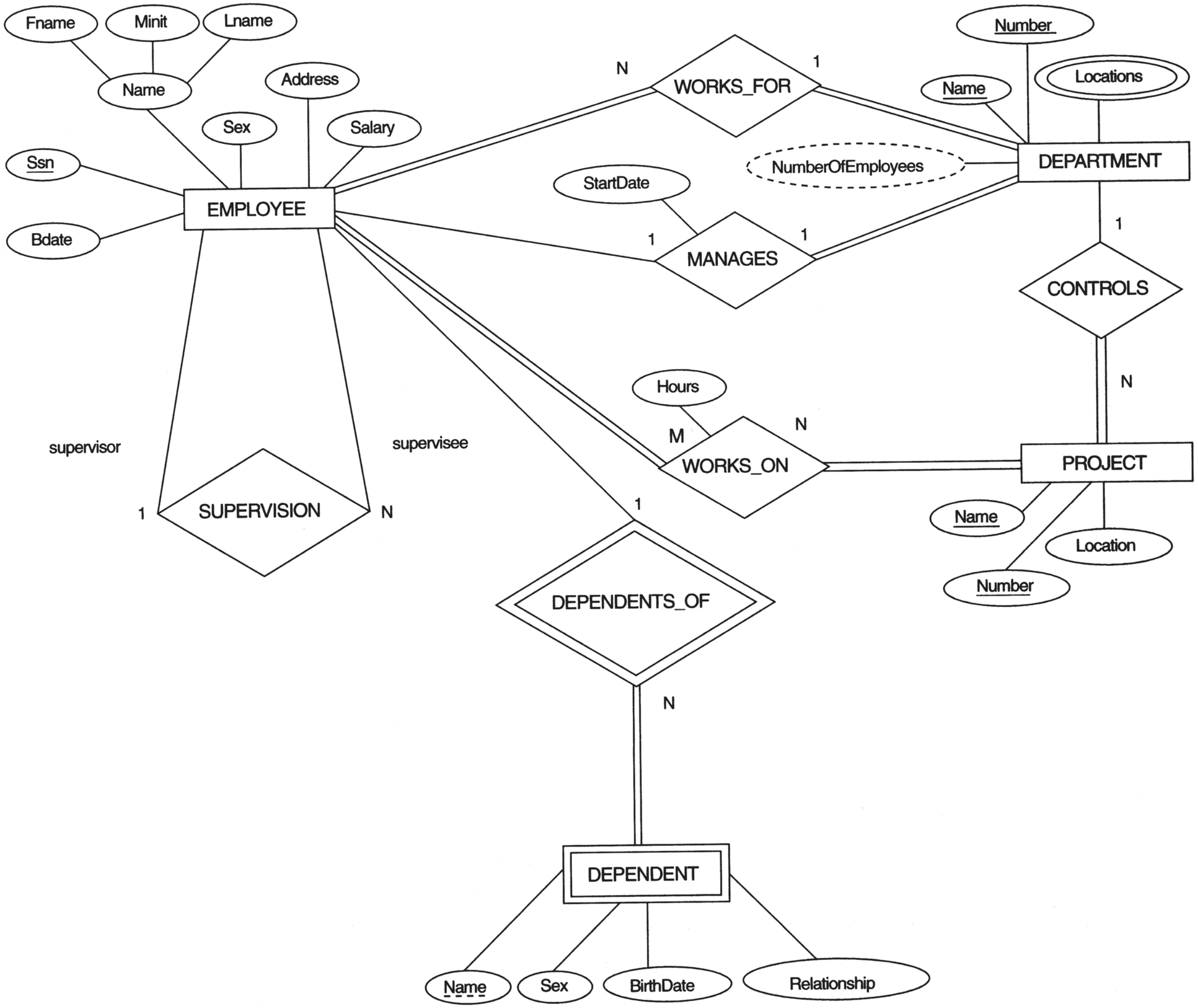
Job-Applicant

Results-In

Job-Offer

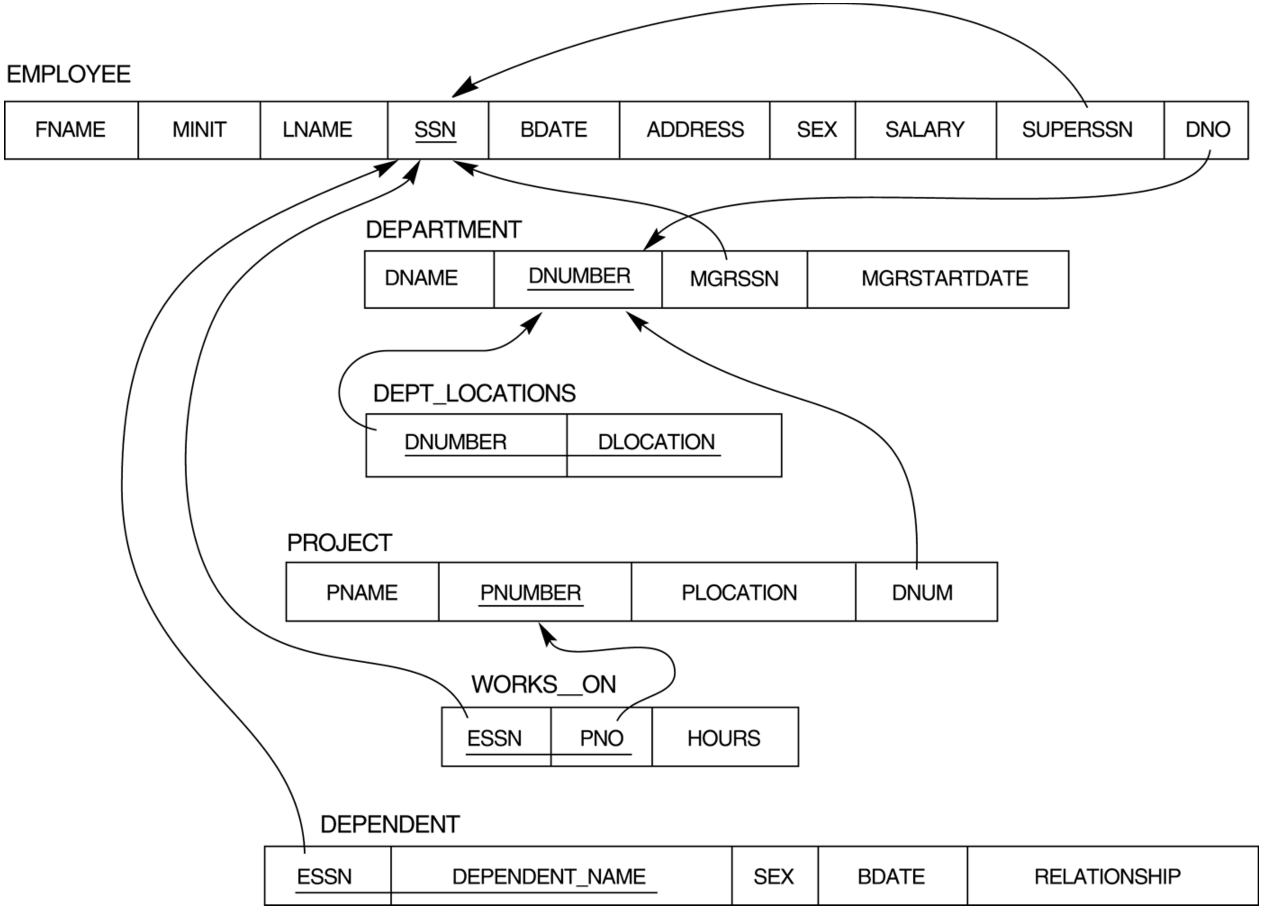
RESULTS_IN	P.K(Job_Offer)	P.K(Company)	P.K(Job-Applicant)

The ER Diagram for the COMPANY Database



Mapping Result

Mapping the COMPANY ER diagram into a relational schema



EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

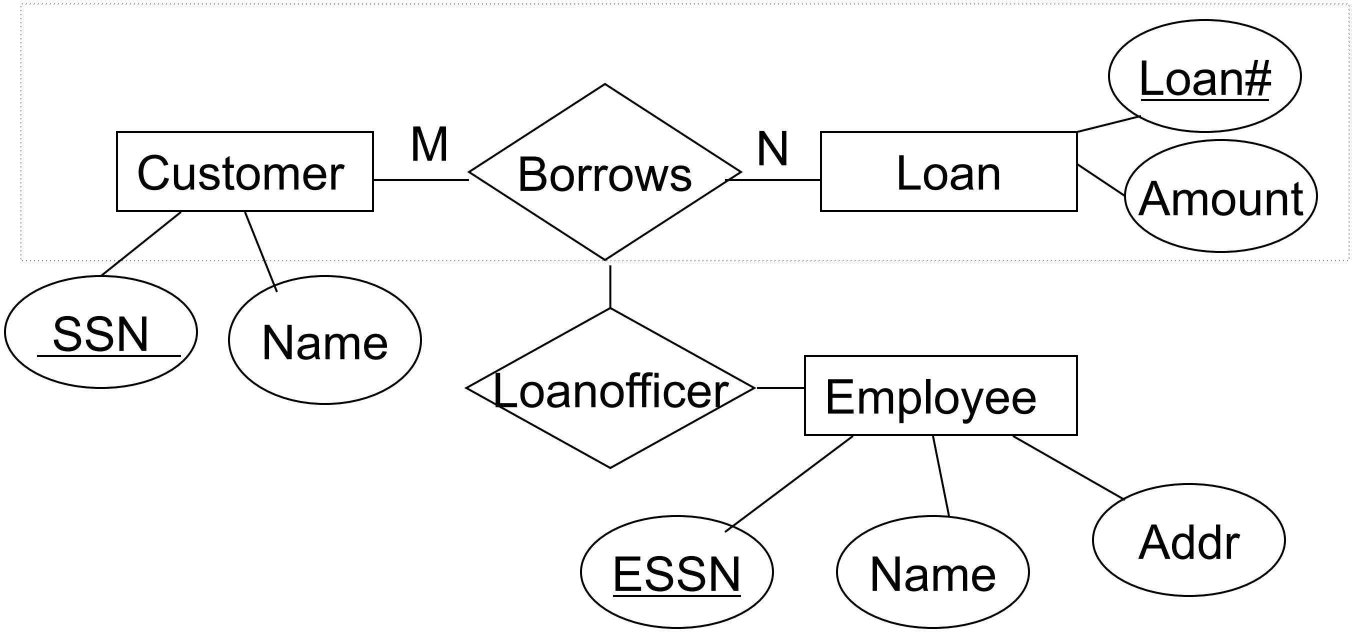
WORKS__ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

ER to Relational Exercise

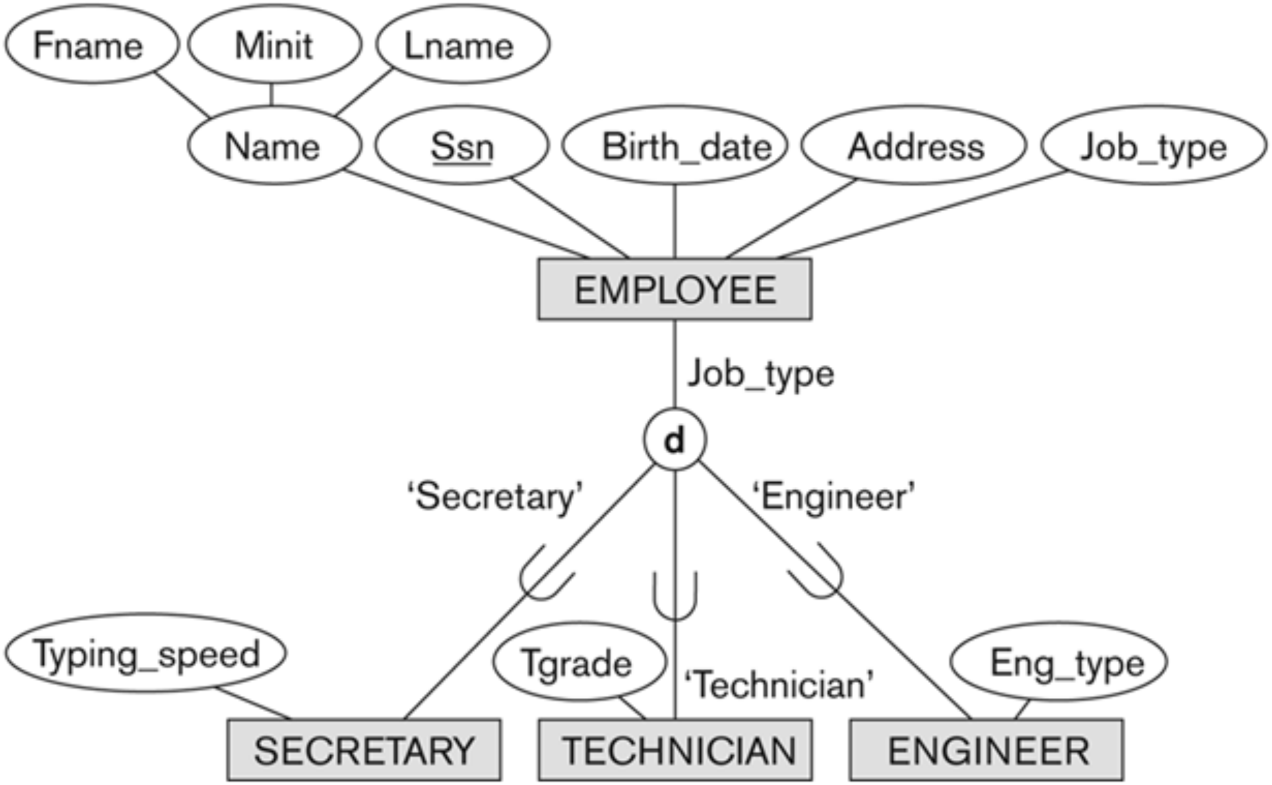


Mapping of Specialization or Generalization

- Convert each specialization with m subclasses $\{S_1, S_2,\ldots, S_m\}$ and generalized superclass C, where the attributes of C are $\{k, a_1,\ldots, a_n\}$ and k is the primary key, into relational schemas using one of the four following options:
  - **Option 1: Multiple relations - Superclass and subclasses**
    - Create a relation L for C with attributes Attrs(L) = $\{k, a_1,\ldots, a_n\}$, PK(L) = k
    - Create a relation L_i for each subclass S_i ($1 \leq i \leq m$) with the attributes Attrs(L_i) = {k} U {attributes of S_i}, PK(L_i)=k
    - This option works **for any specialization** (total or partial, disjoint or over-lapping)

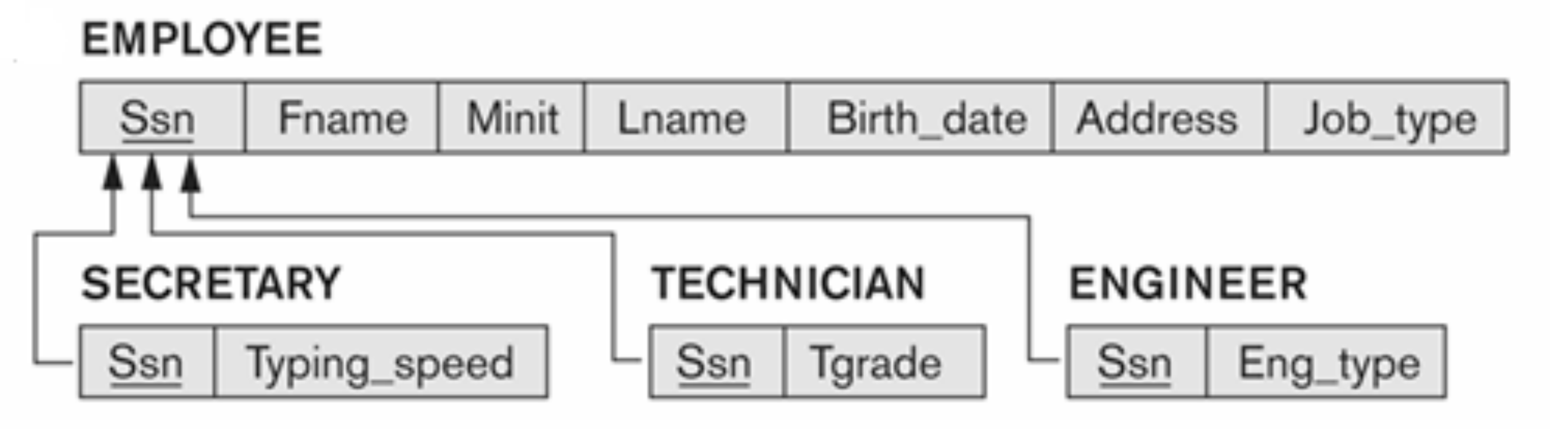
Example

EER diagram for an attribute-defined specialization on Job_type



Example (Cont.)

Mapping the EER diagram using option 1



EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type

SECRETARY

Ssn	Typing_speed

TECHNICIAN

Ssn	Tgrade

ENGINEER

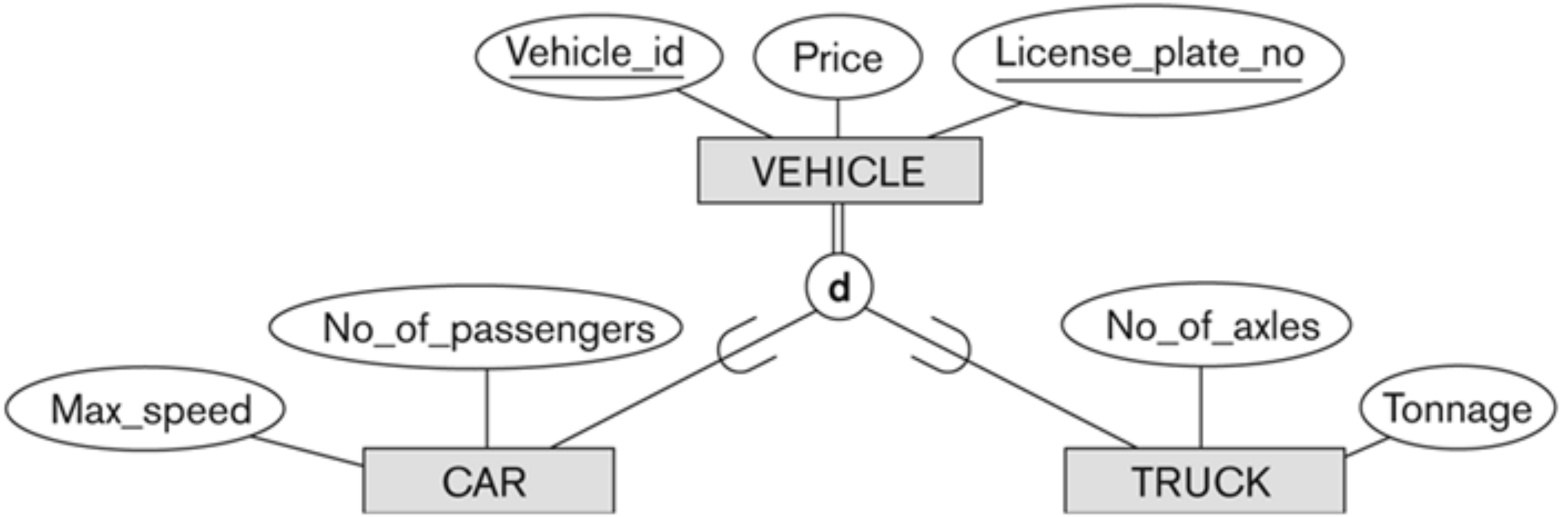
Ssn	Eng_type

Mapping of Specialization or Generalization (Cont.)

- **Option 2: Multiple relations - Subclass relations only**
  - Create a relation L_i for each subclass S_i ($1 \leq i \leq m$) with the attributes $Attrs(L_i)$ = {attributes of S_i} U {k, $a_1, \ldots, a_n$}, $PK(L_i) = k$
  - 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)

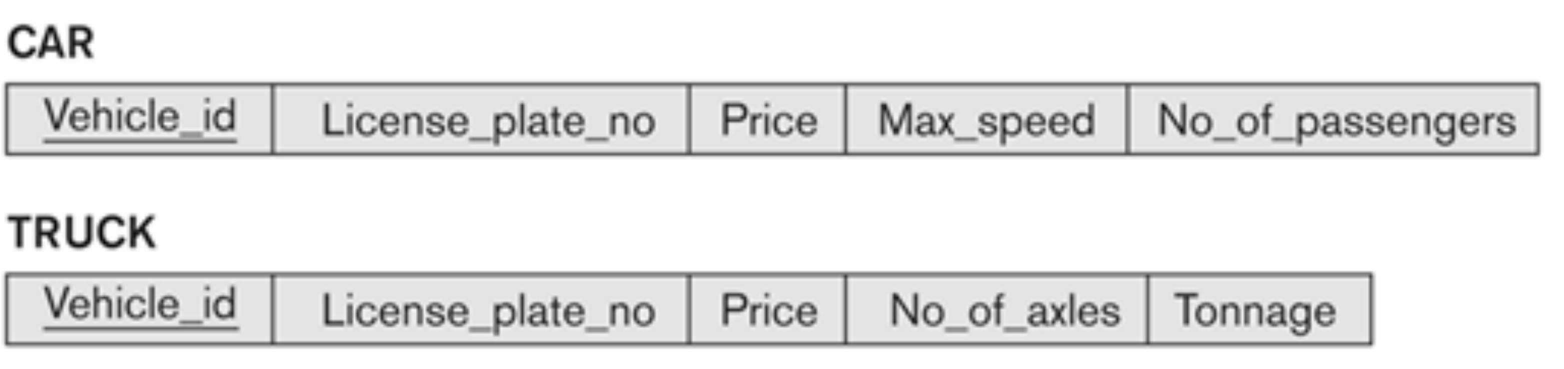
Example

Generalizing CAR and TRUCK into the superclass VEHICLE



Example (Cont.)

Mapping the EER diagram using option 2



CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage

If we use option 2 for overlapping, what would it result in?

- NULL values
- Redundancy
- Loss of information

Total Results: 0

Poll locked. Responses not accepted.

If we use option 2 for overlapping, what would it result in?

NULL values

Redundancy

Loss of information

Powered by Poll Everywhere

🔒 **Poll locked.** Responses not accepted.

If we use option 2 for overlapping, what would it result in?

NULL values

Redundancy

Loss of information

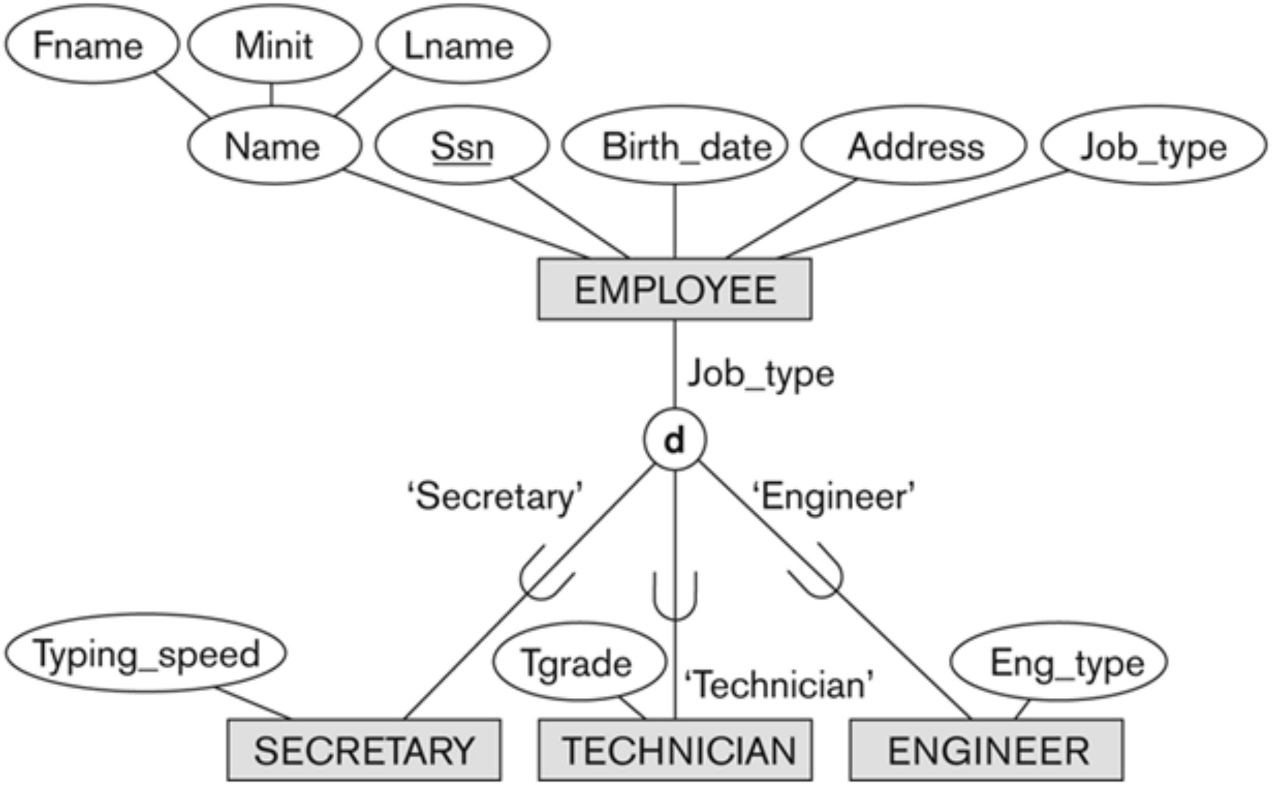
Powered by Poll Everywhere

Mapping of Specialization or Generalization (Cont.)

- **Option 3: Single relation with one type attribute**
  - Create a single relation L with attributes Attrs(L) = {k, a_1,…, a_n} U {attributes of S_1} U…U {attributes of S_m} U {t}, PK(L) = k
  - The attribute t is called a **type** attribute that indicates the subclass to which each tuple belongs, if any
  - This option works only for a specialization whose subclasses are **disjoint**
  - This option has the potential for generating many **null** values if many specific attributes exist in the subclasses

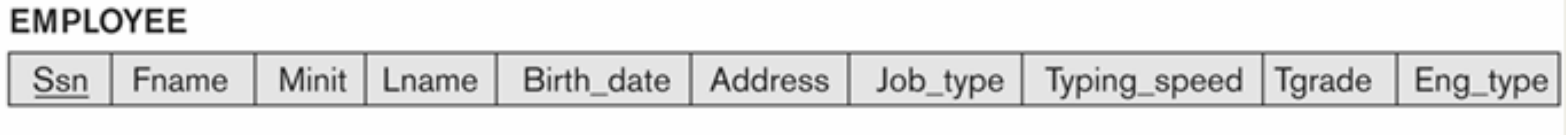
Example

EER diagram for an attribute-defined specialization on Job_type



Example (Cont.)

Mapping the EER diagram using option 3



EMPLOYEE

Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type

When poll is active, respond at **pollev.com/pdeng**

Would option 3 work for partial?

Yes

No

Total Results: 0

Powered by Poll Everywhere

Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

🔒 **Poll locked.** Responses not accepted.

Would option 3 work for partial?

Poll locked. Responses not accepted.

Would option 3 work for partial?

Powered by Poll Everywhere

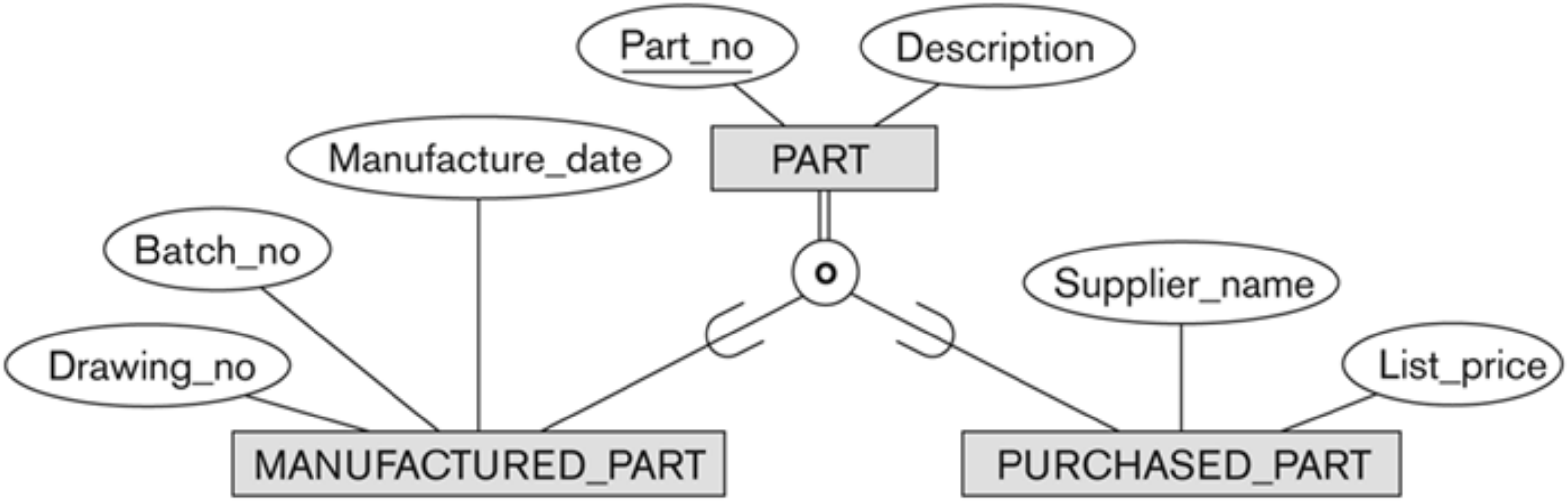
Start the presentation to see live content. For screen share software, share the entire screen. Get help at **pollev.com/app**

Mapping of Specialization or Generalization (Cont.)

- **Option 4: Single relation with multiple type attributes**
  - Create a single relation L with attributes Attrs(L) = {k, a_1,…, a_n} U {attributes of S_1} U…U {attributes of S_m} U {t_1, t_2,…, t_m}, PK(L) = k
  - Each t_i ($1\leq i \leq m$) is a Boolean type attribute indicating whether a tuple belongs to the subclass S_i
  - This option works for a specialization whose subclasses are overlapping (but also works for a disjoint specialization)

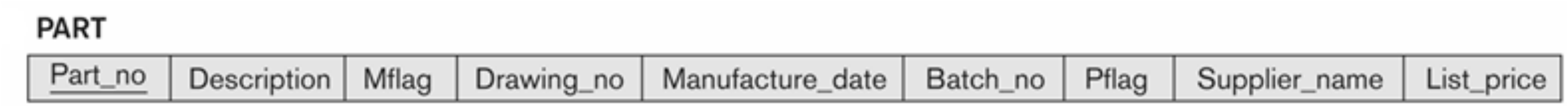
Example

EER diagram for an overlapping specialization



Example (Cont.)

Mapping the EER diagram using option 4 with Boolean type fields Mflag and Pflag



PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price

EER to Relational Summary

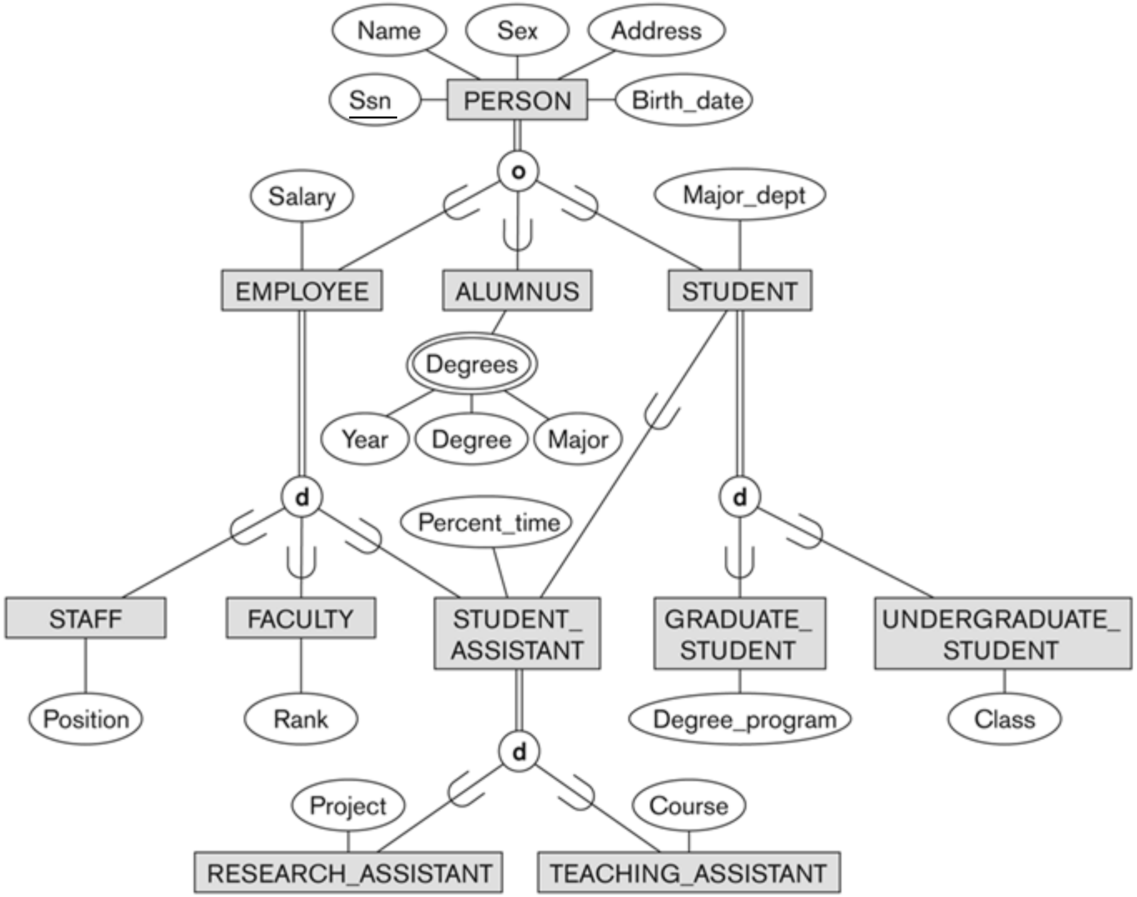
- Subclasses are overlapping (will also work for a disjoint specialization), each t_i ($1 \leq i \leq m$) is a Boolean attribute indicating whether a tuple belongs to subclass, S_i
- Option 1 works on disjoint/overlapping and total/partial
- Option 2 works well when both the disjoint and total constraints hold

Mapping of Shared Subclasses

- A shared subclass indicates multiple inheritance
- Any of the options can be applied to a shared subclass, subject to restrictions

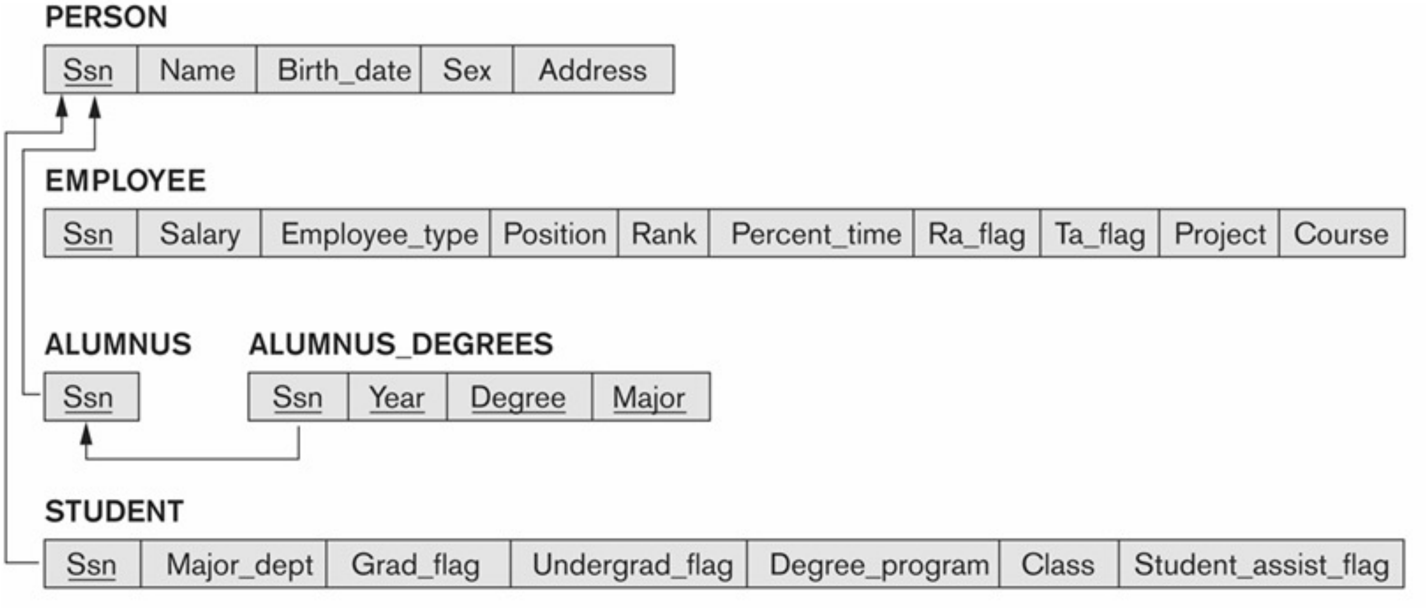
Example

- A specialization lattice for a UNIVERSITY database



Example (Cont.)

- Option 3 is used in the EMPLOYEE relation (Employee_type attribute)
- Option 4 is used in the STUDENT relation (Stud_assist_flag attribute)



PERSON

Ssn	Name	Birth_date	Sex	Address

EMPLOYEE

Ssn	Salary	Employee_type	Position	Rank	Percent_time	Ra_flag	Ta_flag	Project	Course

ALUMNUS

Ssn

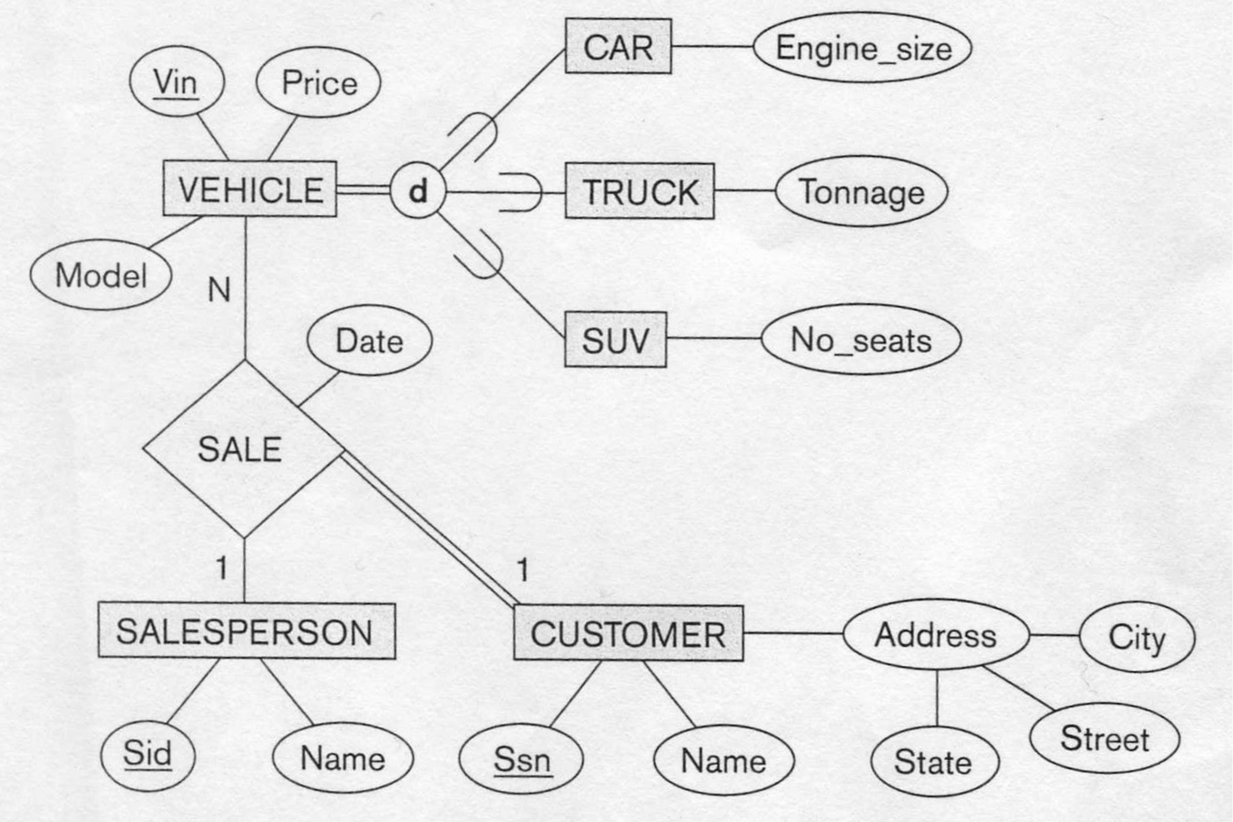
ALUMNUS_DEGREES

Ssn	Year	Degree	Major

STUDENT

Ssn	Major_dept	Grad_flag	Undergrad_flag	Degree_program	Class	Student_assist_flag

Exercise


CAR
Engine_size
Vin
Price
VEHICLE
d
TRUCK
Tonnage
Model
N
SUV
No_seats
Date
SALE
1
1
SALESPERSON
CUSTOMER
Address
City
Street
State
Sid
Name
Ssn
Name