CMSC 127

Relational Model and ER- and EER-to-Relational Mapping

Reginald Neil C. Recario
Institute of Computer Science
University of the Philippines Los Baños

Relational Model

- The Relational Model of Data is based on the concept of a ***Relation***.
- A Relation is a mathematical concept based on the idea of sets.
- The model was first proposed by Edgar Frank “Ted” Codd of IBM Research in 1970.

Informal Definitions

- Informally, a ***relation*** looks like a ***table*** of values.
- A relation typically contains a ***set of rows***.
- Each ***row*** corresponds to a real-world ***entity*** or ***relationship***
  - In the formal model, rows are called ***tuples***

Informal Definitions

- Each ***column*** corresponds to an attribute
- Each column has a column header that indicates the meaning of the data items in that column
  - In the formal model, the column header is called an ***attribute name*** (or just ***attribute***)

Example of a Relation

Relation Name

Attributes

STUDENT

Tuples

Name	No	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

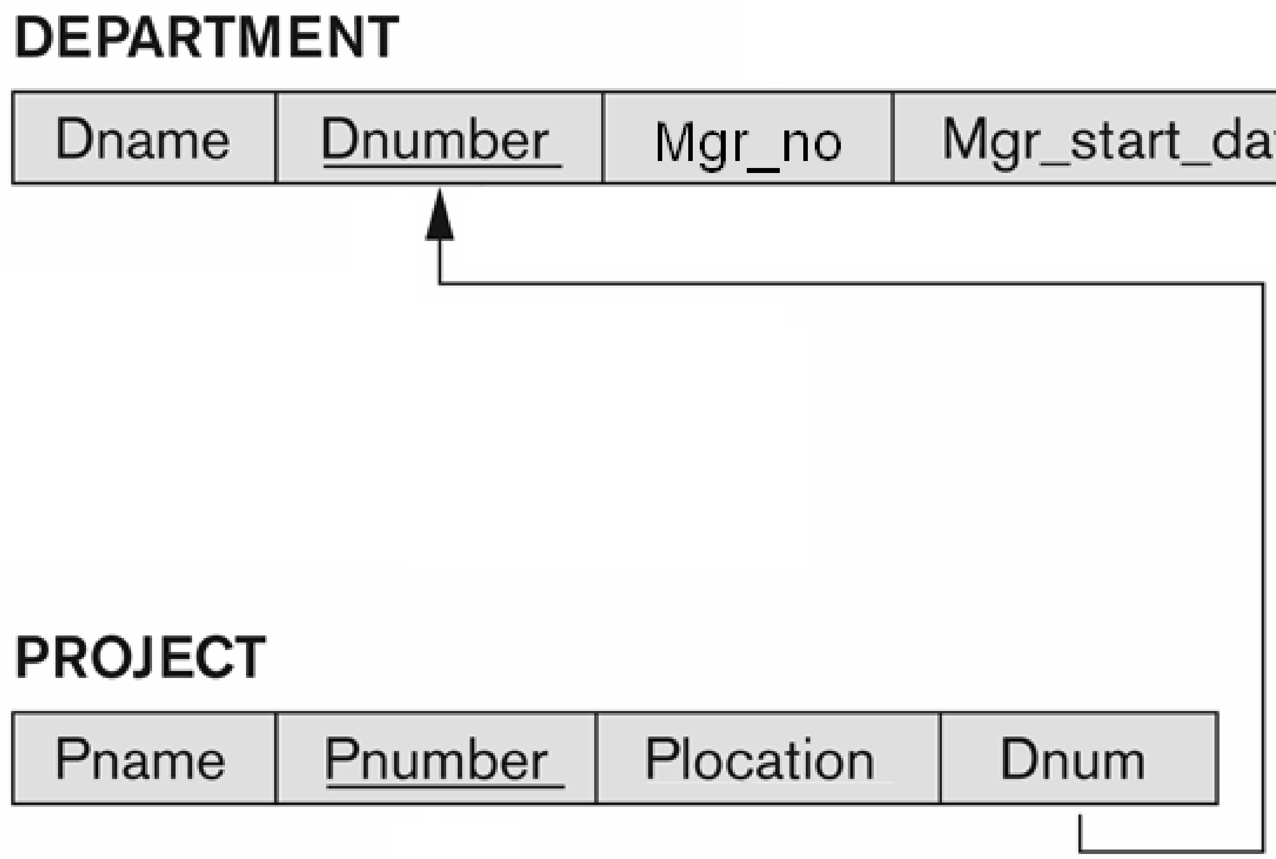
Informal Definitions

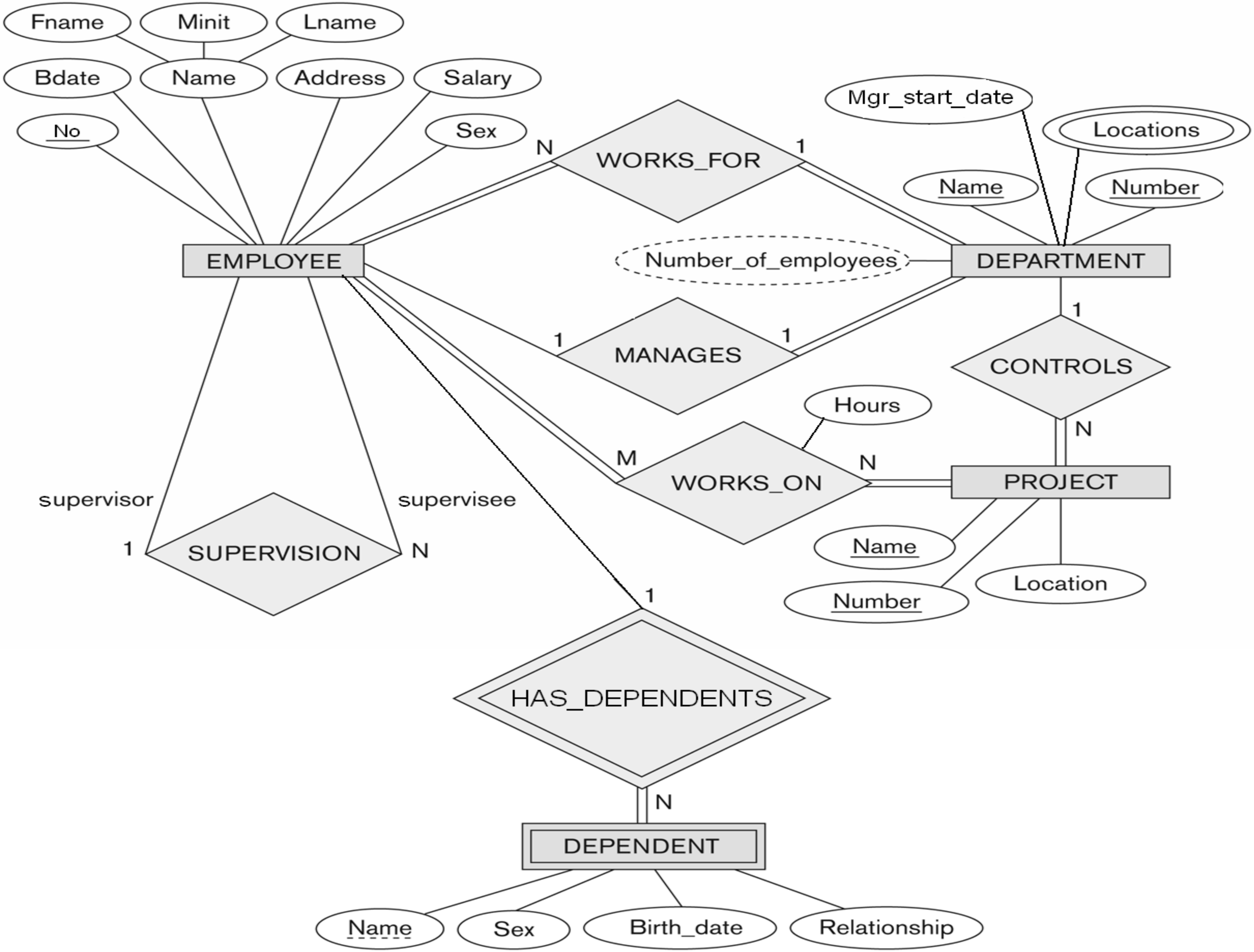
- Key of a Relation:
  - An attribute that uniquely identifies a row in the table
  - In the STUDENT table, No is the key
  - Sometimes row-ids or sequential numbers are assigned as keys
    - Called ***artificial key*** or ***surrogate key***

Informal Definitions

- ***Primary Key***
  - Chosen from the different keys of the relation
- ***Foreign Key***
  - An attribute that references the primary key of another relation

Example of a Foreign Key




Fname
Minit
Lname
Bdate
Name
Address
Salary
No
Sex
N
WORKS_FOR
1
Mgr_start_date
Locations
Name
Number
EMPLOYEE
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
supervisor
supervisee
1
SUPERVISION
N
Name
Location
Number
1
HAS_DEPENDENTS
N
DEPENDENT
Name
Sex
Birth_date
Relationship

ER-to-Relational Mapping

- ***Step 1: Mapping of Regular Entity Types***
  - 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

EMPLOYEE

Fname	Minit	Lname	No	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

ER-to-Relational Mapping

- ***Step 2: Mapping of Weak Entity Types***
  - Create a relation R for the weak entity type W and include all simple attributes of W
  - Include primary key of the owner entity types
  - Set 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

DEPENDENT

Eno	Dependent_name	Sex	Bdate	Relationship

ER-to-Relational Mapping

- ***Step3: Mapping of Binary 1:1 Relationship Types***
  - ***Method1: Foreign key approach***
    - Include primary key of relation T into relation S (as a foreign key)
    - It is better if S has a total participation in the given relationship type

Example

DEPARTMENT

Dname	Dnumber	Mgr_no	Mgr_start_date

ER-to-Relational Mapping

- ***Step3: Mapping of Binary 1:1 Relationship Types***
  - ***Method2: Merged relation approach***
    - If both total participation, merge two relations

ER-to-Relational Mapping

- ***Step3: Mapping of Binary 1:1 Relationship Types***
  - ***Method3: cross-reference or relationship relation approach***
    - Set up a third relation and add the primary keys of the two associated entity types

Example

MANAGES

Mgr_no	Dnum

ER-to-Relational Mapping

- ***Step4: Mapping of Binary 1:N Relationship Types***
  - Include primary key of “1” side into “N” side relation (as a foreign key)
  - Include all simple attributes

Example

EMPLOYEE

Fname	Minit	Lname	No	Bdate	Address	Sex	Salary	Super_no	Dno

PROJECT

Pname	Pnumber	Plocation	Dnum

ER-to-Relational Mapping

- ***Step5: Mapping of Binary M:N Relationship Types***
  - Create a new relation
  - Include primary keys of both side relations (as foreign keys)
  - The primary key of the new relation is the combination of the primary keys of two associated entity types
  - Include relationship attribute(s)

Example

WORKS_ON

Eno	Pno	Hours

ER-to-Relational Mapping

- ***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
  - The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Example

DEPT_LOCATIONS

Dnumber	Dlocation

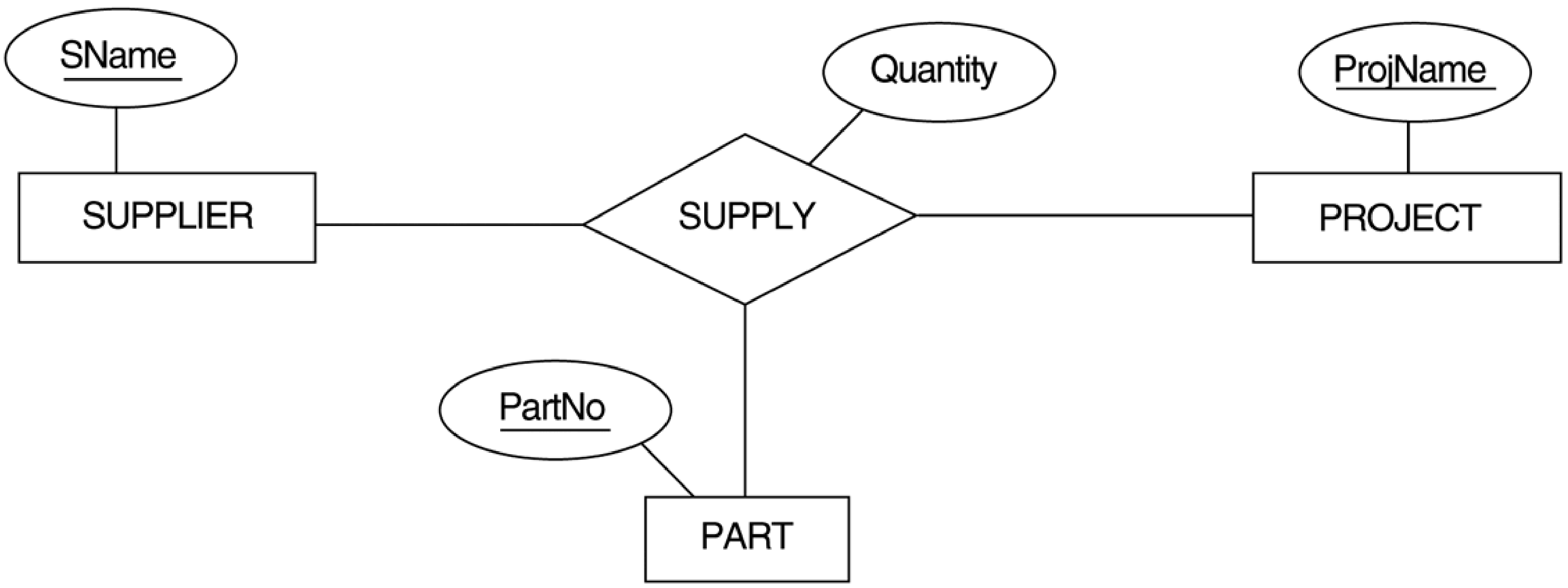
DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

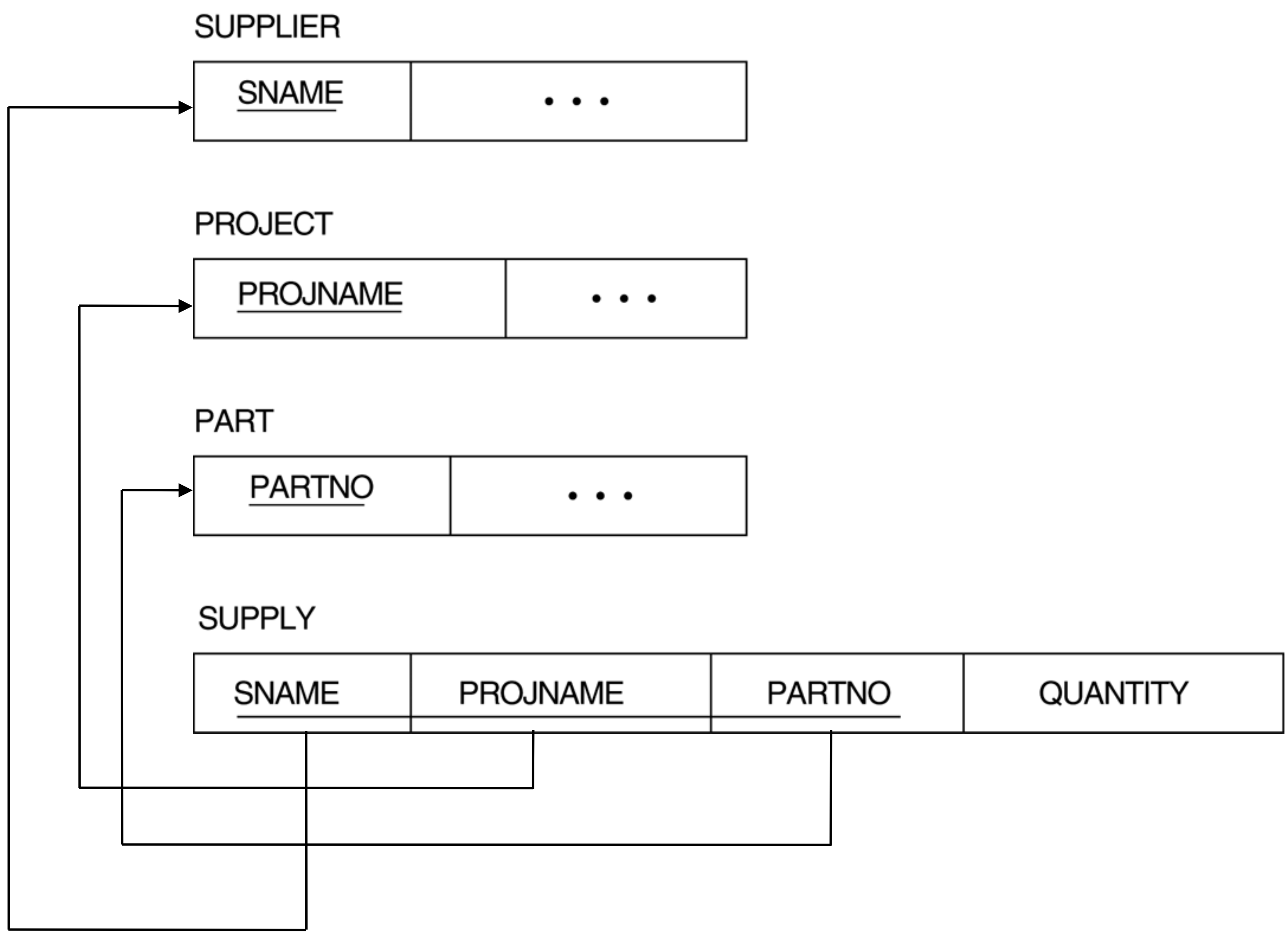
ER-to-Relational Mapping

- ***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.
  - Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.

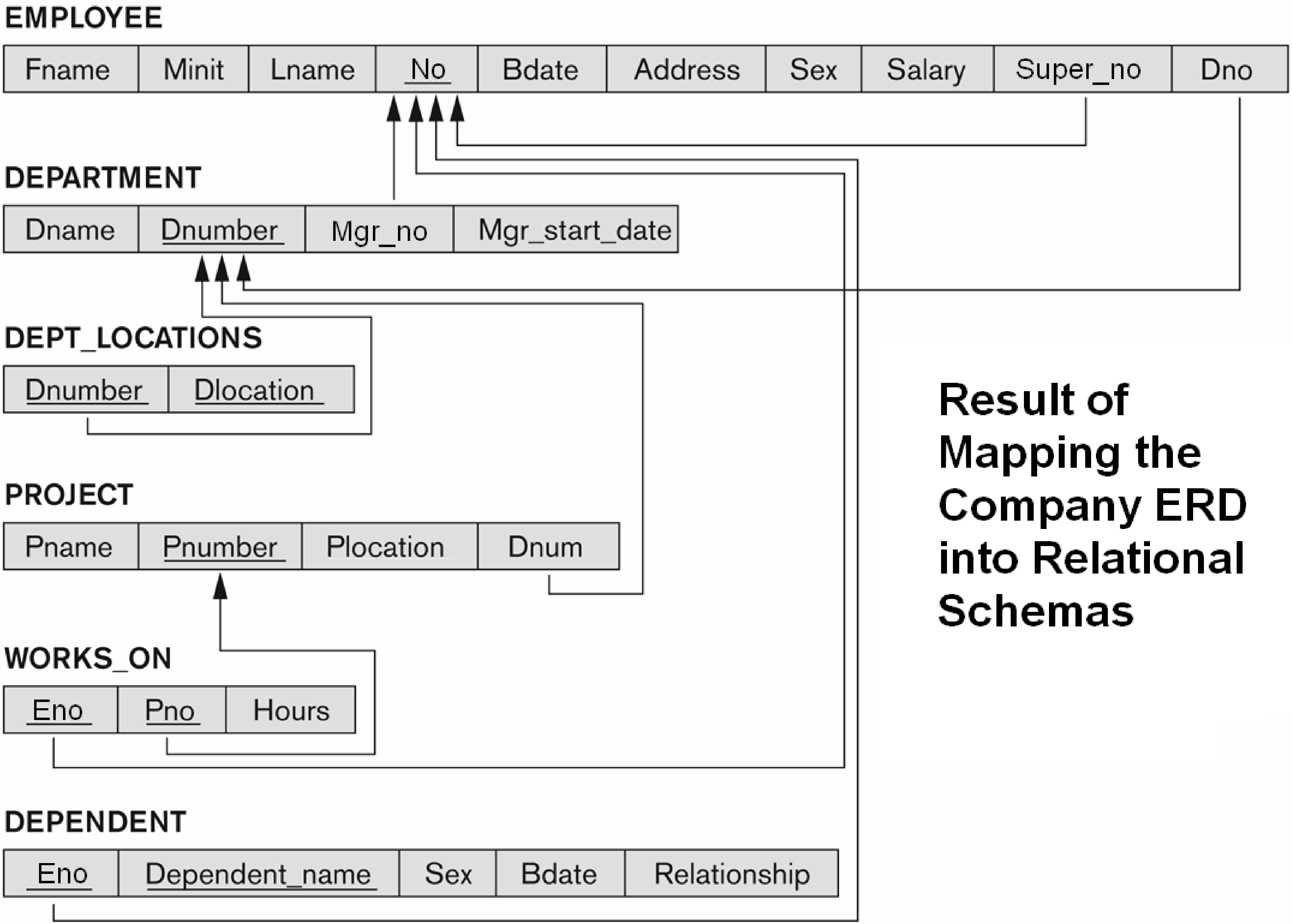
Example


SName
Quantity
ProjName
SUPPLIER
SUPPLY
PROJECT
PartNo
PART

Example



SUPPLIER
SNAME
. . .
PROJECT
PROJNAME
. . .
PART
PARTNO
. . .
SUPPLY
SNAME
PROJNAME
PARTNO
QUANTITY



Result of Mapping the Company ERD into Relational Schemas

Note:

- The identifying relationship type of a weak entity type has no corresponding table because all its attributes are already present in the relation that corresponds to the weak entity type.

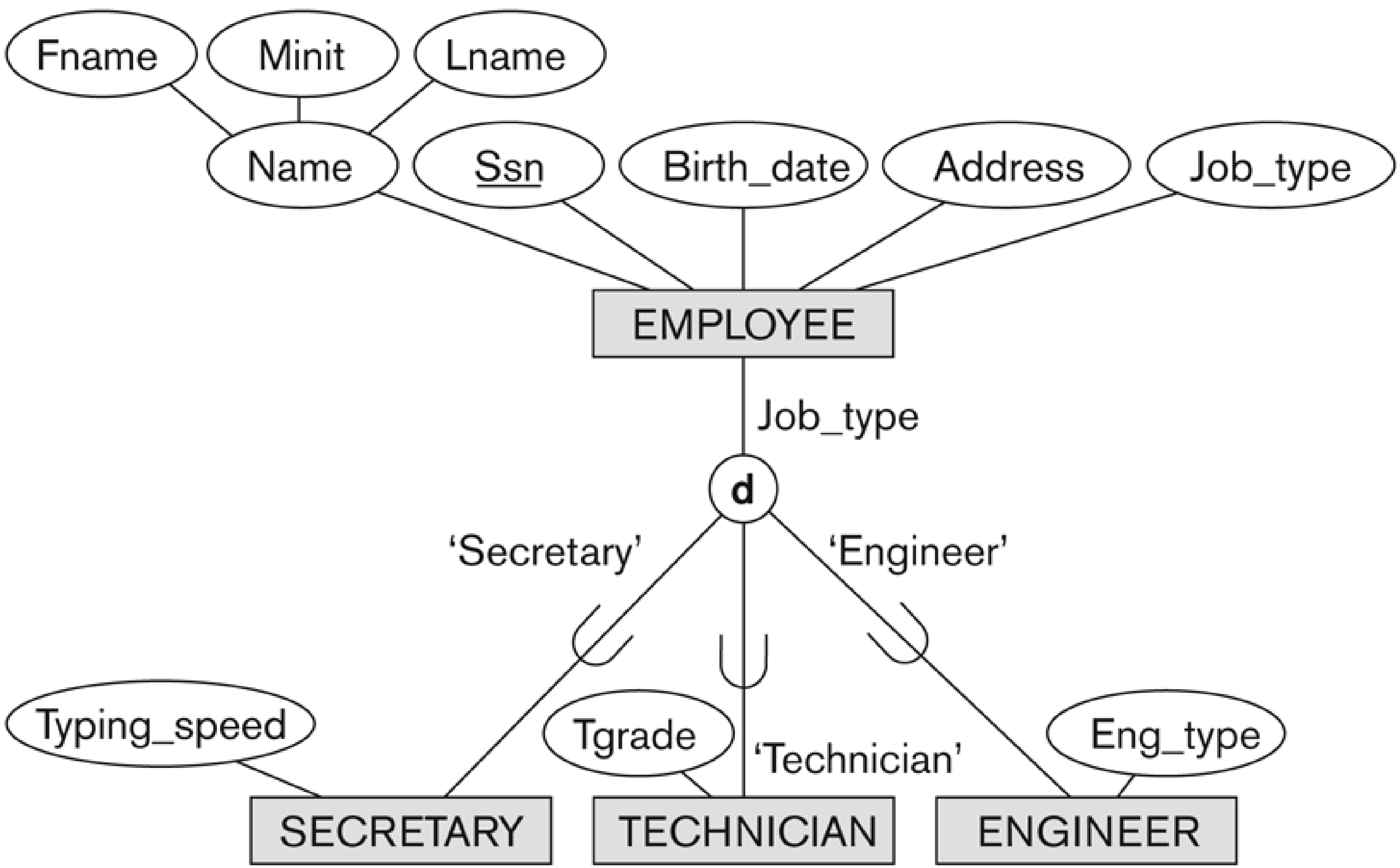
EER-to-Relational Mapping

- Step 8: Options for Mapping Specialization or Generalization
  - Four options:
    - Option 8A: Multiple relations-Superclass and subclasses
    - Option 8B: Multiple relations-Subclass relations only
    - Option 8C: Single relation with one type attribute
    - Option 8D: Single relation with multiple type attributes

EER-to-Relational Mapping

- ***Option 8A: Multiple relations-Superclass and subclasses***
  - Create a relation for the superclass.
  - Create a relation for each subclass, include primary key of superclass and local attributes.
  - This option works for any specialization (total or partial, disjoint or overlapping).

Example



Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
d
'Secretary'
'Engineer'
Typing_speed
Tgrade
'Technician'
Eng_type
SECRETARY
TECHNICIAN
ENGINEER

Using Option 8A

EMPLOYEE

SSN	FName	MInit	LName	BirthDate	Address	JobType

SECRETARY

SSN	TypingSpeed

TECHNICIAN

SSN	TGrade

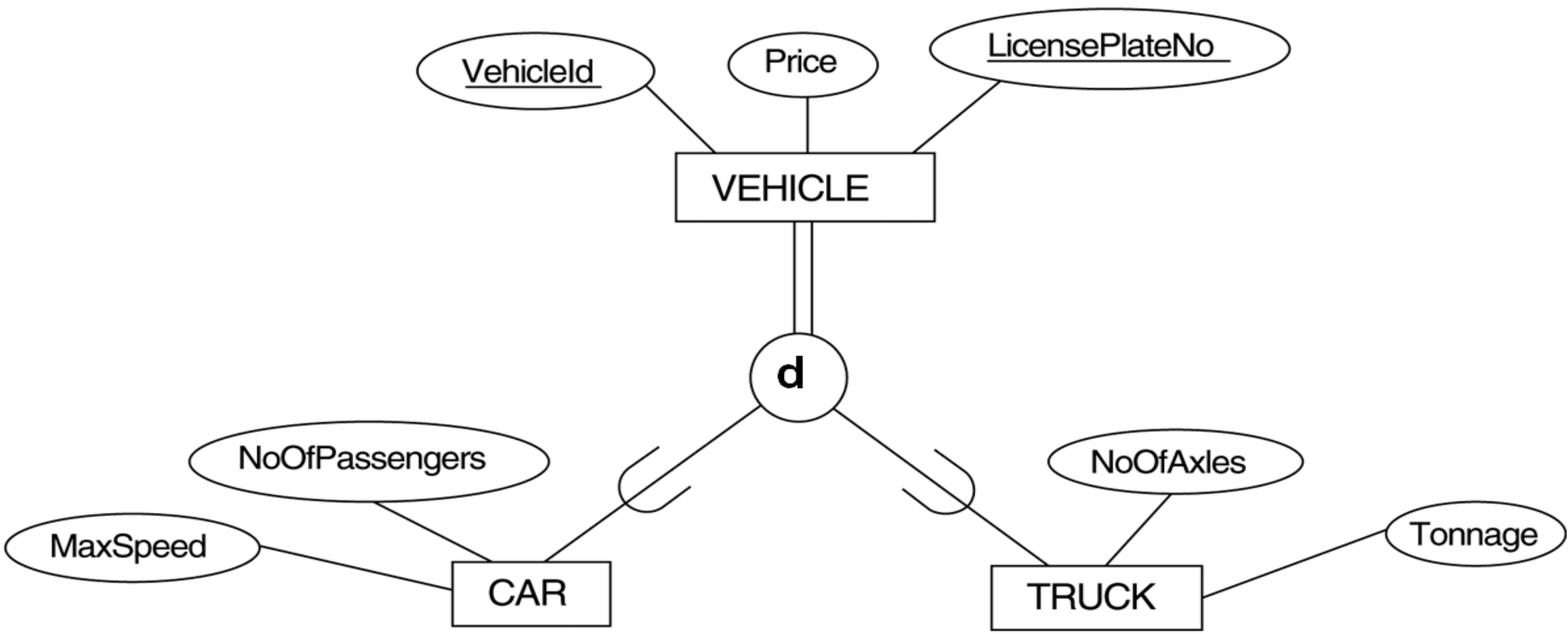
ENGINEER

SSN	EngType

EER-to-Relational Mapping

- ***Option 8B: Multiple relations-Subclass relations only***
  - Create a relation for each subclass with all local and inherited attributes
  - This option only works for a total specialization.
  - This is also more suitable for a disjoint specialization.

Example


VehicleId
Price
LicensePlateNo
VEHICLE
d
NoOfPassengers
MaxSpeed
CAR
NoOfAxles
Tonnage
TRUCK

Using Option 8B

CAR

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

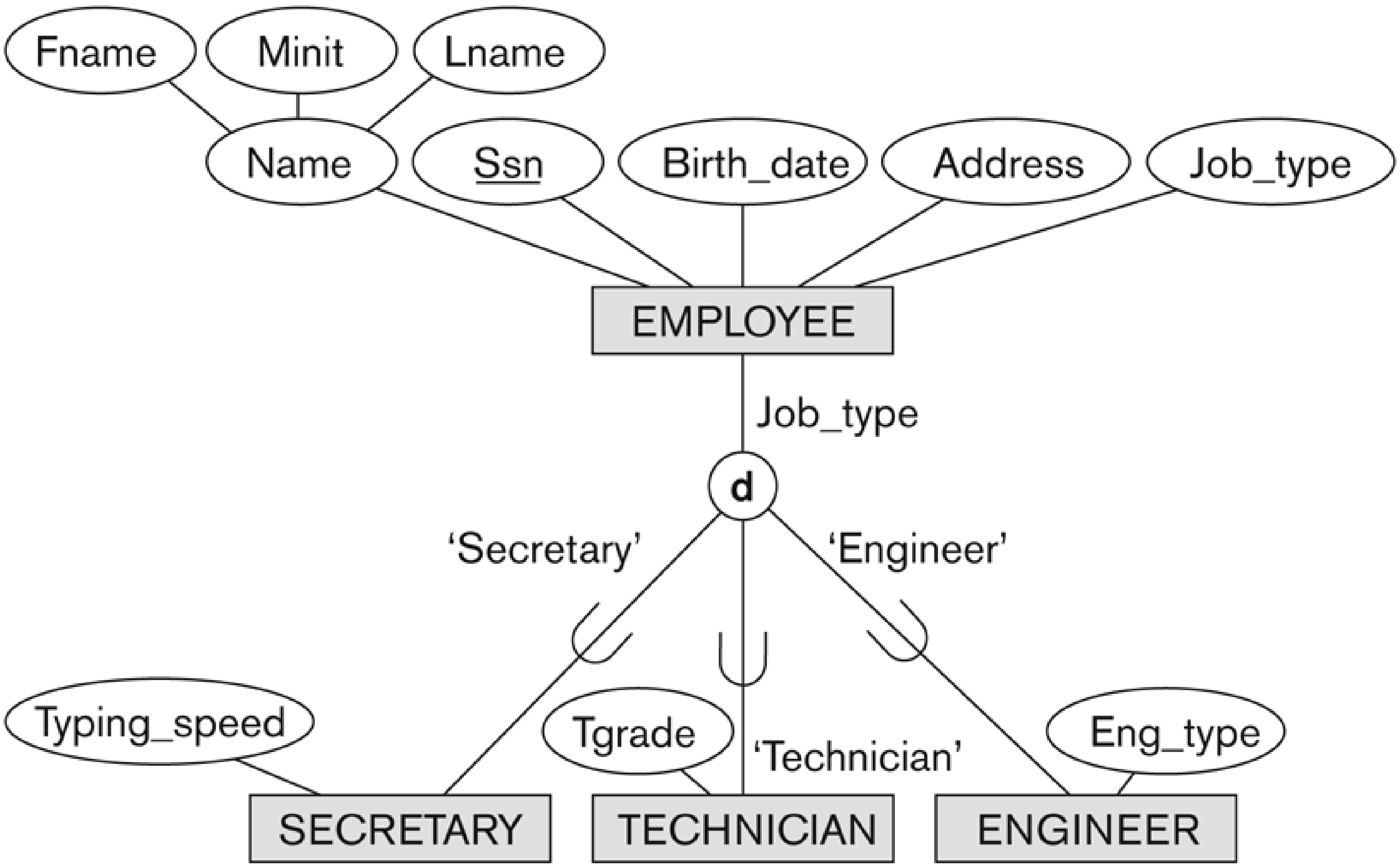
TRUCK

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

EER-to-Relational Mapping

- ***Option 8C: Single relation with one type attribute***
  - Create a single relation containing all the attributes of the superclass and its subclasses.
  - Add/assign a type(discriminating) attribute that indicates the subclass to which each tuple belongs.
  - This option only works for a disjoint specialization.

Example


Fname
Minit
Lname
Name
Ssn
Birth_date
Address
Job_type
EMPLOYEE
Job_type
d
'Secretary'
'Engineer'
Typing_speed
Tgrade
'Technician'
Eng_type
SECRETARY
TECHNICIAN
ENGINEER

Using Option 8C

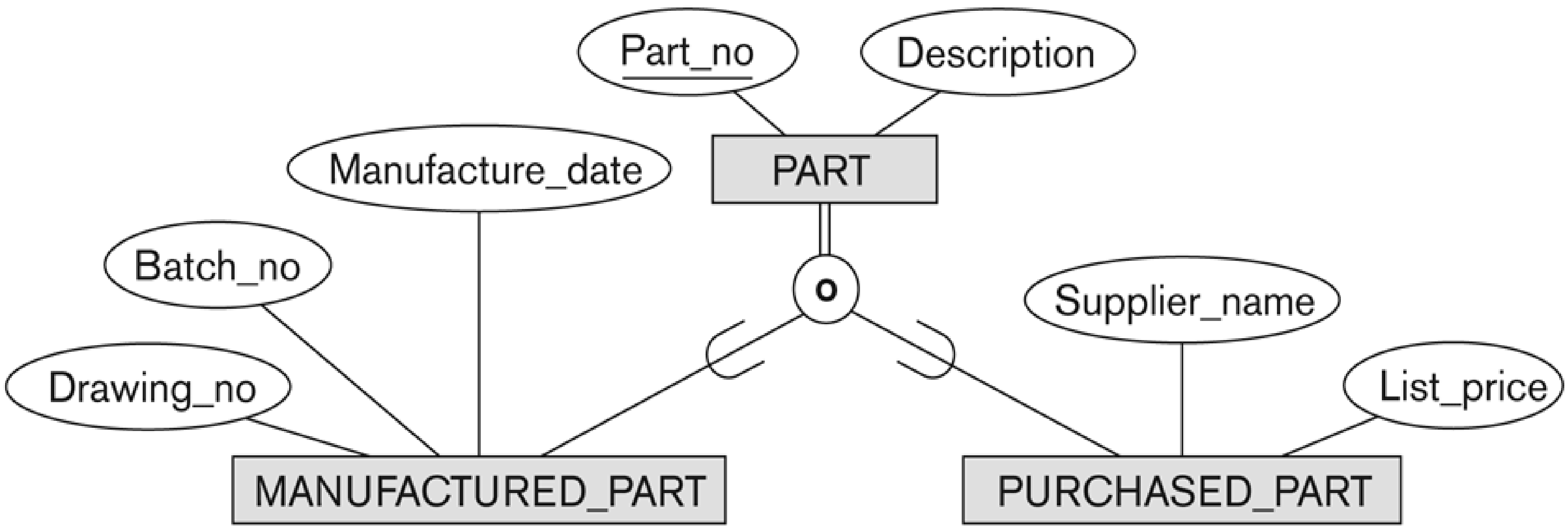
EMPLOYEE

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

EER-to-Relational Mapping

- **Option 8D: Single relation with multiple type attributes**
  - Create a single relation containing all the attributes of the superclass and its subclasses.
  - Add/assign multiple type attributes that indicate the subclass(es) to which each tuple may belong.
  - Each type attribute is of boolean datatype.
  - This option works for both overlapping and disjoint specializations.

Using Option 8D



Using Option 8D

PART

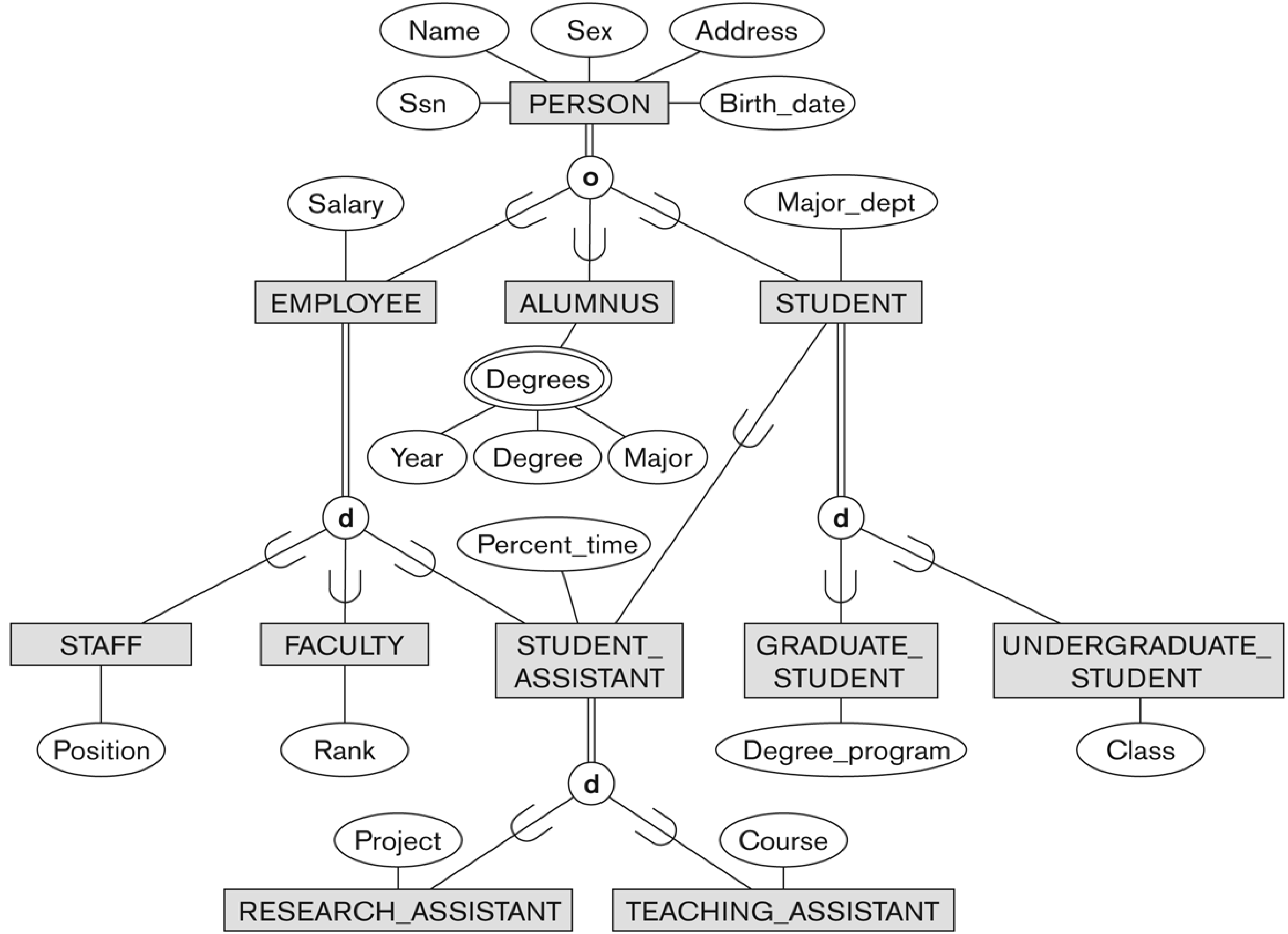
PartNo	Description	MFlag	DrawingNo	ManufactureDate	BatchNo	PFlag	SupplierName	ListPrice

EER-to-Relational Mapping

- ***Options 8A*** and ***8B***: called as ***multiple-relation options***
- ***Options 8C*** and ***8D***: called as ***single-relation options***
- ***Options 8C*** and ***8D***:
  - not recommended if there are many specific attributes

EER-to-Relational Mapping

- ***Mapping of Shared Subclasses (Multiple Inheritance)***
  - Any of the options discussed in Step 8 can be applied to a shared subclass.


Name
Sex
Address
Ssn
PERSON
Birth_date
o
Salary
Major_dept
EMPLOYEE
ALUMNUS
STUDENT
Degrees
Year
Degree
Major
d
Percent_time
d
STAFF
FACULTY
STUDENT_ ASSISTANT
GRADUATE_ STUDENT
UNDERGRADUATE_ STUDENT
Position
Rank
Degree_program
Class
d
Project
Course
RESEARCH_ASSISTANT
TEACHING_ASSISTANT

With Shared Subclass

- Both 8C and 8D are used for the shared subclass STUDENT_ASSISTANT.

PERSON

SSN	Name	BirthDate	Sex	Address

EMPLOYEE

SSN	Salary	EmployeeType	Position	Rank	PercentTime	RAFlag	TAFlag	Project	Course

ALUMNUS

SSN

ALUMNUS_DEGREES

SSN	Year	Degree	Major

STUDENT

SSN	MajorDept	GradFlag	UndergradFlag	DegreeProgram	Class	StudAssistFlag

Example of a Relation

Relation Name

Attributes

STUDENT

Tuples

Name	No	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Formal Definitions - Schema

- The ***Schema*** (or description) of a Relation:
  - Denoted by R(A1, A2, ...An)
  - R is the ***name*** of the relation
  - A1, A2, ..., An are the ***attributes*** of the relation are
- Example: CUSTOMER (Cust-id, Cust-name, Address, Phone#)

Formal Definitions - Schema

- ***Domain of an attribute***
  - a set of valid values of an attribute
    - For example, the domain of Cust-name is a string with a maximum length of 25
  - has name or a logical definition:
    - Example: "Philippine_phone_numbers" are the set of 11 digit phone numbers valid in the Philippines

Formal Definitions - Schema

- **Domain of an attribute**
  - has a data-type or a format defined for it
    - Example: date can have a format like yyyy-mm-dd

Formal Definitions - Tuple

- an ordered set of values (enclosed in angled brackets ‘< … >’)
- Each value is derived from an appropriate ***domain***.
- Example: 4-tuple
  - <632895, "John Smith", "101 Main St. Atlanta, GA 30332", "(404) 894-2000">
- A relation is a ***set*** of such tuples (rows).

Formal Definitions - State

- denoted by r(R)
- is a subset of the Cartesian product of the domains of its attributes.
  - Given R(A1, A2, ..., An)
  - $r(R) \subset dom(A1) \times dom(A2) \times \ldots \times dom(An)$

Formal Definitions - Example

- Let R(A1, A2) be a relation schema:
  - Let dom(A1) = {0,1}
  - Let dom(A2) = {a,b,c}
- Then dom(A1) X dom(A2) is all possible combinations:
  - {<0,a> , <0,b> , <0,c>, <1,a>, <1,b>, <1,c> }
- r(R) could be {<0,a> , <0,b> , <1,c> }

Definition Summary

Informal Terms	***Formal Terms***
Table	Relation
Column Header	Attribute
All possible column values	Domain
Row	Tuple
Table Definition	Schema of a Relation
Populated Table	State of a Relation

Characteristics Of Relations

- Ordering of tuples in a relation r(R):
  - not considered to be ordered
- Ordering of attributes in a relation schema R (and of values within each tuple):
  - ordered

Example – A relation STUDENT

Relation Name

Attributes

STUDENT

Name	No	Home_phone	Address	Office_phone	Age	Gpa
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25

Tuples

Same relation state

STUDENT

Name	No	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21

Characteristics Of Relations

- ***Value in a tuple:***
  - atomic (indivisible)
  - must be from the domain of the attribute for that column
  - A special ***NULL*** value is used to represent values that are unknown or inapplicable to certain tuples.

Notation

- ***Component value*** of a tuple t:
  - t[Ai] or t.Ai
- ***subtuple*** of t containing the values of attributes Au, Av, ..., Aw, respectively in t
  - t[Au, Av, ..., Aw]

Example

STUDENT

Name	No	Home_phone	Address	Office_phone	Age	Gpa
Dick Davidson	422-11-2320	NULL	3452 Elgin Road	749-1253	25	3.53
Barbara Benson	533-69-1238	839-8461	7384 Fontana Lane	NULL	19	3.25
Rohan Panchal	489-22-1100	376-9821	265 Lark Lane	749-6492	28	3.93
Chung-cha Kim	381-62-1245	375-4409	125 Kirby Road	NULL	18	2.89
Benjamin Bayer	305-61-2435	373-1616	2918 Bluebonnet Lane	NULL	19	3.21

- If t represents the 4th tuple, then
  - t[Home_phone] = <375-4409>
  - t[Name, Home_phone, Office_phone] = <Chung-cha Kim, 375-4409, NULL>

Relational Model Constraints

- ***conditions*** that must hold on ***all*** valid relation states.
- Three main categories:
  - Inherent model-based or implicit constraints
  - Schema-based or explicit constraints
  - Application-based or semantic constraints

Inherent Model-based Constraints

- Based from some characteristics of relations
- Example:
  - Each value in a tuple is atomic
  - An unknown or inapplicable value should be represented using NULL

Schema-based Constraints

- Constraints that can be expressed in the schema of the relational model via the DDL
- Categories:
  - **Domain** constraint
  - **Key** constraints
  - **Entity integrity** constraint
  - **Not NULL** constraint
  - **Referential integrity** constraint

Domain Constraint

- Every value in a tuple must be from the ***domain of its attribute*** (or it could be NULL, if allowed for that attribute).

Key Constraints

- **Superkey** of relation R:
  - is a set of attributes of R which are sufficient to identify a unique tuple of each relation *r(R)*
  - Example:
    - EMPLOYEE(Emp_no, TIN, Name, Position, Salary)
    - {Emp_no, Name, Position}
    - {Emp_no}
    - {Position, Salary}
    - {TIN}

Key Constraints

- **Key** of R:
  - is a ***minimal*** superkey
  - is a superkey with the additional property that removing any attribute from K results in a set of attributes that is not a superkey anymore

Key Constraints

- Example: Consider the EMPLOYEE relation schema:
  - EMPLOYEE(Emp_no, TIN, Name, Position, Salary)
  - Possible superkeys of EMPLOYEE:
    - Superkey1 = {Emp_no, Name, Position}
    - Superkey2 = {Emp_no}
    - Superkey3 = {TIN}
  - {Emp_no, Name, Position} is a superkey but *not* a key.
  - Superkey2 and Superkey3 are keys.

Key Constraints

- In general:
  - Any ***key*** is a ***superkey*** (but not vice versa)
  - Any set of attributes that *includes a key* is a ***superkey***
- If a relation has several keys, each of the keys is called a ***candidate key***.
- A ***primary key PK*** is a candidate key chosen as the principal means of identifying tuples within a relation.
  - The primary key attributes are **underlined**.

Key Constraints

- A primary key is also used to ***reference*** the tuple from another tuple
  - General rule: Choose as primary key the smallest of the candidate keys (in terms of size) or whose value never or very rarely changes.

Emp_no as the primary key

EMPLOYEE

Emp_no	TIN	Name	Position	Salary

Entity Integrity

- The ***primary key attributes PK*** of each relation schema R in S cannot have null values in any tuple of r(R).
  - If PK has several attributes, null is not allowed in any of these attributes.

Not NULL

- Specifies that an attribute other than the members of the primary key cannot have NULL values

Referential Integrity

- Tuples in the ***referencing relation*** R1 have attributes FK (called ***foreign key attributes***) that reference the primary key attributes PK of the ***referenced relation*** R2.
  - A tuple t1 in R1 is said to reference a tuple t2 in R2 if t1[FK] = t2[PK].

Referential Integrity (or foreign key) Constraint

- Statement of the constraint
  - The value in the foreign key column (or columns) FK of the ***referencing relation*** R1 can be either:
    - (1) a value of an existing primary key value of a corresponding primary key PK in the referenced relation R2, <u>or</u>
    - (2) a null.

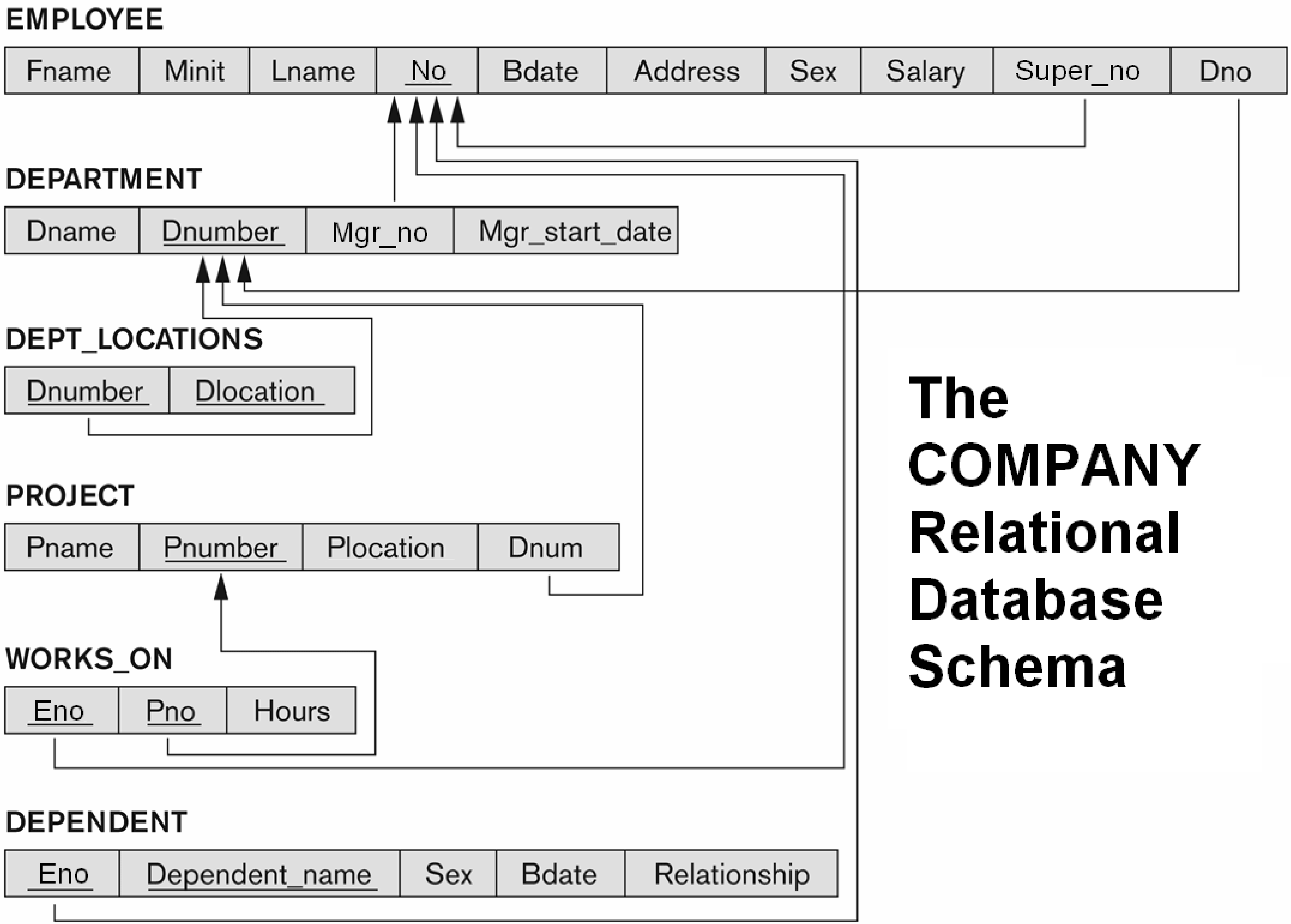
Application-based Constraints

- Based on application semantics and cannot be expressed by the model per se
  - Examples:
    - the max. no. of hours per employee for all projects he or she works on is 56 hrs per week
    - the salary of an employee should not exceed the salary of the employee's supervisor

Relational Database Schema

- ***Relational Database Schema S:***
  - A set of relation schemas S = {R1, R2, ..., Rn} and a set of integrity constraints
  - S is the name of the whole **database schema**
  - R1, R2, …, Rn are the names of the individual **relation schemas** within the database S

The COMPANY Relational Database Schema



Relational database state

- The ***relational database state*** is a union of all the individual relation states
- Whenever the database is changed, a new state arises
- Basic operations for changing the database:
  - INSERT a new tuple in a relation
  - DELETE an existing tuple from a relation
  - MODIFY an attribute of an existing tuple

EMPLOYEE

Fname	Minit	Lname	No	Bdate	Address	Sex	Salary	Super_no	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

DEPARTMENT

Dname	Dnumber	Mgr_no	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

WORKS_ON

Essn	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0

PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Stafford	4
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

DEPENDENT

Eno	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	Theodore	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	Michael	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter

Reference(s):

- **Elmasri, R. and S.B. Navathe.** 2010. Fundamentals of Database Systems. 6th Edition. Addition Wesley. ISBN-13: 978-0-136-08620-8
- **Elmasri, R. and S.B. Navathe.** 2007. Fundamentals of Database Systems. 5th Edition. Addition Wesley. ISBN: 981-06-9800-3