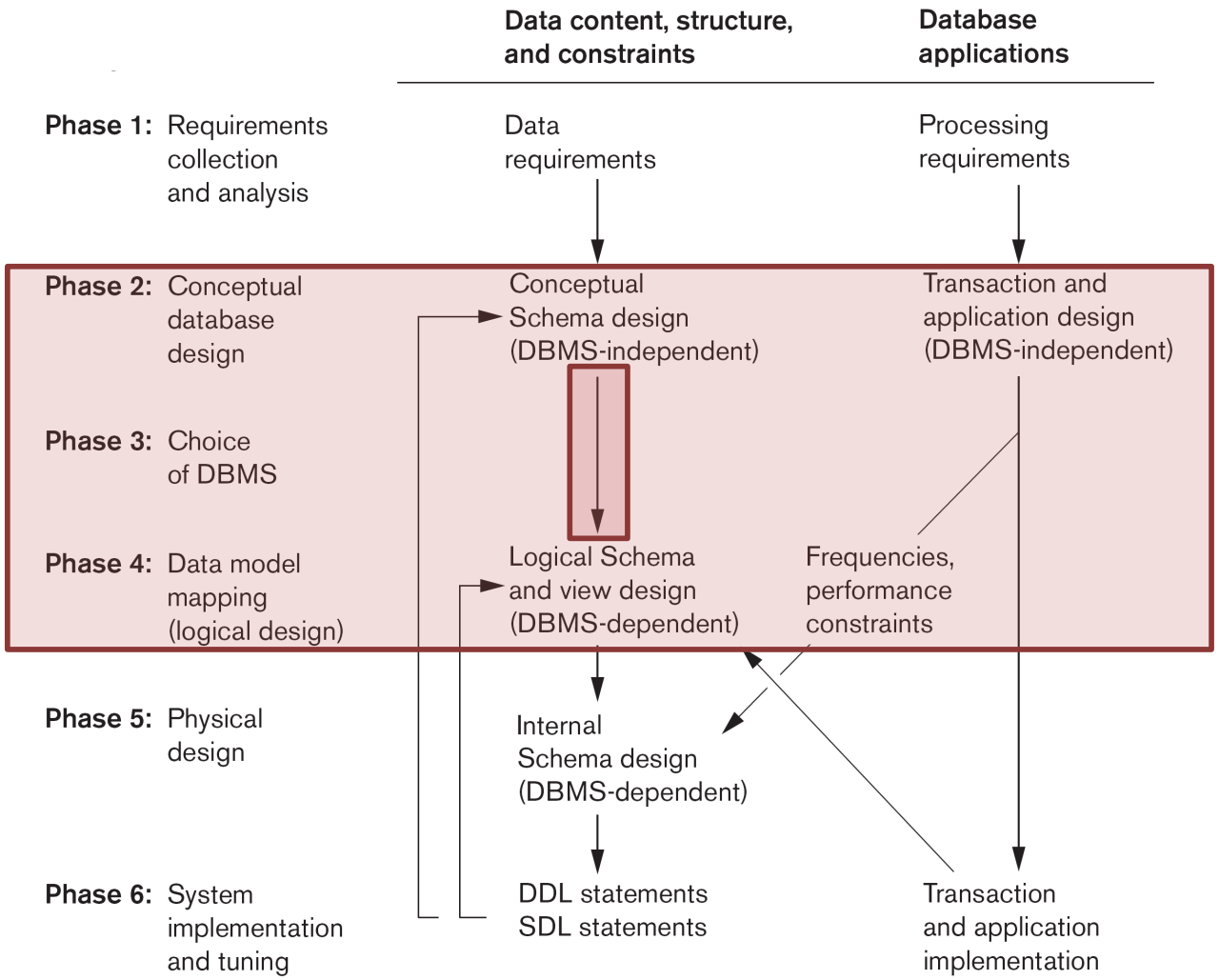
ER-to-Relational Mapping

Lecture 9

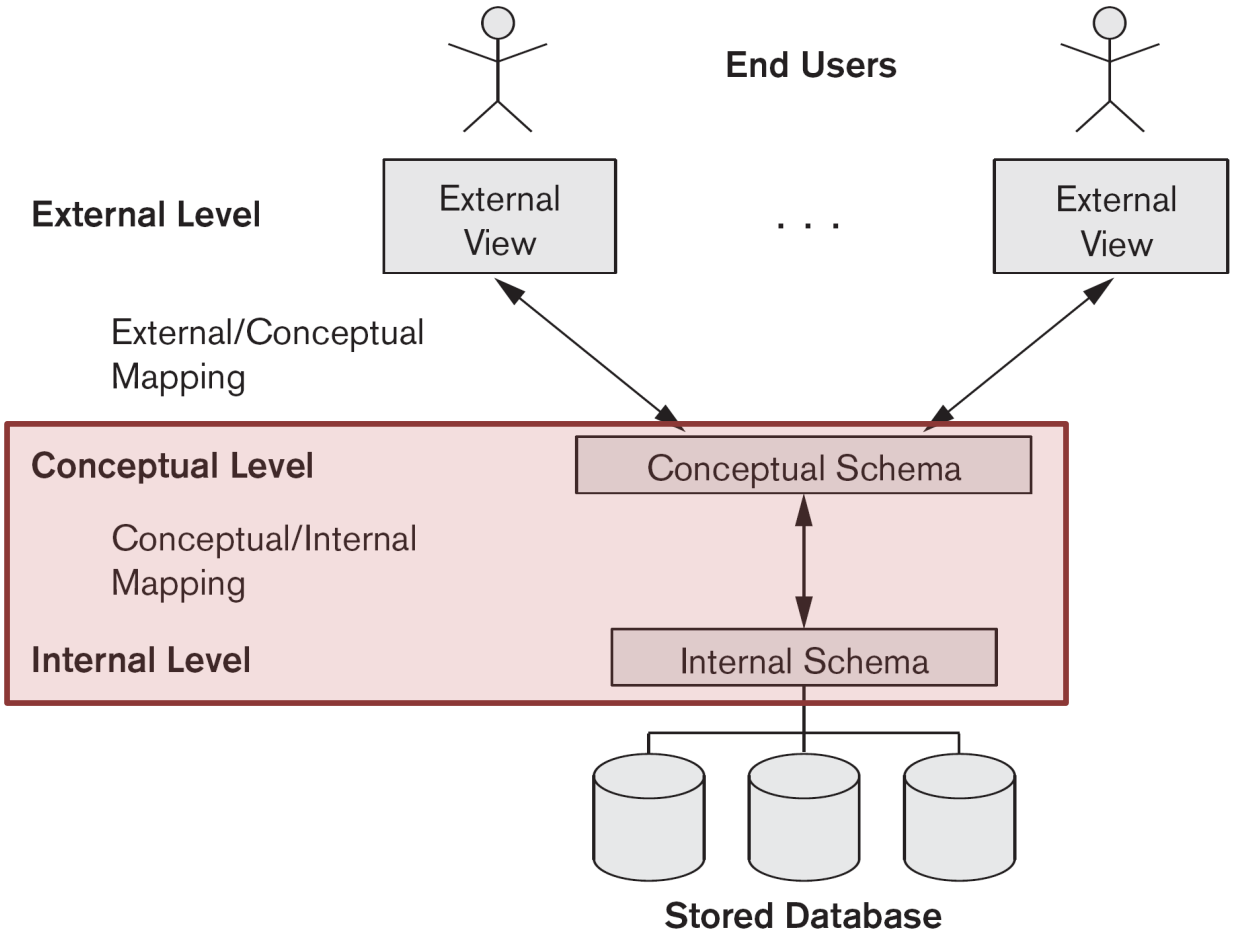
Outline

1. Context
2. The Algorithm

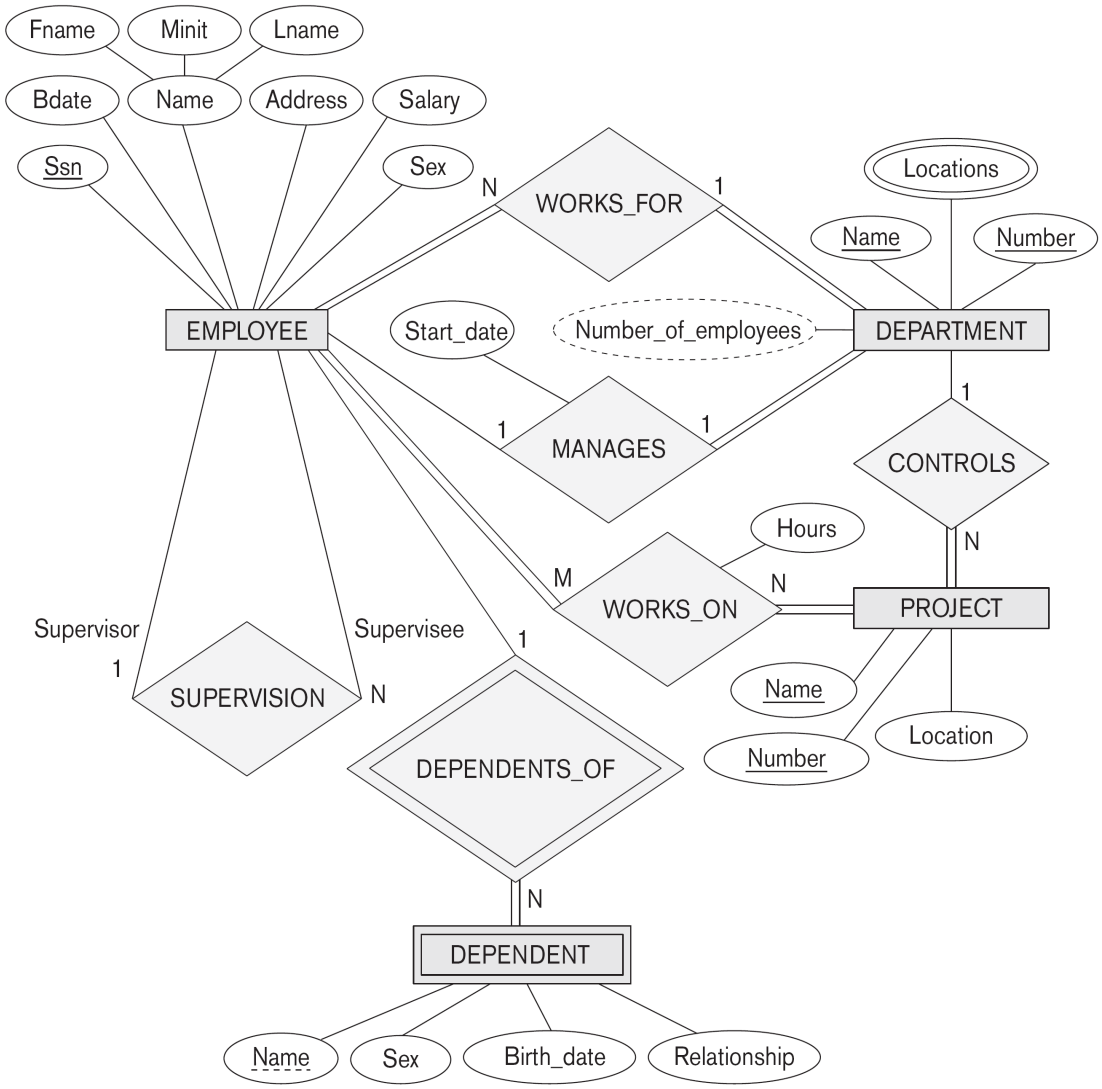
Database Design and Implementation Process



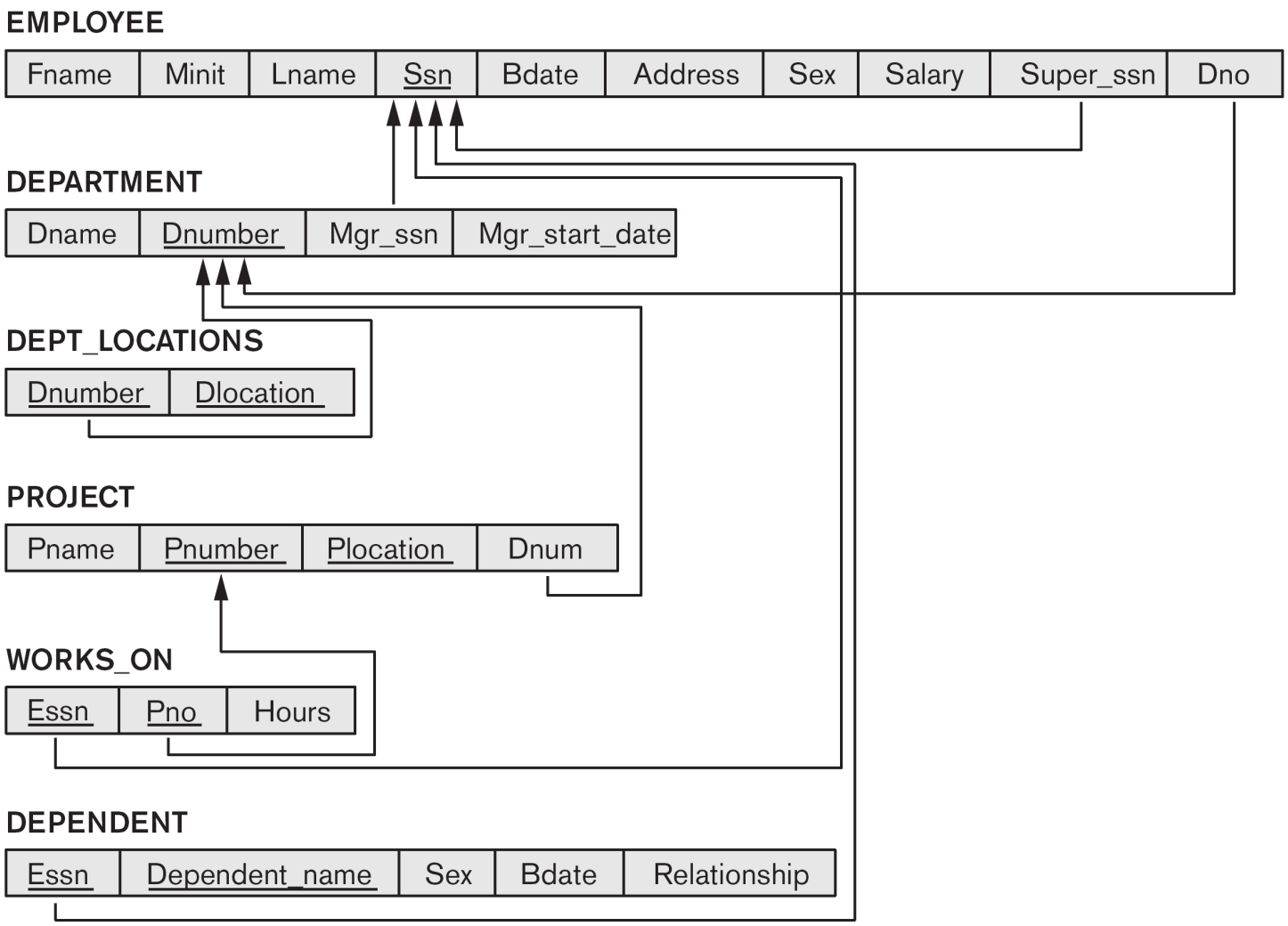
Data Models


End Users
External Level
External View
. . .
External View
External/Conceptual Mapping
Conceptual Level
Conceptual Schema
Conceptual/Internal Mapping
Internal Level
Internal Schema
Stored Database

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Resulting Relational Schema



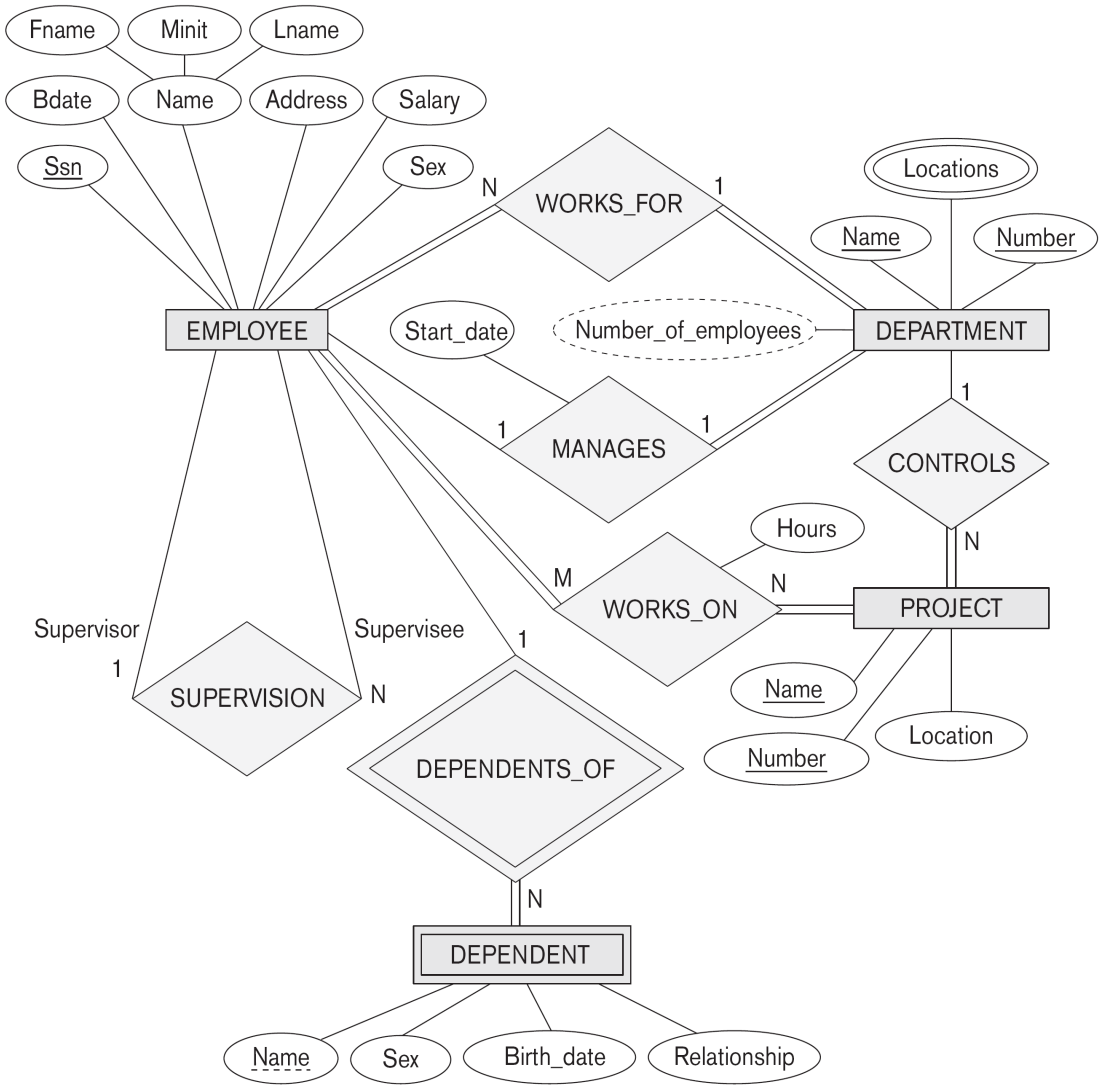
Step 1: Regular Entity Types

i. For each regular/strong entity type, create a corresponding relation that includes all the simple attributes (includes simple attributes of composite relations)

ii. Choose one of the key attributes as primary
   - If composite, the simple attributes together form the primary key

iii. Any remaining key attributes are kept as secondary unique keys (these will be useful for physical tuning w.r.t. indexing analysis)

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
N
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
MANAGES
1
1
1
CONTROLS
Hours
M
N
WORKS_ON
N
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 1 Result

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

PROJECT

Pname	Pnumber	Plocation

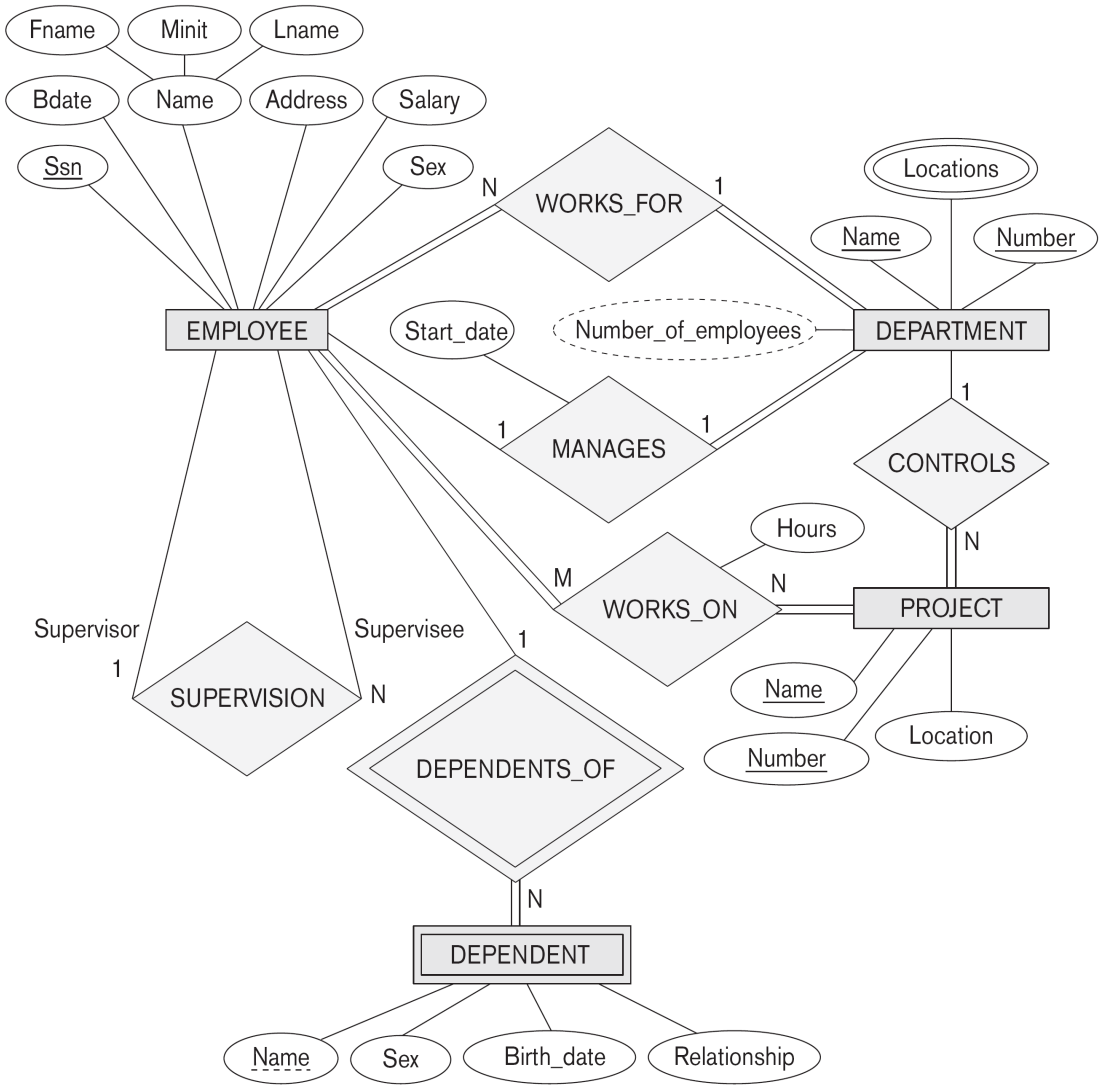
Step 2: Weak Entity Types

i. For each weak entity type, create a corresponding relation that includes all the simple attributes

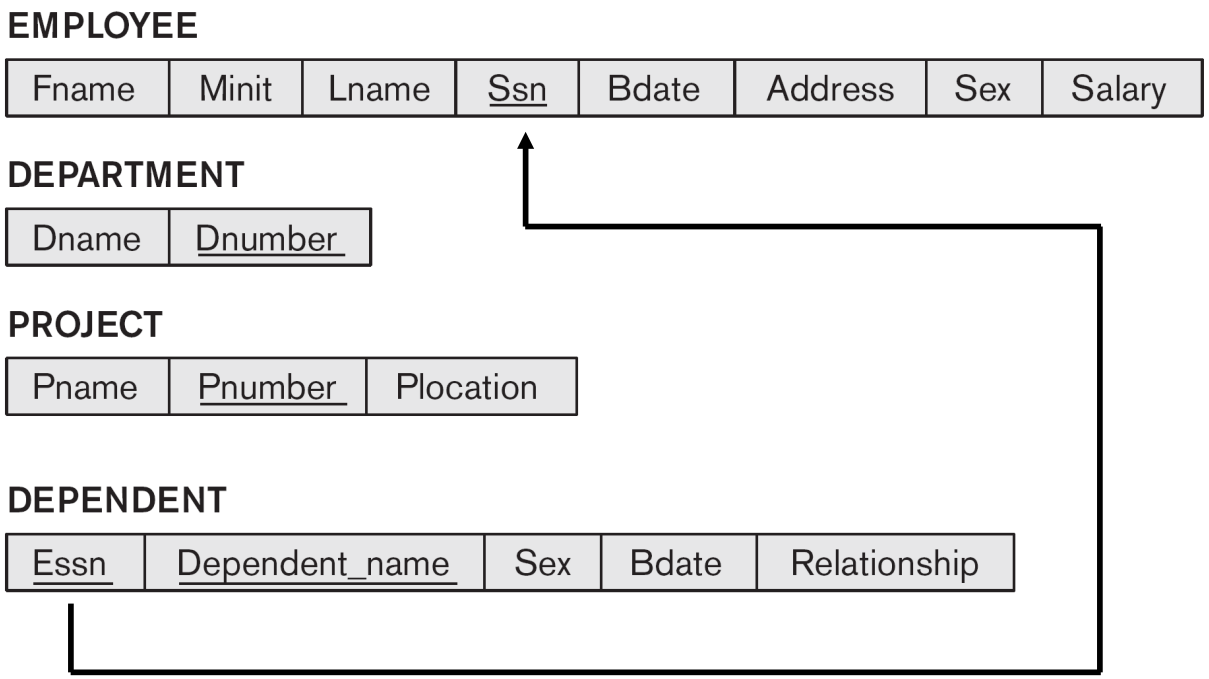
ii. Add as a foreign key all of the primary key attribute(s) in the entity corresponding to the owner entity type

iii. The primary key is the combination of all the primary key attributes from the owner and the partial key of the weak entity, if any

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
N
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
MANAGES
1
1
1
CONTROLS
Hours
M
N
WORKS_ON
N
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 2 Result



Step 3: Mapping Binary 1-to-1

Three approaches

- **Foreign Key**
  - Usually appropriate
- Merged Relation
  - Possible when both participations are total
- Relationship Relation
  - Not discussed

Step 3: Mapping Binary 1-to-1

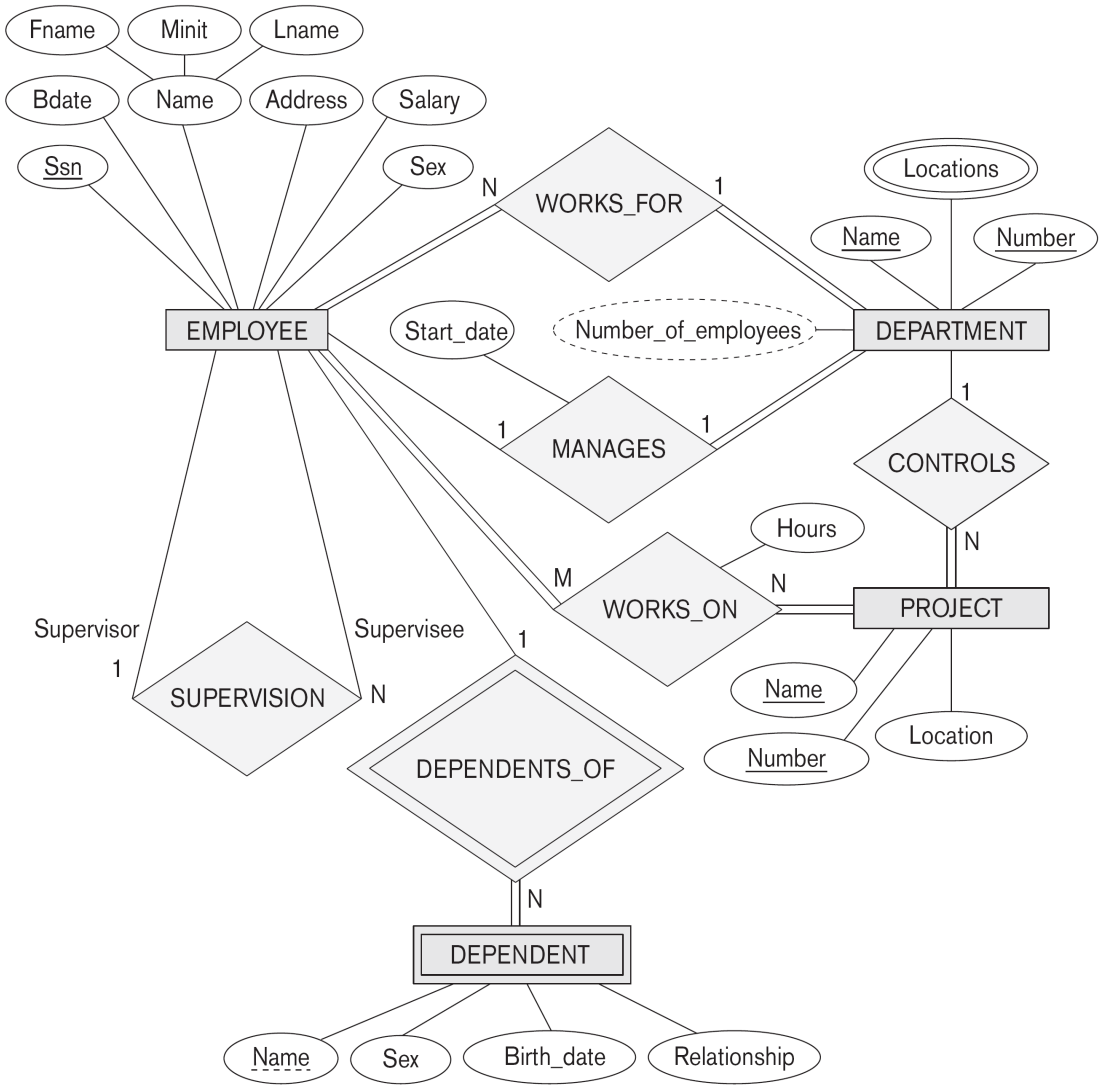
Foreign Key

i. Choose one relation as *S*, the other *T*
   - Better if S has total participation (reduces number of NULL values)

ii. Add to *S* all the simple attributes of the relationship

iii. Add as a foreign key in *S* the primary key attributes of *T*

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
N
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
MANAGES
1
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 2 Result

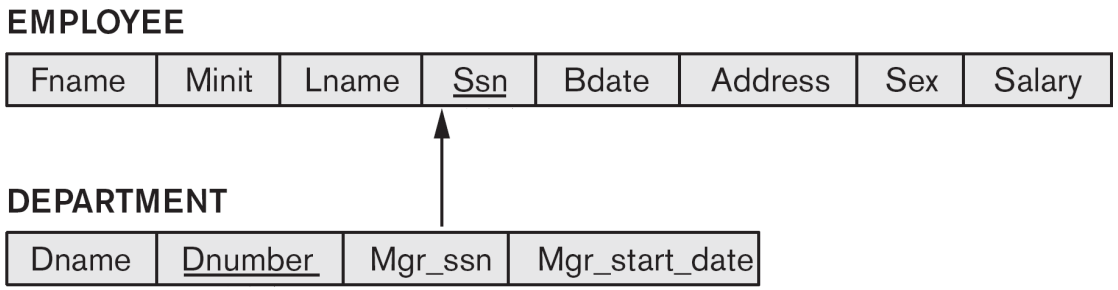
EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary

DEPARTMENT

Dname	Dnumber

Step 3 Result



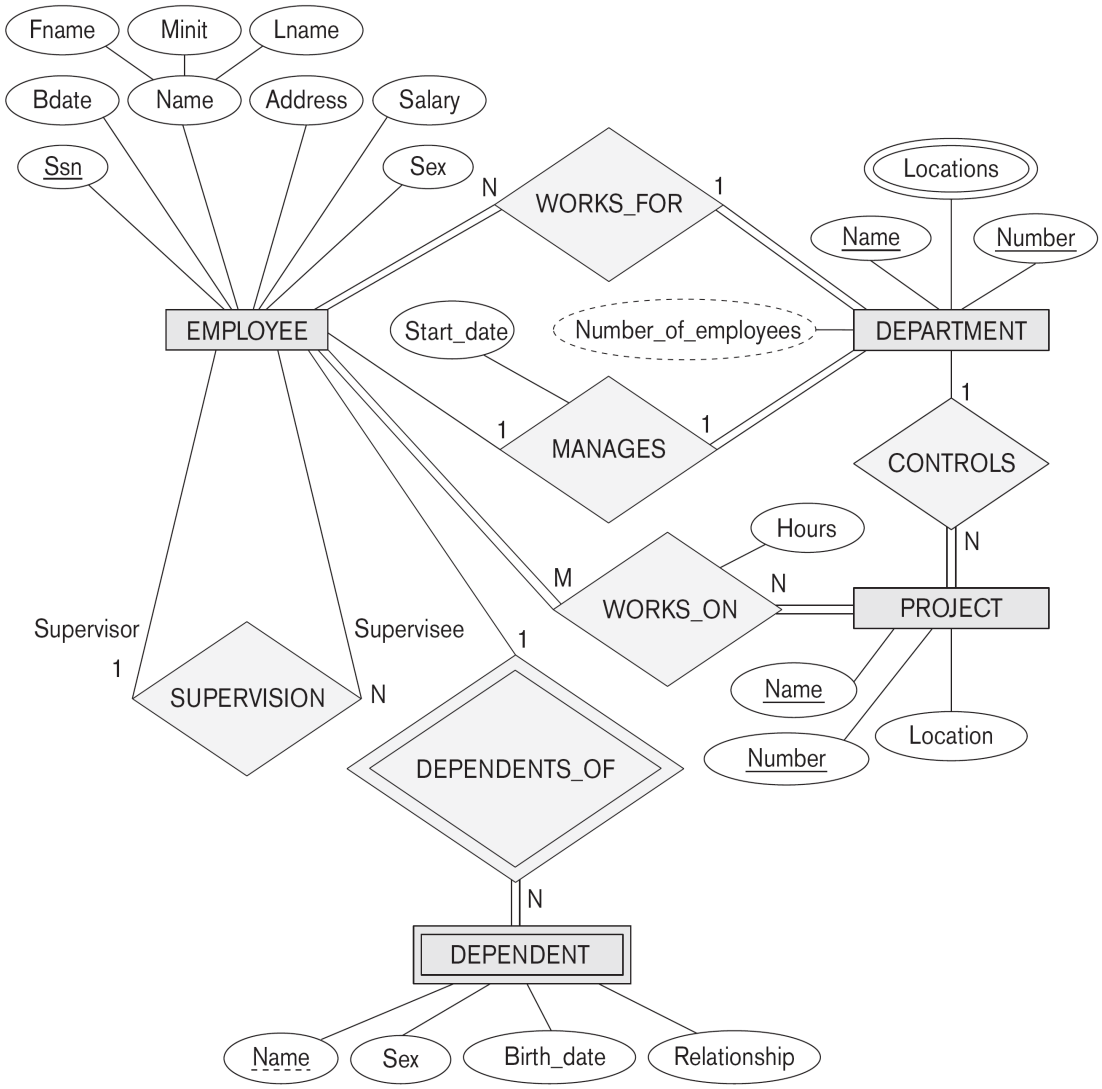
Step 4: Binary 1-to-N

i. Choose the *S* relation as the type at the N-side of the relationship, other is *T*

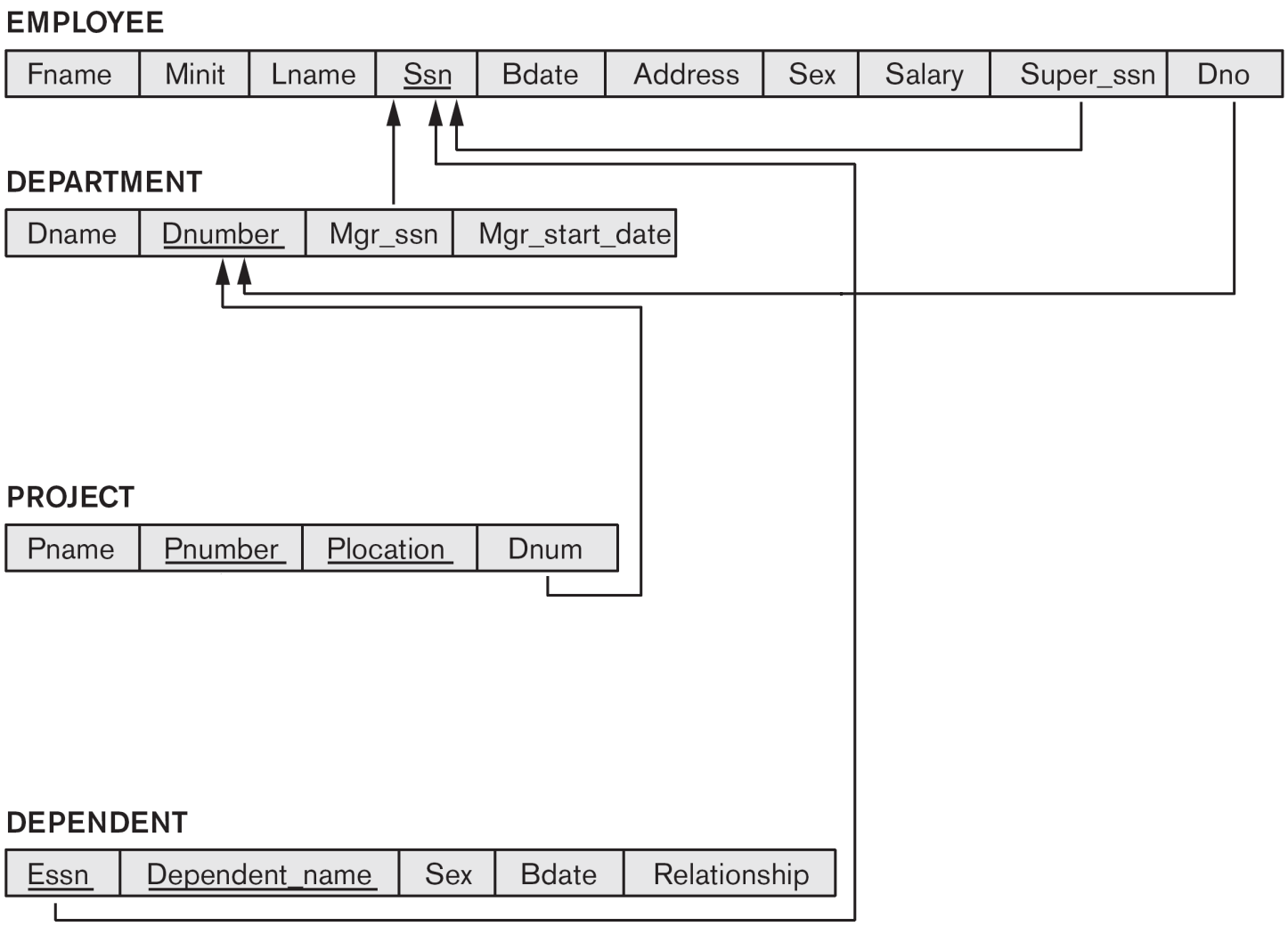
ii. Add as a foreign key to *S* all of the primary key attribute(s) of *T*

Another approach: create a relationship relation

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
N
WORKS_FOR
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
1
MANAGES
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 4 Result



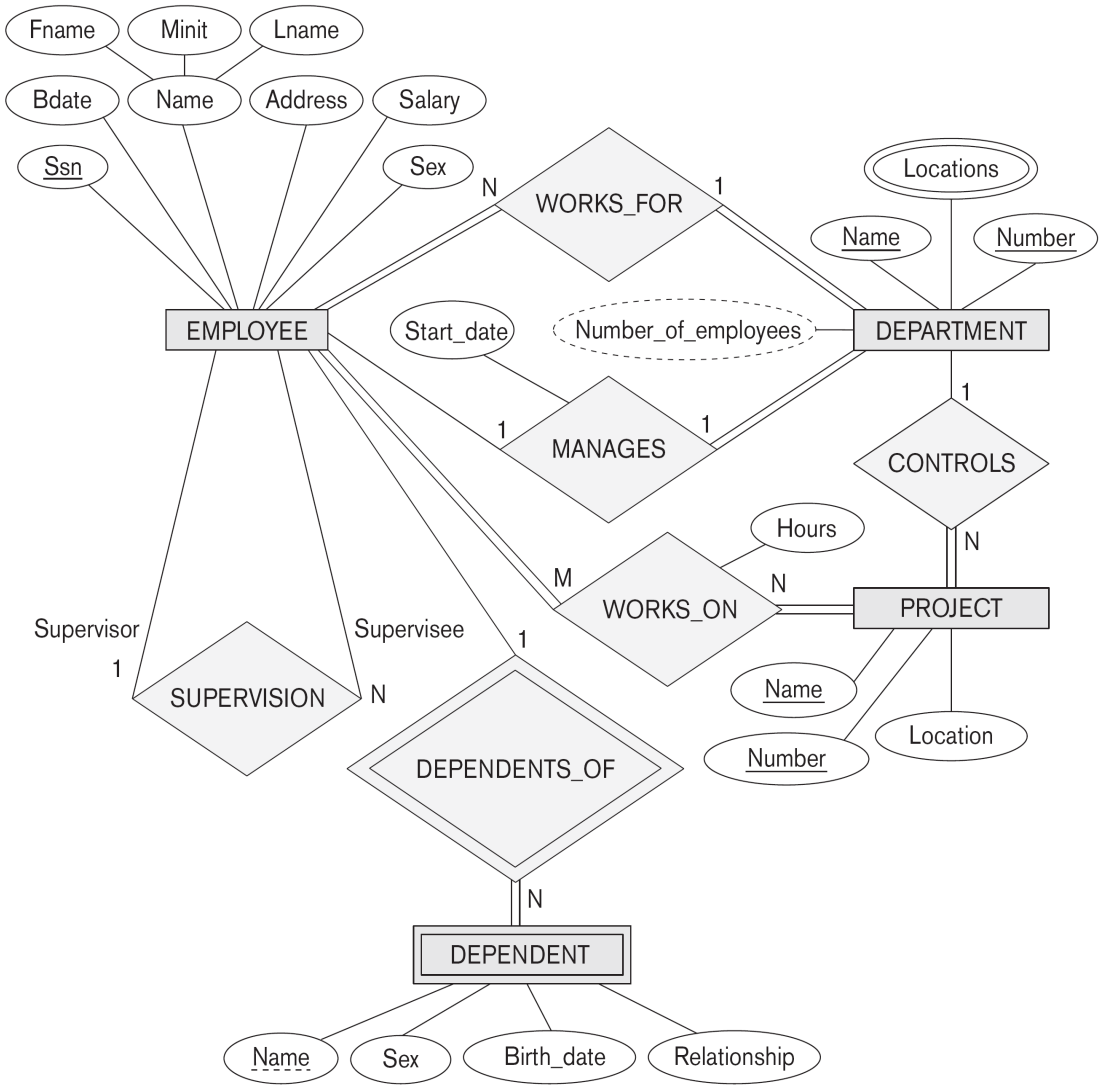
Step 5: Binary M-to-N

i. Create a new relation *S* (termed: *relationship relation*)
   - In some ERD dialects, actually drawn in

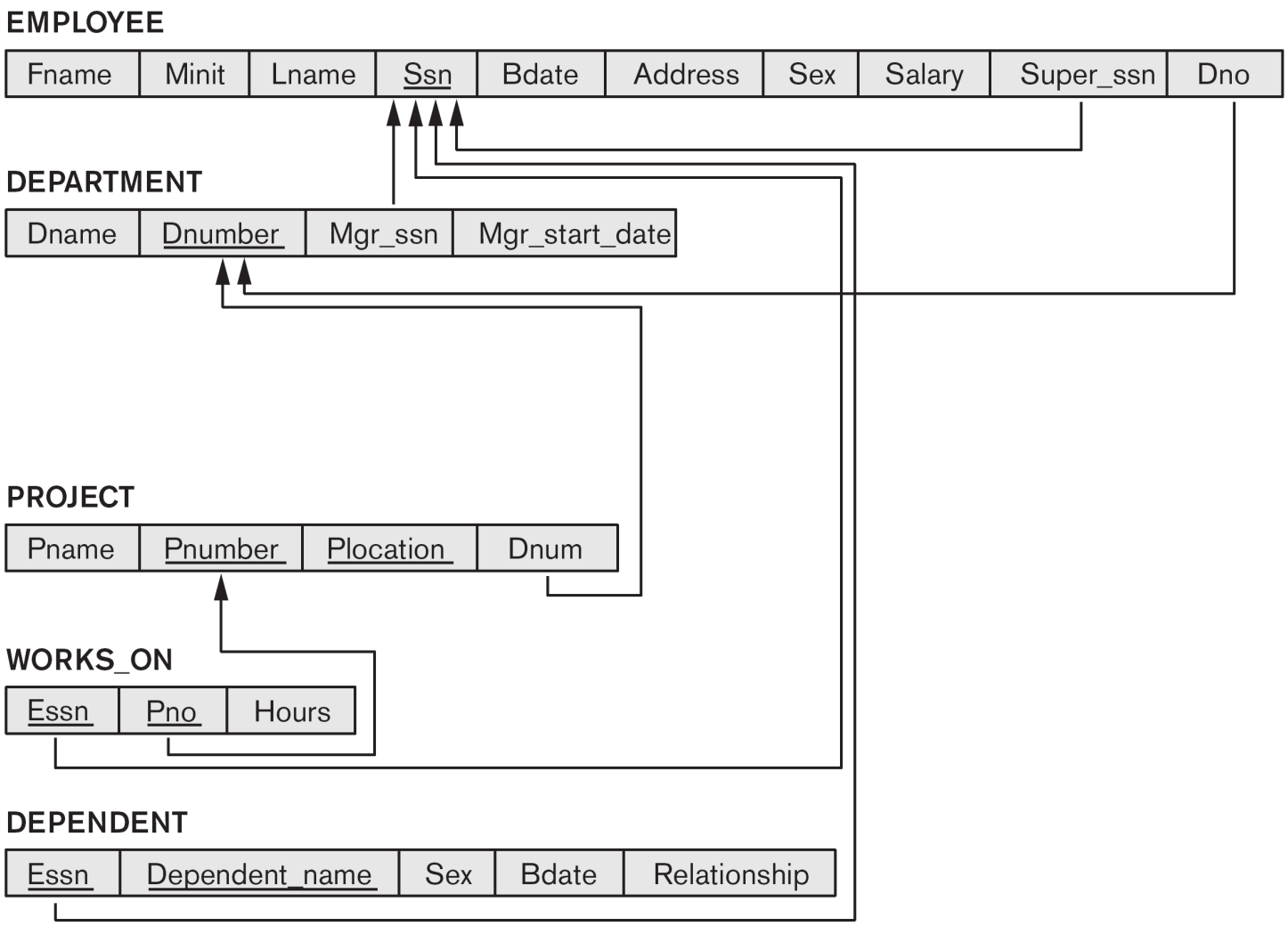
ii. Add as foreign keys the primary keys of both relations; their combination forms the primary key of *S*

iii. Add any simple attributes of the M:N relationship to *S*

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
N
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
MANAGES
1
1
CONTROLS
Hours
N
M
N
WORKS_ON
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 5 Result



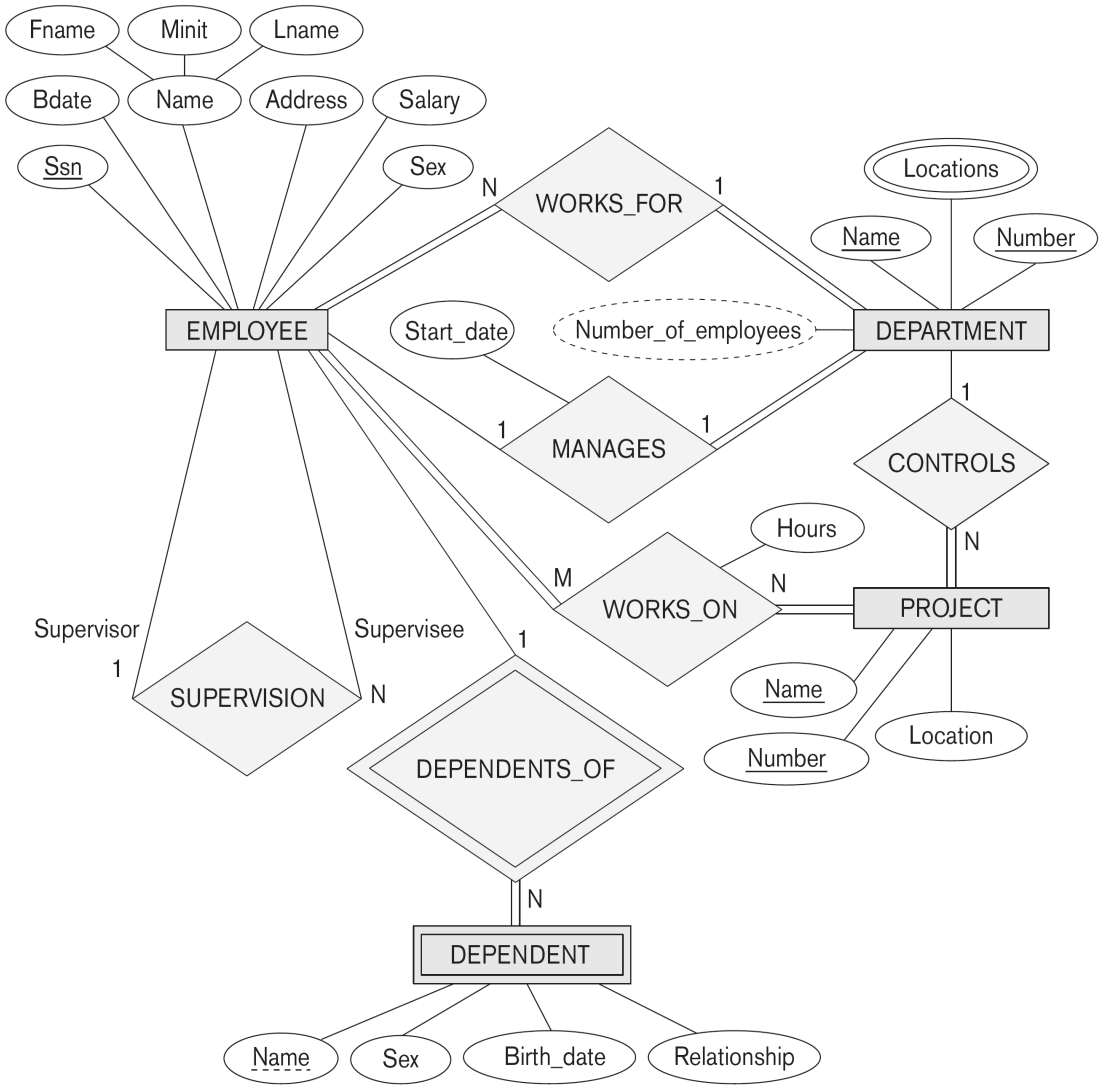
Step 6: Multivalued Attributes

i. Create a new relation *S*

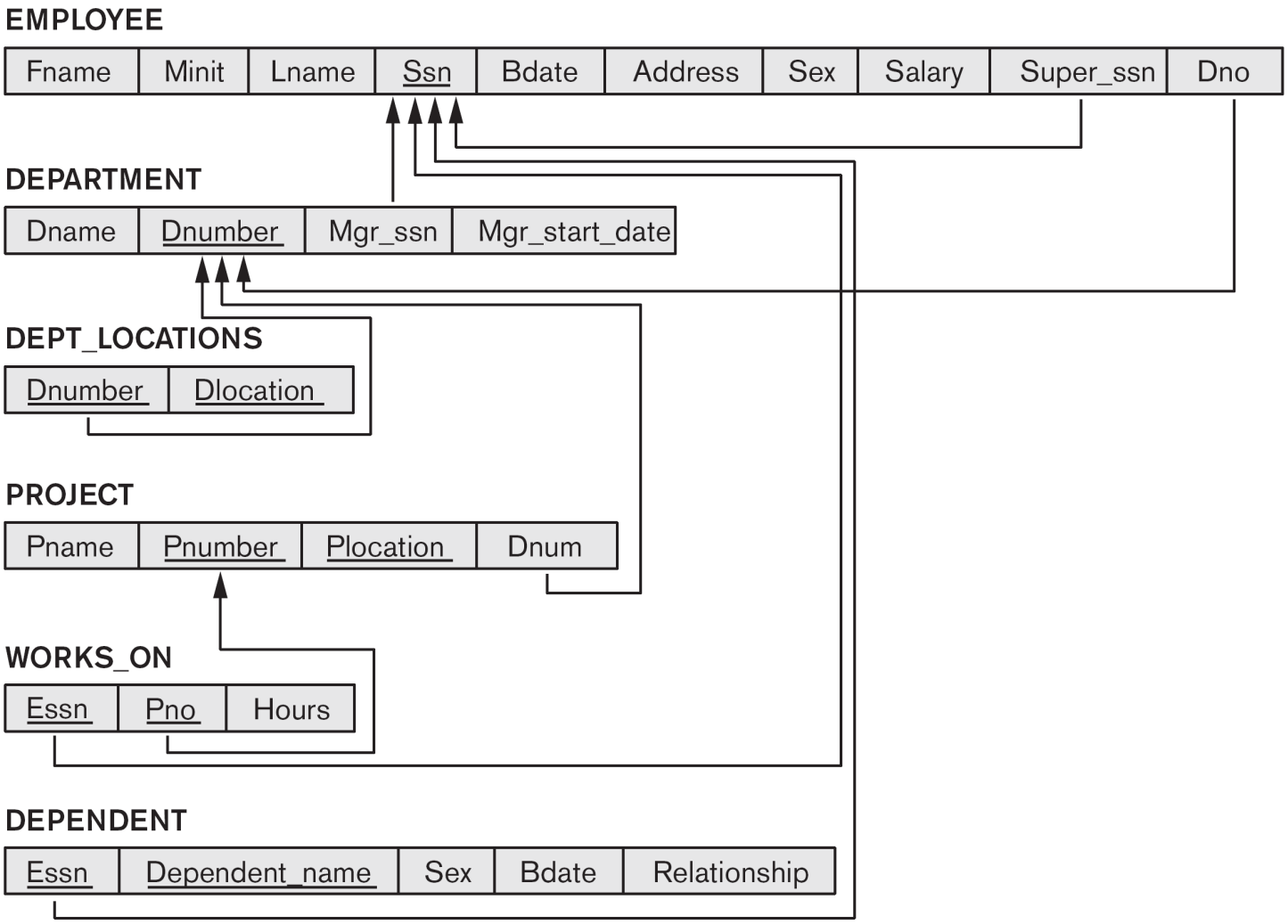
ii. Add as foreign keys the primary keys of the corresponding relation

iii. Add the attribute to *S* (if composite, the simple attributes); the combination of all attributes in *S* forms the primary key

Example ERD


Fname
Minit
Lname
Name
Bdate
Address
Salary
Ssn
Sex
WORKS_FOR
N
1
Locations
Name
Number
EMPLOYEE
Start_date
Number_of_employees
DEPARTMENT
1
MANAGES
1
1
CONTROLS
N
Hours
M
N
WORKS_ON
PROJECT
Supervisor
1
SUPERVISION
N
Supervisee
1
Name
Location
Number
DEPENDENTS_OF
N
DEPENDENT
Name
Sex
Birth_date
Relationship

Step 6 Result



Step 7: Specialization/Generalization

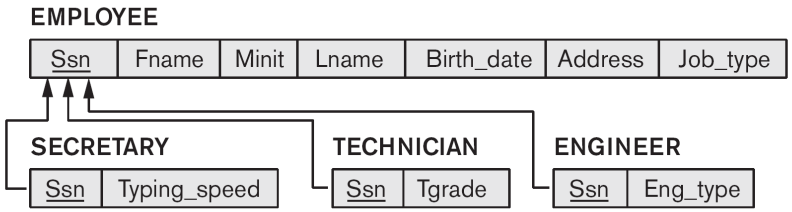
A. Multiple relations – subclass and superclass
   - Usually works (assumes unique id at parent)

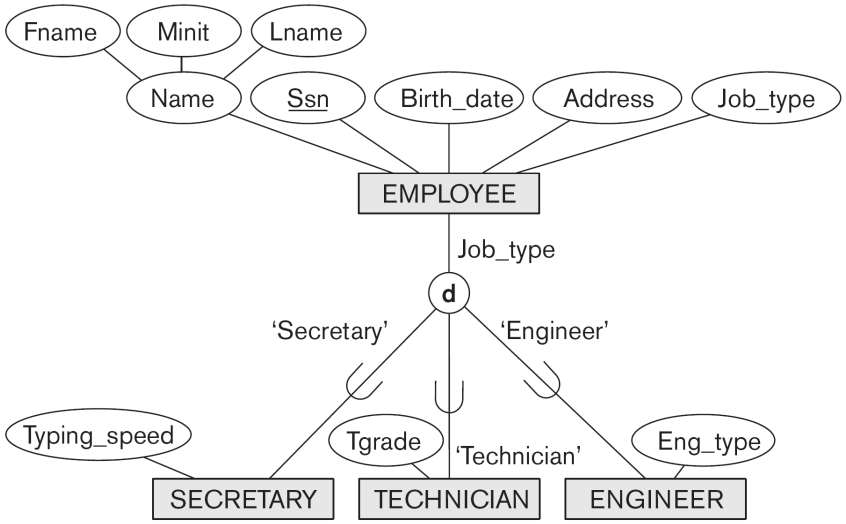
B. Multiple relations – subclass only
   - Should only be used for disjoint

C. Single relation with one type attribute
   - Only for disjoint, can result in many NULLs

D. Single relation with multiple type attributes
   - Better for overlapping, could be disjoint

Specialization/Generalization (A)





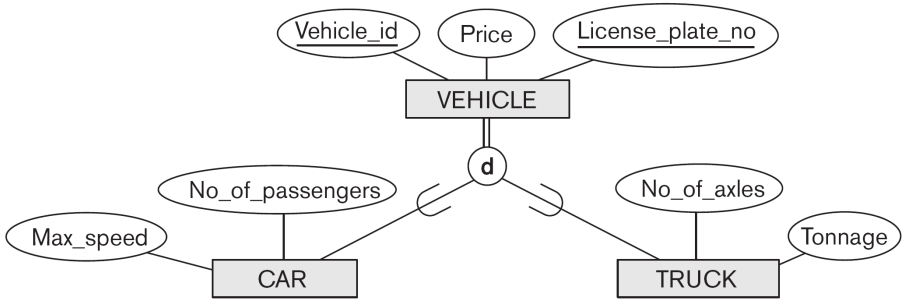
Specialization/Generalization (B)

CAR

Vehicle_id	License_plate_no	Price	Max_speed	No_of_passengers

TRUCK

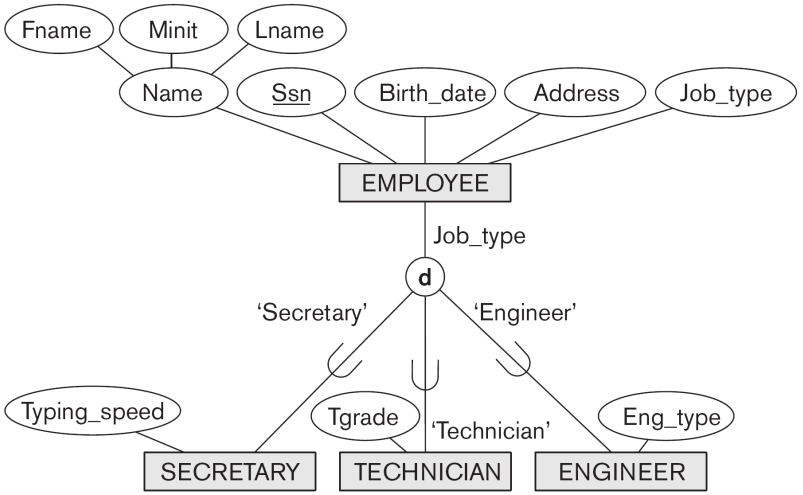
Vehicle_id	License_plate_no	Price	No_of_axles	Tonnage



Specialization/Generalization (C)

EMPLOYEE

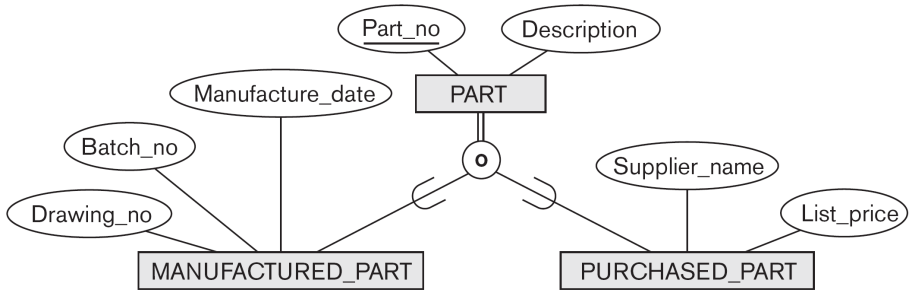
Ssn	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type



Specialization/Generalization (D)

PART

Part_no	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price



Summary

- Mapping from ERDs to relations is an algorithmic process

- Some choice points involve comparing time-space tradeoffs (more in physical design)