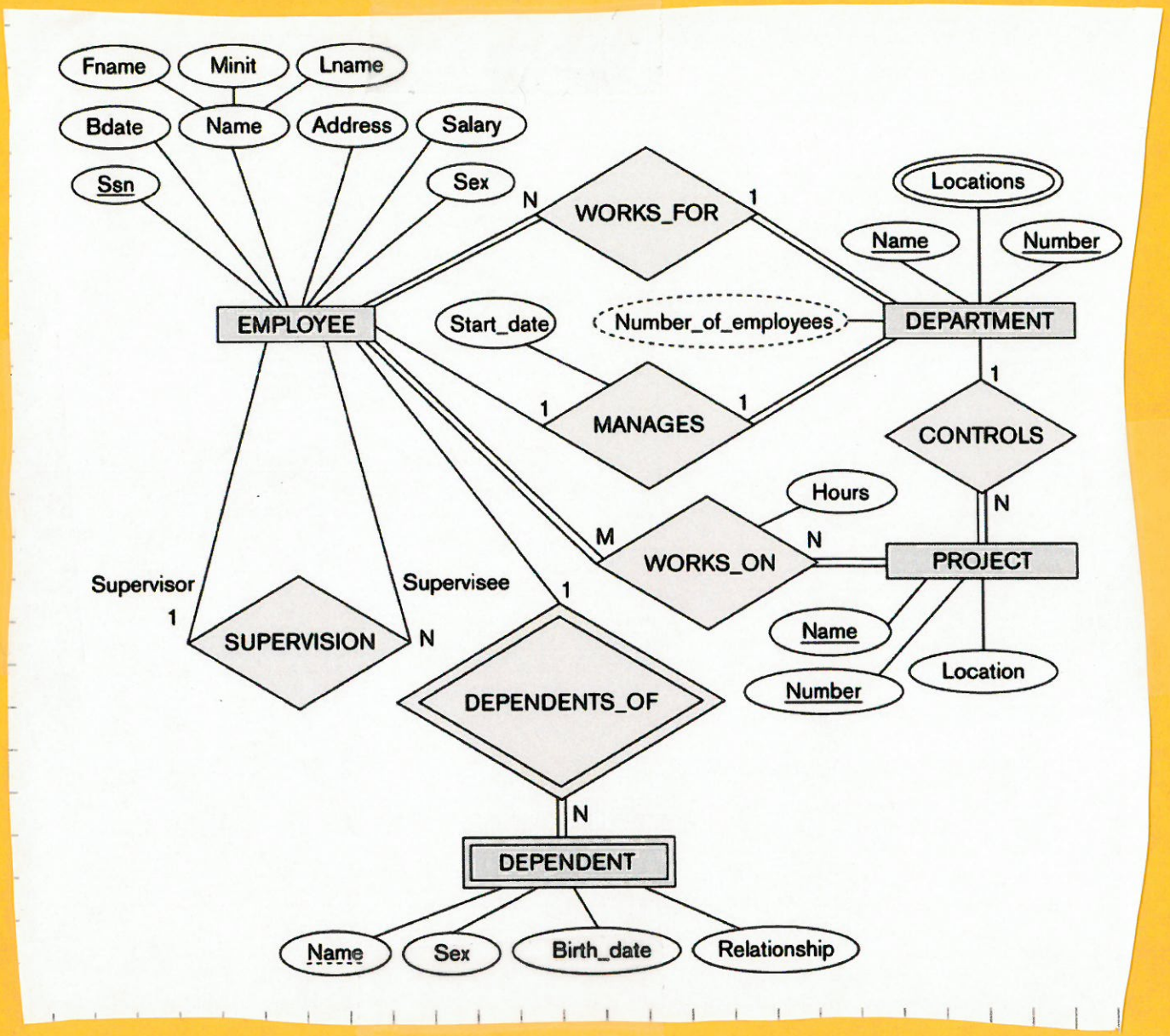
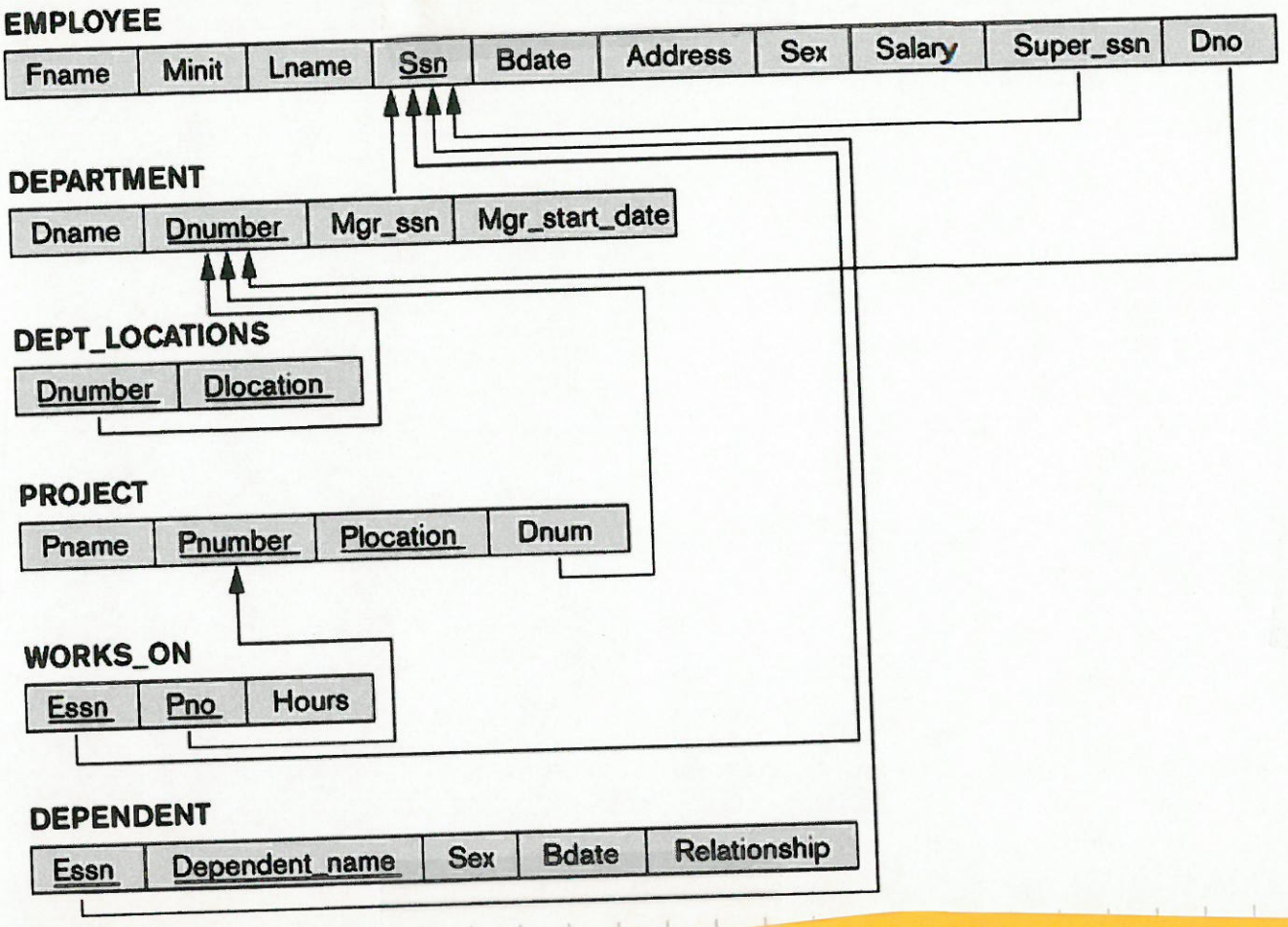
ER→ Relational Mapping Algo 9/24



Goal:

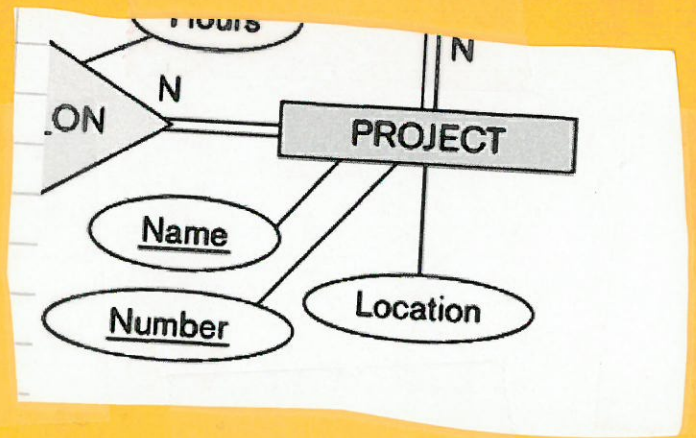


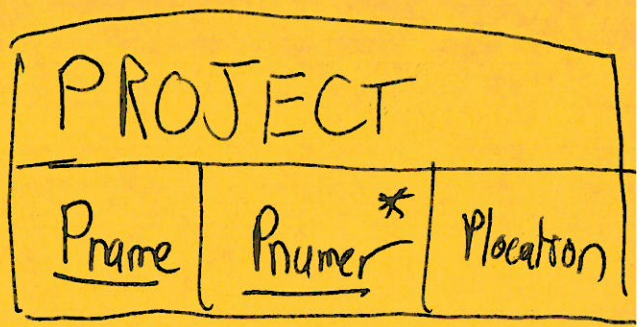
- tables w/ single valued attributes
- primary keys (other keys)
- refferential integrety constraints between relations

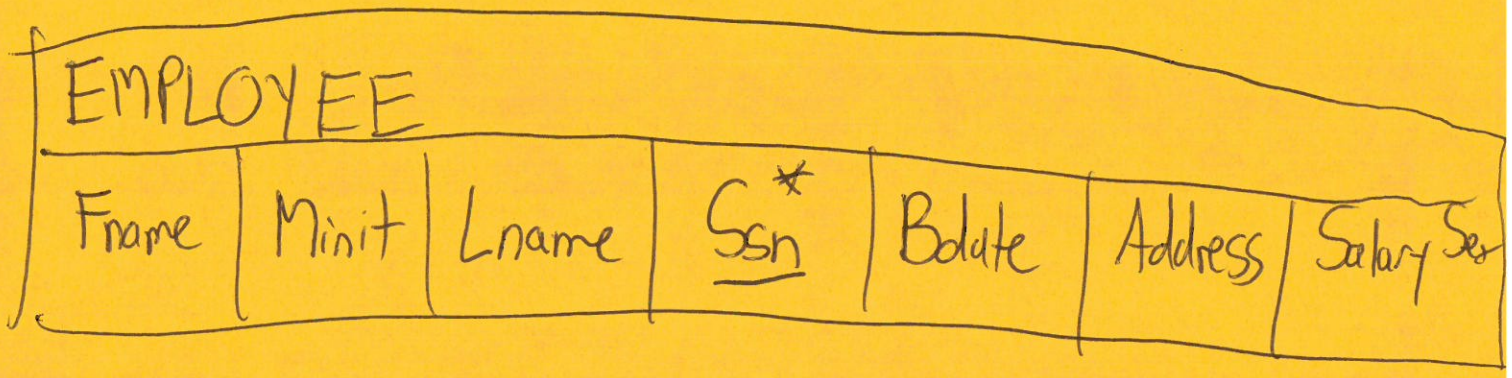
Step 1: Strong entity types

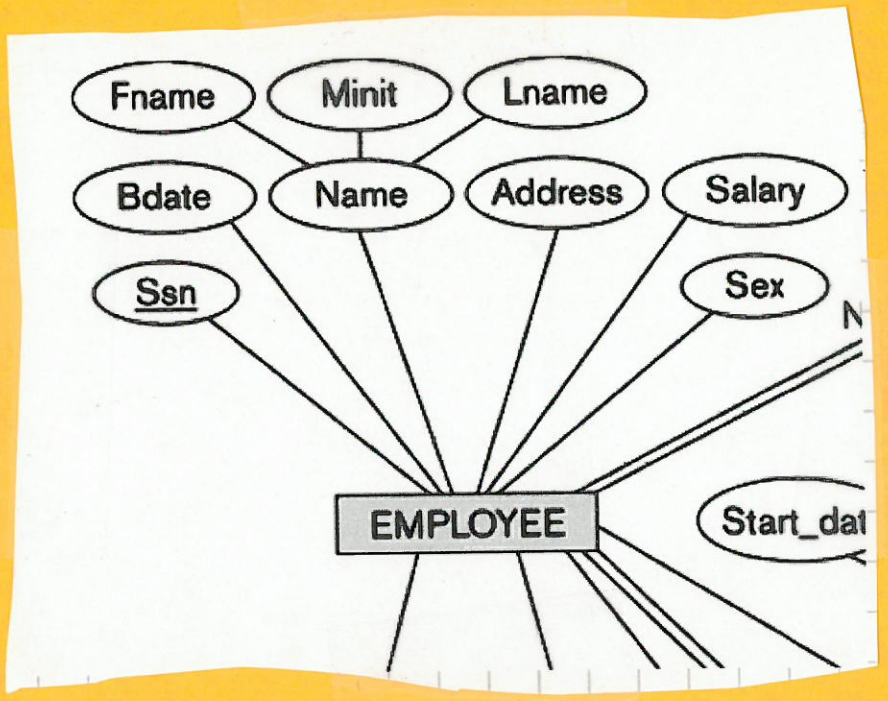
for each strong entity type E
create a relation R for E
add all simple attributes of E to R
for each composite attribute C
flatten and add C to R
(C: attributes of)
Choose key attribute of E
and set as primary key of R
For remaining key of E
add additional keys

Eg.






EMPLOYEE
Fname
Minit
Lname
Ssn
Bdate
Address
Salary
Sex


Fname
Minit
Lname
Bdate
Name
Address
Salary
Ssn
Sex
EMPLOYEE
Start_dat

Step 2: Weak entities

For each weak entity type W

E is the owner entity of W

(if E is weak type ~~add~~ process later)

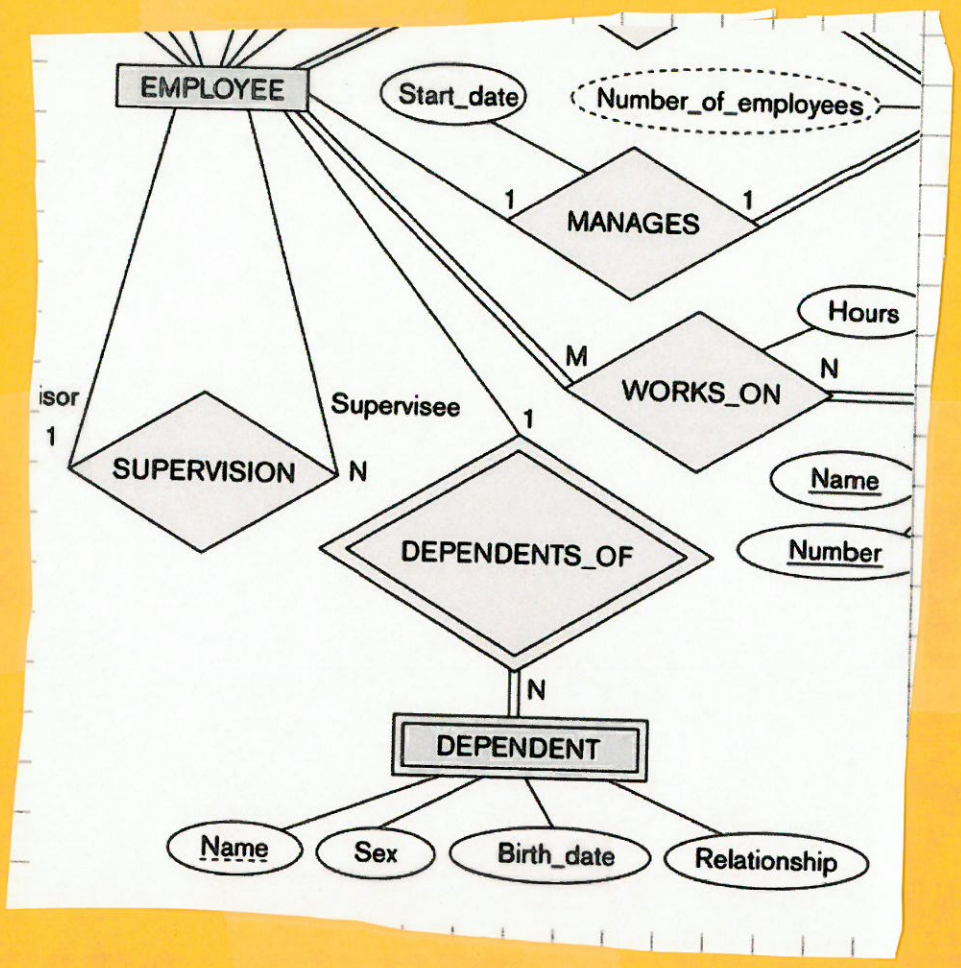
create a relation R for W (add attributes as before)

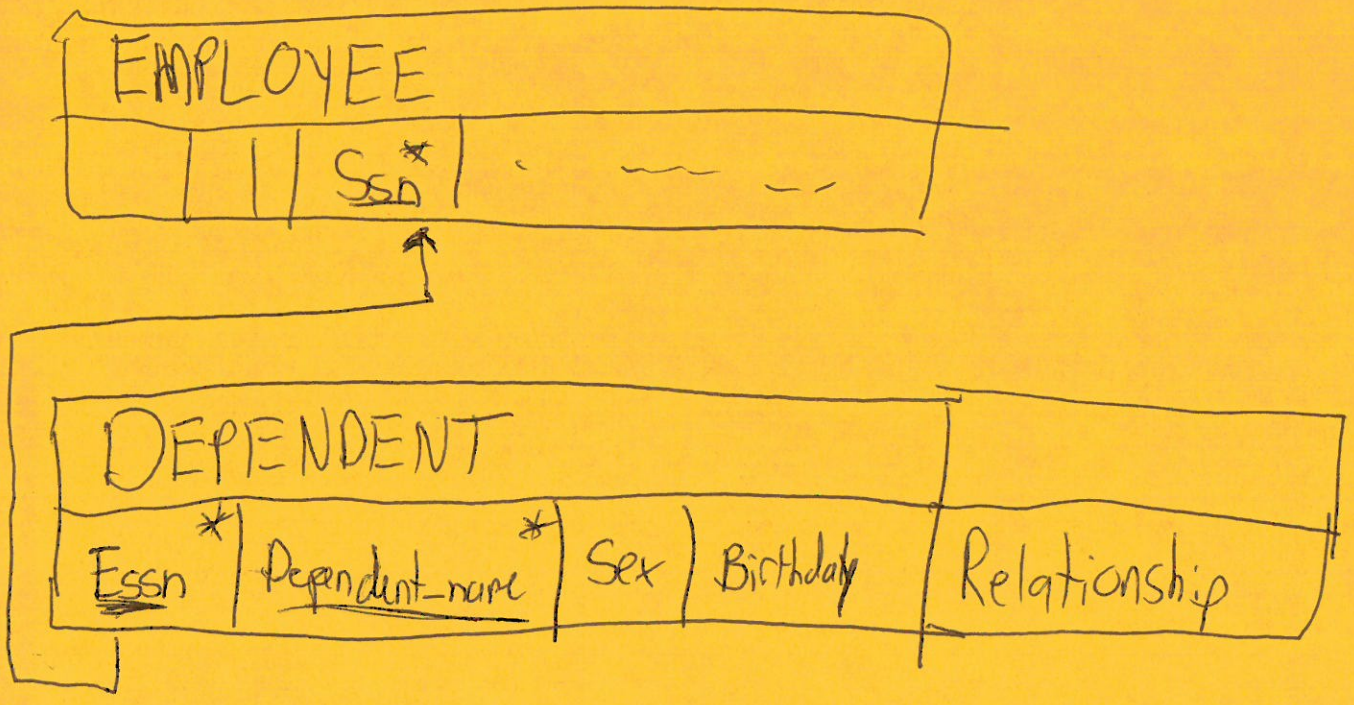
add foreign key attribute to R

from primary key attribute of W

Set primary key of ~~E~~ R as primary

key of owner and partial key of W


EMPLOYEE
Start_date
Number_of_employees
1
MANAGES
1
Hours
M
WORKS_ON
N
isor
1
SUPERVISION
Supervisee
N
1
Name
DEPENDENTS_OF
Number
N
DEPENDENT
Name
Sex
Birth_date
Relationship


EMPLOYEE
Ssn*
DEPENDENT
Essn*
Dependent_name*
Sex
Birthday
Relationship

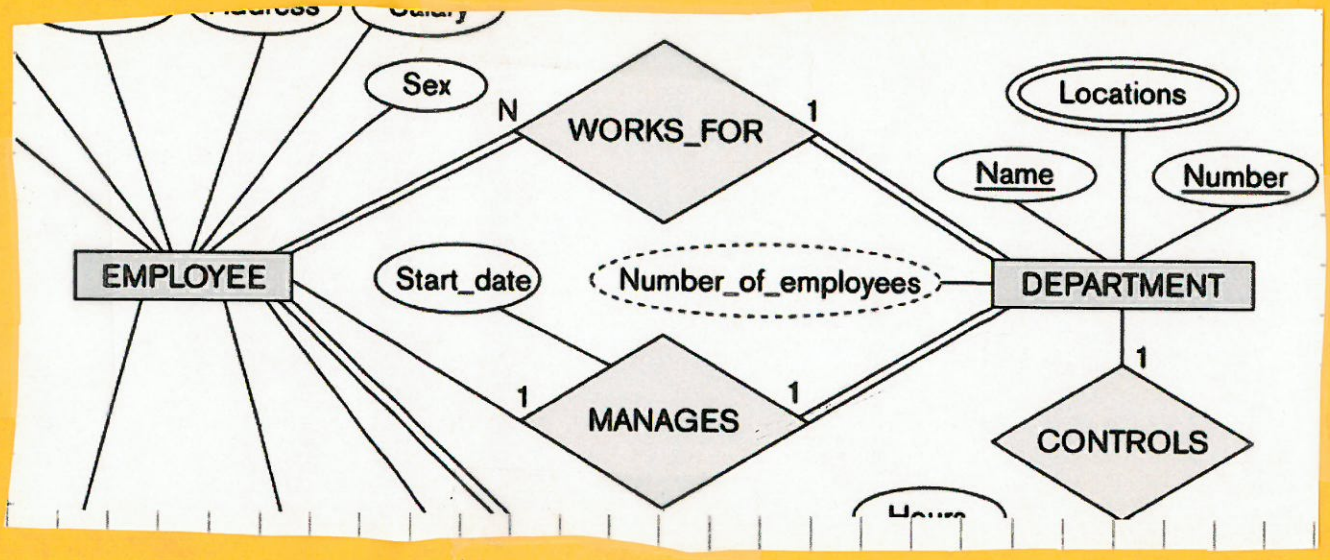
Step 3: Mapping 1:1 relationships

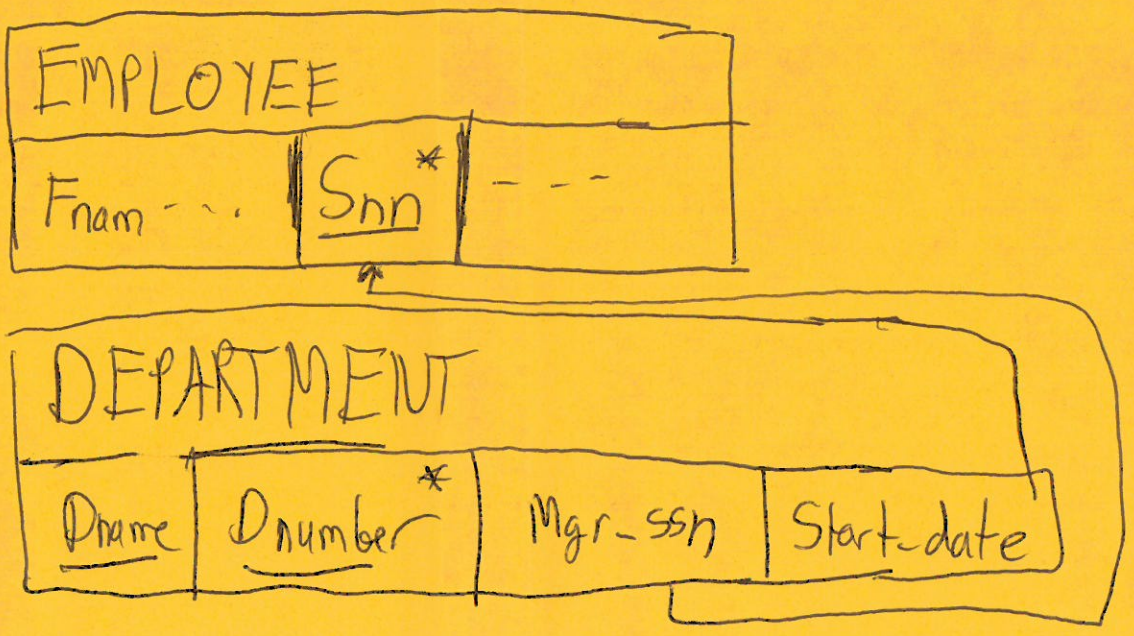
For each 1:1 relationship R
identify relations S and T for
entity types participating in R
pick approach and perform:

- foreign key (most common)
- merged relationship
- cross reference

Foreign Key

choose one as foreign key of other
(e.g. S will have a foreign key to T)
(often choose S w/ total participation)
add foreign key attribute





Merged relation (useful if both entity types have total participation)

- merge two relations into 1