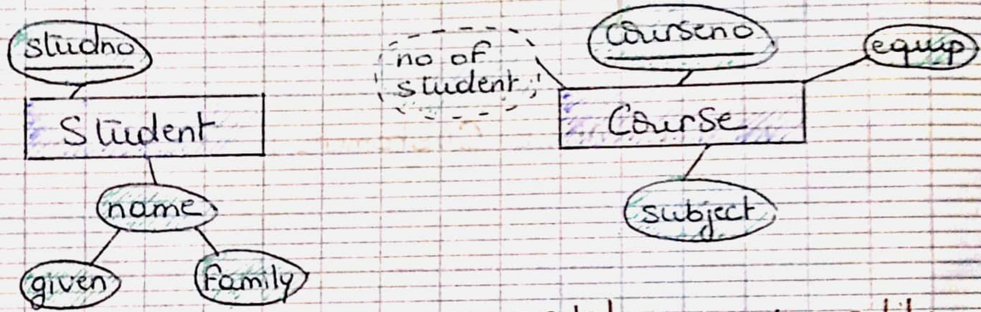
CH5: Mapping ER Model to Relational Model

→ Step 1: Mapping of Regular Entity Types

* A regular Entity become a relation (table)
* Attribute become the column header
  - → Simple Attribute:
    map directly into the relation
  - → Composite Attribute:
    map only their simple attribute
  - → Derived Attribute:
    not included in the relation



ال entity صارت relation

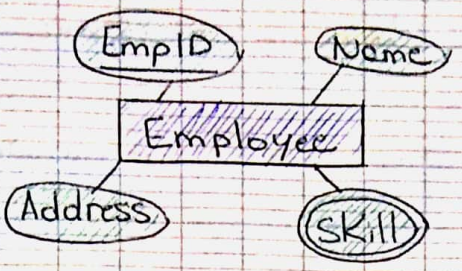
⇓ لغينا ال attr الأساسية بال composite attr

وحافظنا على ال simple attr

Student (studno , givenname, Familyname)

Course (courseno , equip , subject)

→ Multivalued Attribute
Become a separate attribute with a foreign key that is the primary key of the superior entity



Employee

EmpID	Employee Name	Employee Address

Employee_SKill

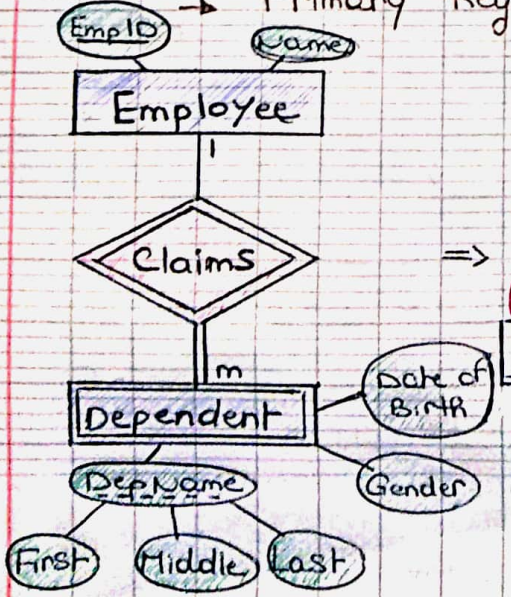
EmpID	SKill

* ال Multi.valued attribute (SKill) تحول إلى relation مستقلة تحوي على PK الذي هو foreign key في ال Entity الأساسية بالإضافة إلى key ثانية وهي ال attr كما هي تماما.

* FK (PK of the superior Ent)
+ PK (The multivalued attr)
= PK of the new relation

→ Step 2: Mapping of Weak Entity Types

* Weak Entity become a separate relation
* With a Primary Key composed of:
  - → Partial identifier of the weak Entity
  - → Primary Key of the strong entity



=>

Employee

Employee ID	Employeename

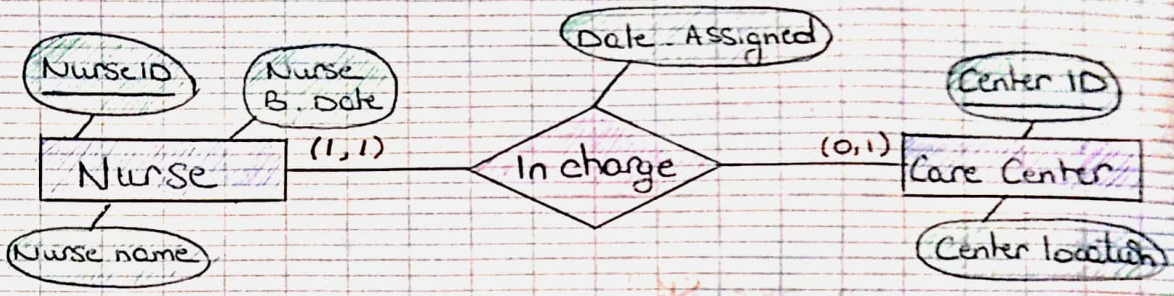
Dependent

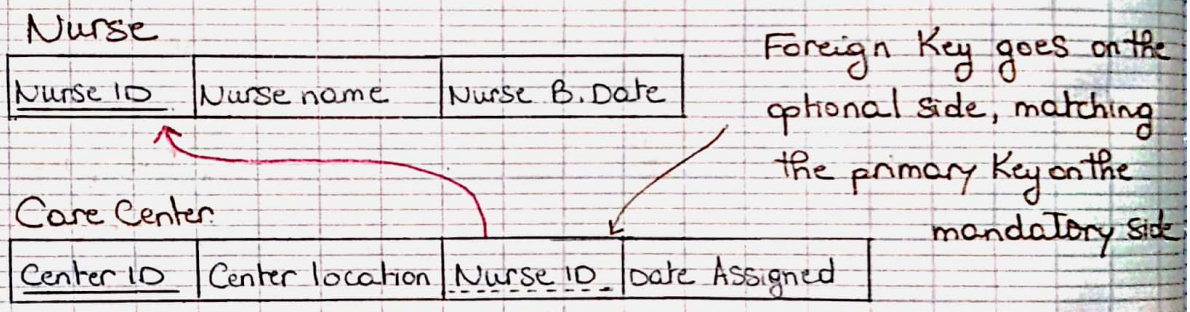
EmpID	F.name	M.name	L.name	Gender	Date of Birth

Note!! The domain constraint for the foreign key should not allow null value cause Dependent is a weak entity

→ Step 3: Mapping of Binary 1:1 Relationship Types

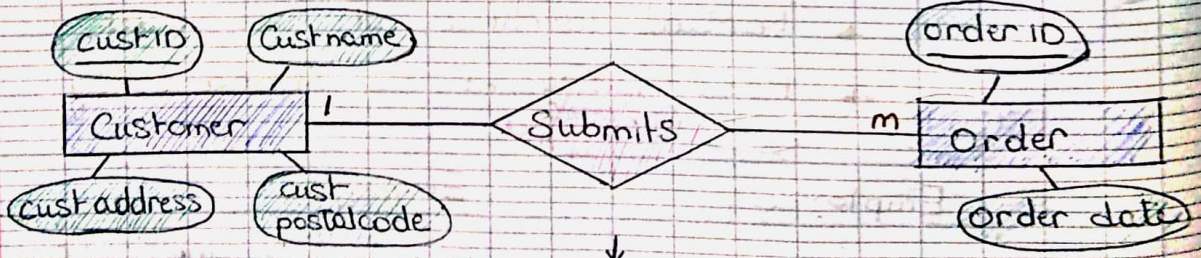
*Primary Key on the mandatory / (1,1) / ═ side become a Foreign Key on the optional / (0,1) / — side

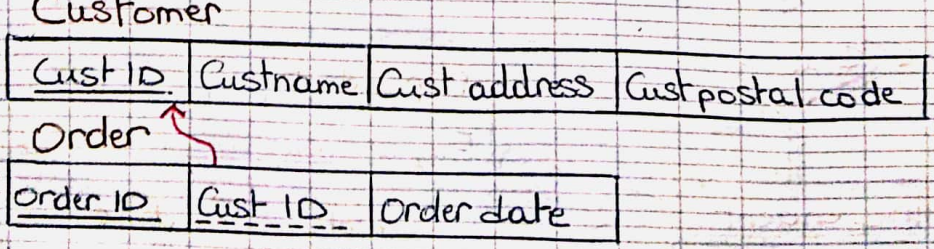




→ Step 4: Mapping of Binary 1:N Relationship Types

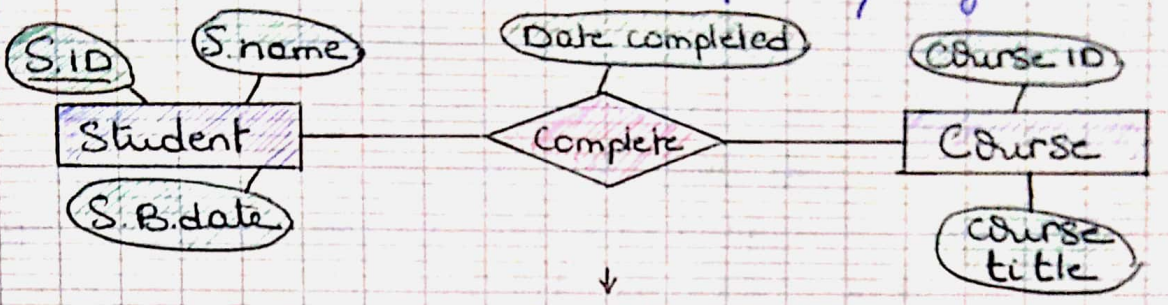
Primary Key on the one side become a foreign Key on the many side.





→ Step 5: Mapping of Binary M:N Relationship Types

Creat a new relation with the primary Keys of the two entities as its primary Key



Student

S.ID	S.name	S.B.date

Certificate

S.ID	Course ID	Date completed

Course

Course ID	Course title

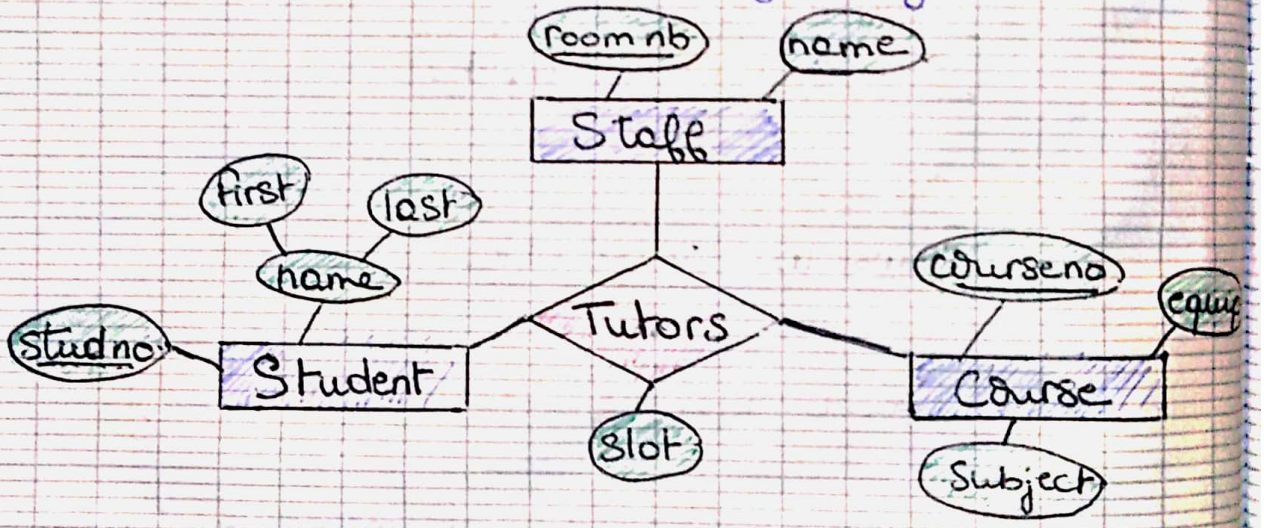
→ Step 6: Mapping Associative Entity

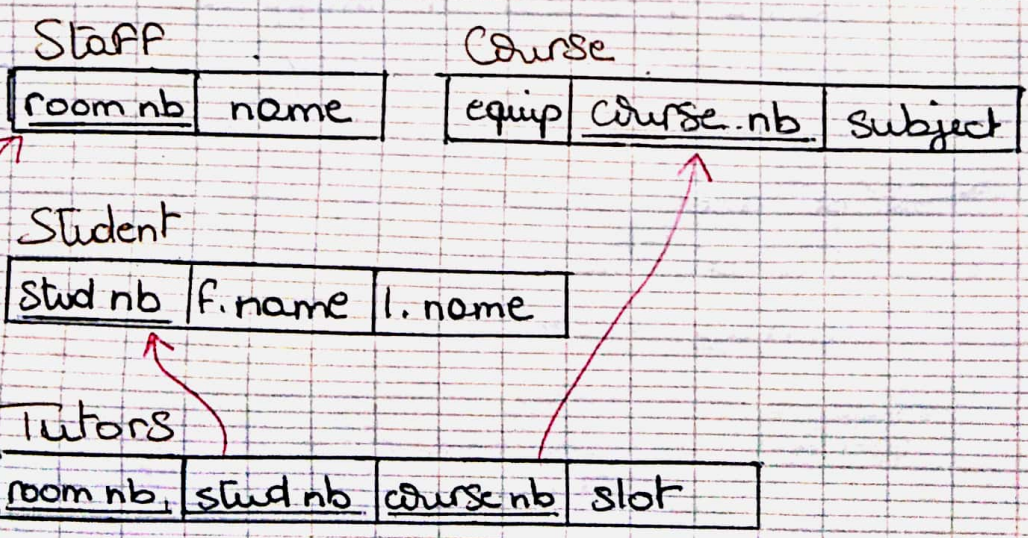
- Same process of mapping M:N relationship
- The associative entity become a separate relation
- The only difference is that the primary key of the relations may not be the primary key of the other 2 relations — it may differs from foreign keys

→ Step 7: Mapping of N-ary Relationship Types

↳ Mapping of N-ary relationship with $n > 2$:

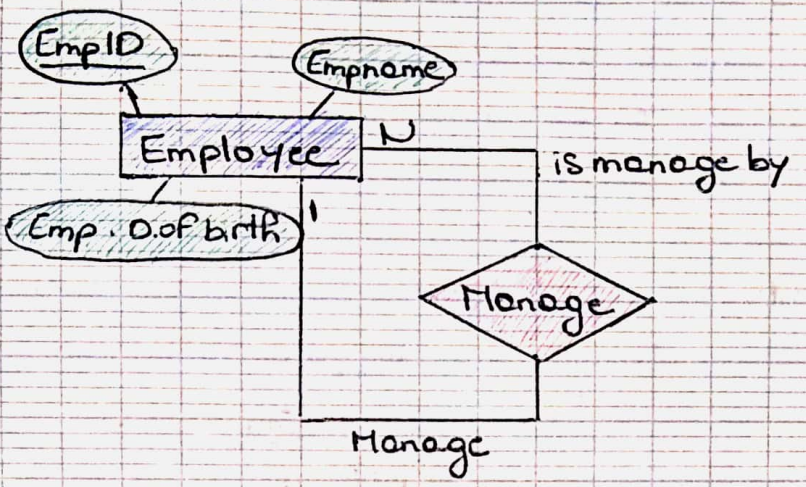
1. Create new relation that represent the relationship
2. Add the primary Keys of each entity participating in the relationship as ForeignKeys
3. All attribute of the relationship are included in the relation
4. The primary Key of the relation is composite of all forgein Keys





↳ Mapping Unary relationship Type

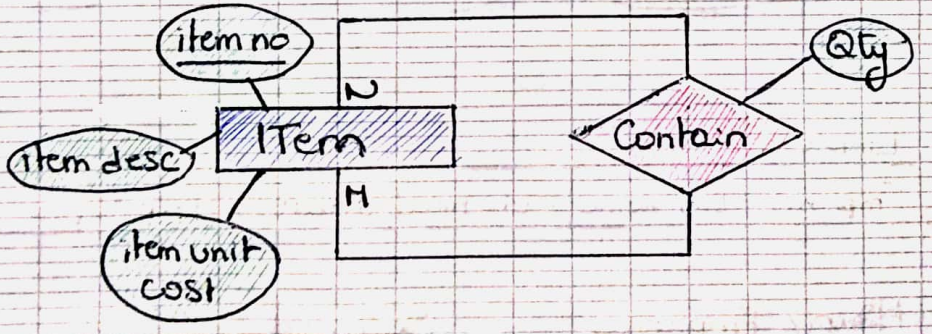
* (1 : N) : Recursive Foreign Key in the same relation



Employee

Emp ID	EmpName	Emp. D. of birth	Manger ID

* (N:M) : 2 relations are created
  - → one to present the entity
  - → one to present the relationship



Item

item no	item desc	item unit cost

Component

item no	Component no	Qty

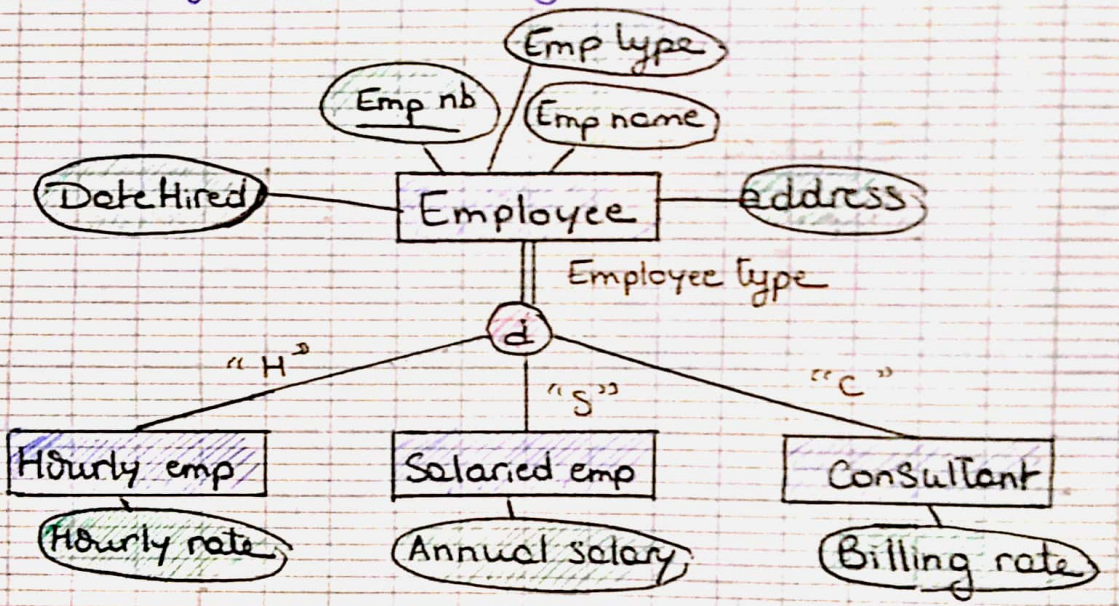
The primary Key of the associative relation has 2 attribute, both taken from the primary Key of the entity

→ Step 8: Mapping Specialization or generalization

↳ Option A: Multiple relations : superclass and subclass

- creat a relation for the superclass ant its attribute
- and a relation for each subclass includ their special attribute plus the primary Key of the superclass wich become its primary Key also

N.B !! OphonA work for any constraint on specialization disjoint, overlapping, total or partial



Employee

Emp nb	Emp name	Date hired	address	Emp type

Hourly Employee

HEmp nb	Hourly rate

Salaried Employee

SEmp nb	Annual salary

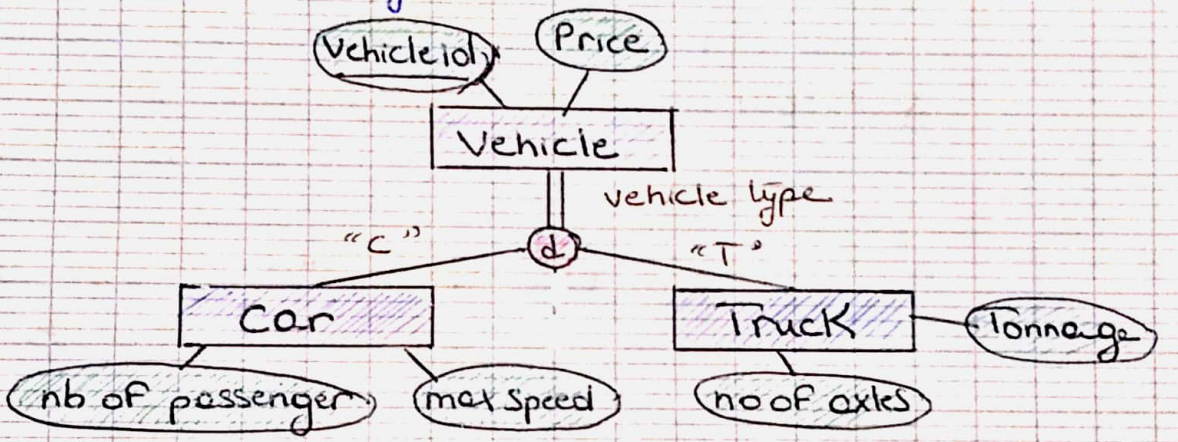
Consultant

CEmp nb	Billing rate

↳ Option B: Multiple relations : subclasses only

- Creates a relation for each subclass include the specific attributes and all the attributes of the superclass
- The primary key of the superclass is propagated to subclass relation and become its primary key

Note!! Option B work well only when both the disjoint and total constraint hold



Car

Cvehicle ID	Price	nb of passenger	max Speed

Truck

Tvehicle ID	Price	no of axles	Tonnage

↳ Options 8C and 8D

- Creat a single relation to present the superclass and all its subclasses

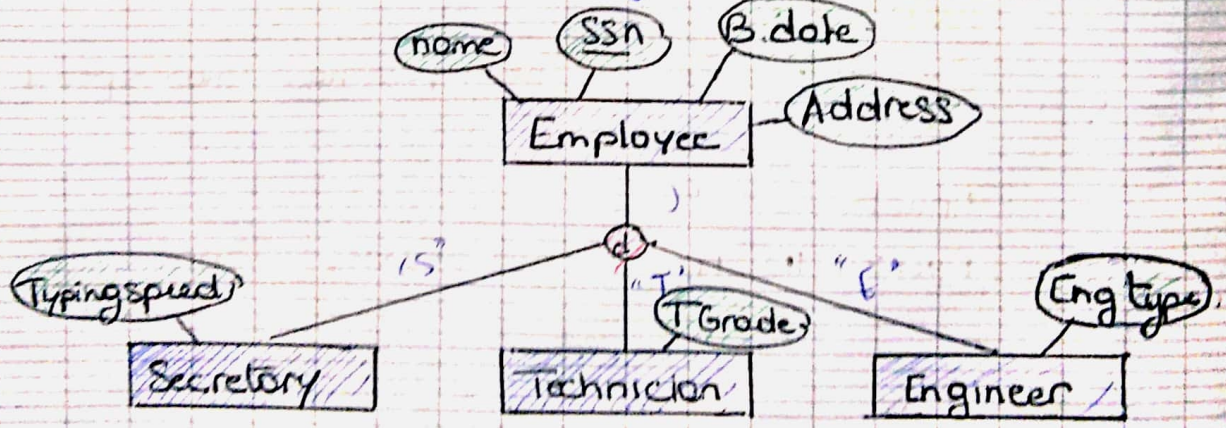
- An Entity that does not belong to some of the subclasses will have NULL values for the specific attribute of these subclasses

- These options are not recommended if many specific attributes are defined for the subclasses

هذه الخيارات غير محبّذة في حالة كانت الـ specific attr كثيرة لـ subclasses

أما إذا كان عدد الـ Specific attr قليل فهنه الخيارات أفضل من 8A and 8B.

1. option 8C: Single relation with one type attribute is used to handle disjoint subclasses



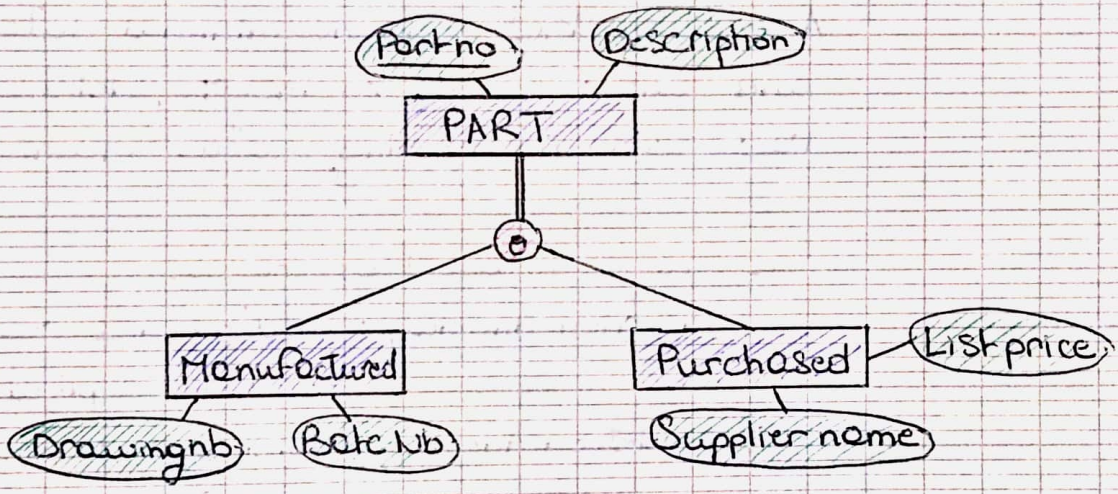
Employee

SSN	name	B.date	Address	Job type	Typing speed	T grade	Eng type

2. Option 8C : Single relation with multiple type attribute

- is used to handle overlapping subclasse by including Boolean type , one for each subclass

- Each type field can have a domain {yes, no} where a value yes indicates that the tuple is a member of subclass



Part

Partno	Desc	MFlag	Drawingnb	BatchNb	PFlag	Suppname	Listprice

Boolean type have a domain :

- yes if the part is a member in the subclass
- no if the part isn't a member in the subclass