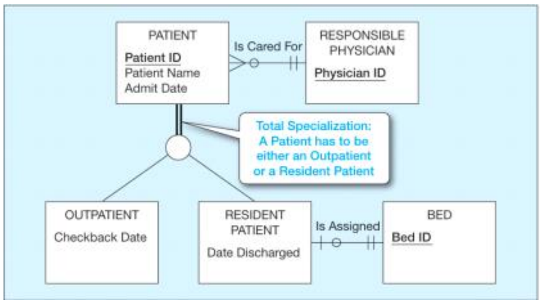
SPECIFYING CONSTRAINTS IN SUPERTYPE/SUBTYPE RELATIONSHIPS

"In this section, we introduce additional notation to represent constraints on supertype/subtype relationships. These constraints allow you to capture some of the important business rules that apply to these relationships. The two most important types of constraints that are described in this section are complete ness and disjointness constraints"

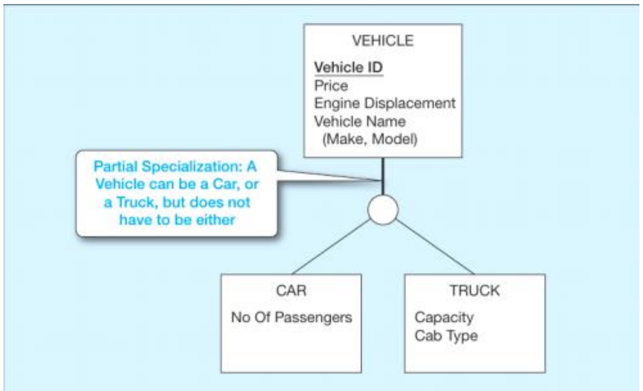
Specifying <u>Completeness Constraints</u>: addresses the question of whether an instance of a supertype must also be a member of at least one subtype.

- **<u>total specialization rule</u>**: specifies that each entity instance of the supertype must be a member of some sub type in the relationship.
    - ○ In an EER model this is indicated by double lines to the big circle



In this example, every time a new instance of PATIENT is inserted into the super type, a corresponding instance is inserted into either OUTPATIENT or RESIDENT PATIENT.

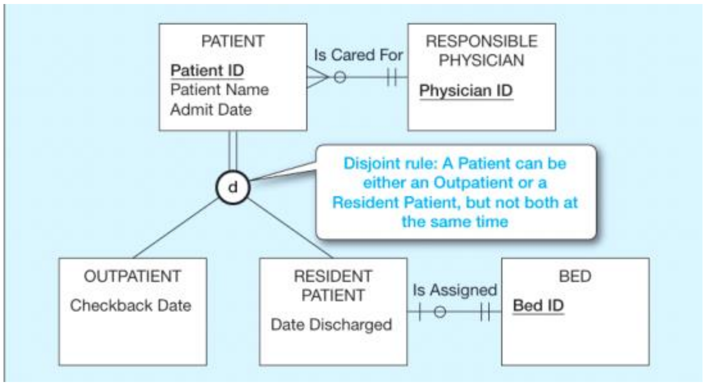
- **<u>partial specialization rule</u>**: specifies that an entity instance of the supertype is allowed not to belong to any subtype.
    - ○ In an EER model this is represented by a single line to the big circle



Recall that in this example, motorcycle is a type of vehicle, but it is not represented as a subtype in the data model. Thus, if a vehicle is a car, it must appear as an instance of CAR, and if it is a truck, it must appear as an instance of TRUCK. However, if the vehicle is a motorcycle, it cannot appear as an instance of any subtype because it has no attributes or relationships other than those for the supertype of VEHICLE. This is an example of partial specialization

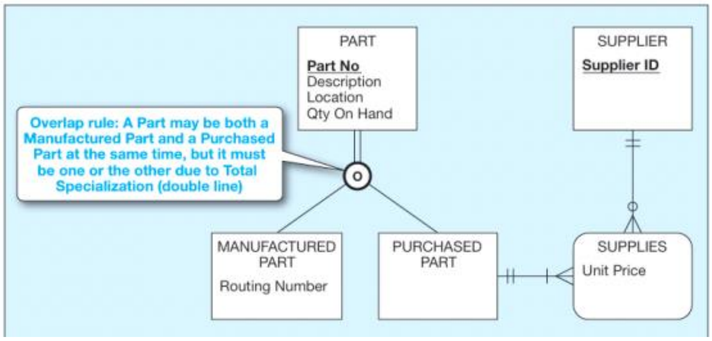
<u>Specifying Disjointness Constraints</u>: addresses whether an instance of a supertype may simultaneously be a member of two (or more) subtypes.

- **<u>Disjoint rule:</u>** specifies that if an entity instance (of the supertype) is a member of one subtype, it cannot simultaneously be a member of any other subtype.
    - ○ In EER model we use the d letter inside of the big circle to indicate disjoint rule



At any given time, a patient must be either an outpatient or a resident patient but cannot be both.

- **<u>Overlap rule:</u>** specifies that an entity instance can simultaneously be a member of two (or more) subtypes.
  - In EER model we use the letter o inside of the big circle to indicate the overlap rule

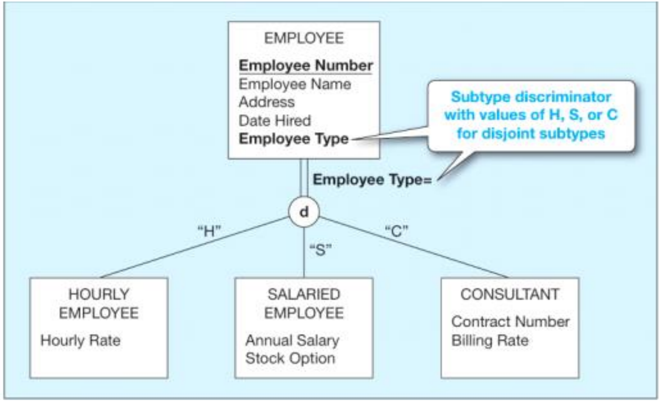

PART
Part No
Description
Location
Qty On Hand
SUPPLIER
Supplier ID
Overlap rule: A Part may be both a Manufactured Part and a Purchased Part at the same time, but it must be one or the other due to Total Specialization (double line)
o
MANUFACTURED PART
Routing Number
PURCHASED PART
SUPPLIES
Unit Price

Recall from our discussion of this example that some parts are both manufactured and purchased. Thus, any part must be either a purchased part or a manufactured part, or it may simultaneously be both of these.

Note: in this example we are talking about whole units and not individual units of a whole unit i.e say ford comes out with specialized wrenches and your warehouse has them from both the manufacturer and from home depot we would say it is both manufactored and a purchased part but we wouldn't look deeper into each part to say what individual wrench has been manu. Or purchased .

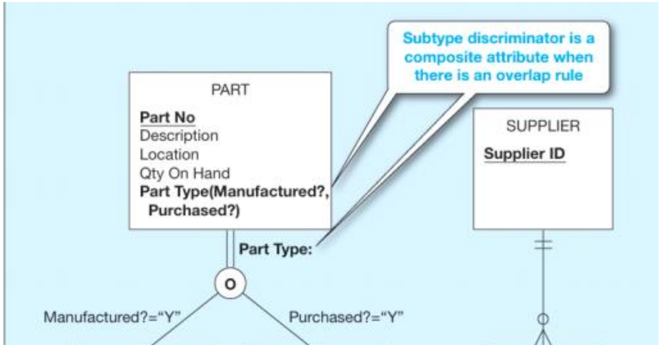
<u>Defining Subtype Discriminators</u>: is an attribute of a supertype whose values determine the target subtype or subtypes.

- This is essentially a key to tell you what the target subtype or subtypes are
- This plays along with the idea of the disjointness contraints
- **<u>DISJOINT SUBTYPES DESCRIMINATORS:</u>**

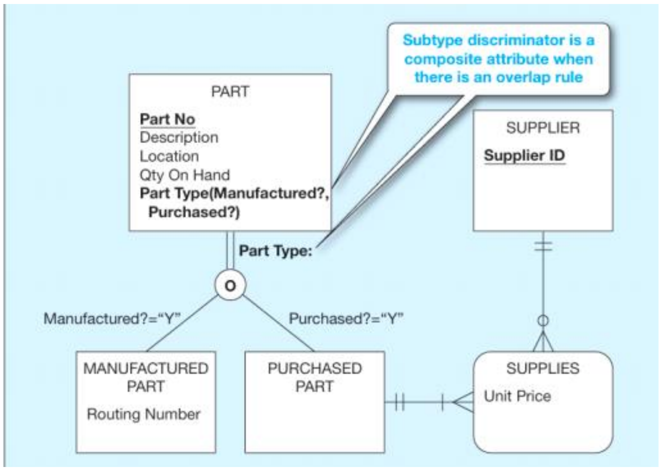


We now add an attribute inside of the supertype entity that specifies to us what subtype the entity being created falls under since it is total specialization and disjoint it must fall under <u>one</u> of the subtypes

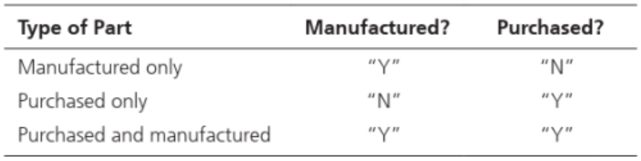
- **<u>OVERLAPPING SUBTYPES:</u>**


Subtype discriminator is a composite attribute when there is an overlap rule
PART
Part No
Description
Location
Qty On Hand
Part Type(Manufactured?, Purchased?)
SUPPLIER
Supplier ID
Part Type:
o
Manufactured?="Y"
Purchased?="Y"

We now add an attribute inside of the supertype which is a compound attribute that has essentially booleans that tell us what <u>subtypes</u> the part falls under since it is total specialization we know it falls under one of them and since it is overlap we know it could be more than one so we have a compound attribute that tells us which is which



We now add an attribute inside of the supertype which is a compound attribute that has essentially booleans that tell us what subtypes the part falls under since it is total specialization we know it falls under one of them and since it is overlap we know it could be more than one so we have a compound attribute that tells us which is which



Type of Part	Manufactured?	Purchased?
Manufactured only	"Y"	"N"
Purchased only	"N"	"Y"
Purchased and manufactured	"Y"	"Y"

Defining Supertype/Subtype Hierarchies