DATABASE - 1

3RD CLASS

COMPUTER SCIENCE DEPARTMENT

3rd Lecture – The Enhanced E-R Model and Business Rules

Sunday 6th of October 2024

LECTURER :

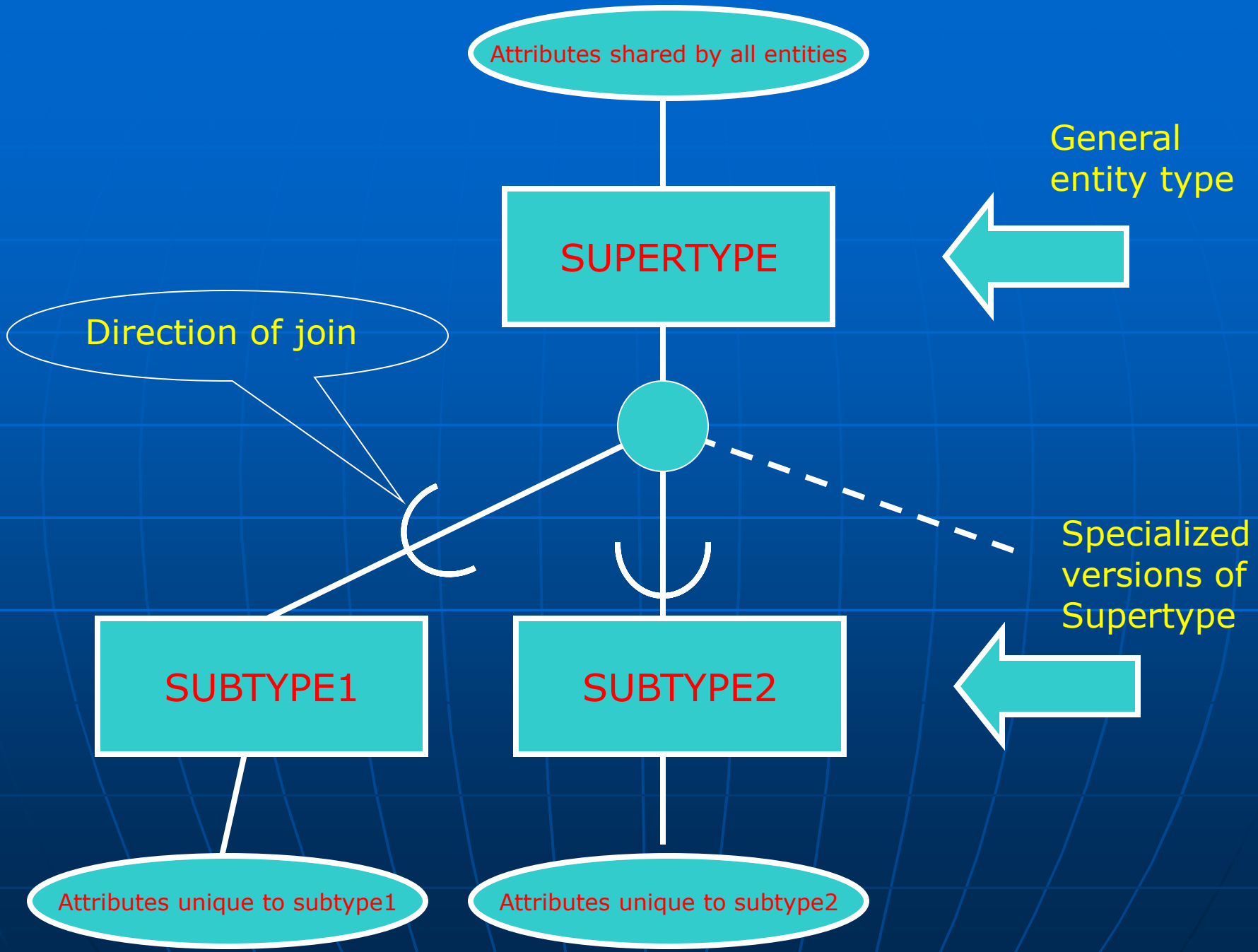
DR. RAYAN YOUSIF ALKHAYAT

*Enhanced Entity – Relationship model (EER):

- Is used to identify the model that has resulted from extending the original E-R Model with new modeling constraints .
- The most important new modeling constraint incorporated in the EER Model is Supertype / Subtype relationship .
- This facility allows us to model a general entity type called the Supertype and then subdivide it into several specialized entity type called subtype.

Representing Supertype and Subtypes:

- **Subtype:** a subgrouping of the entities in an entity type that is meaningful to the organization and that share common attributes or relationship distinct from other subgroupings.
- **Supertype**: is a generic entity type that has a relationship with one more subtypes

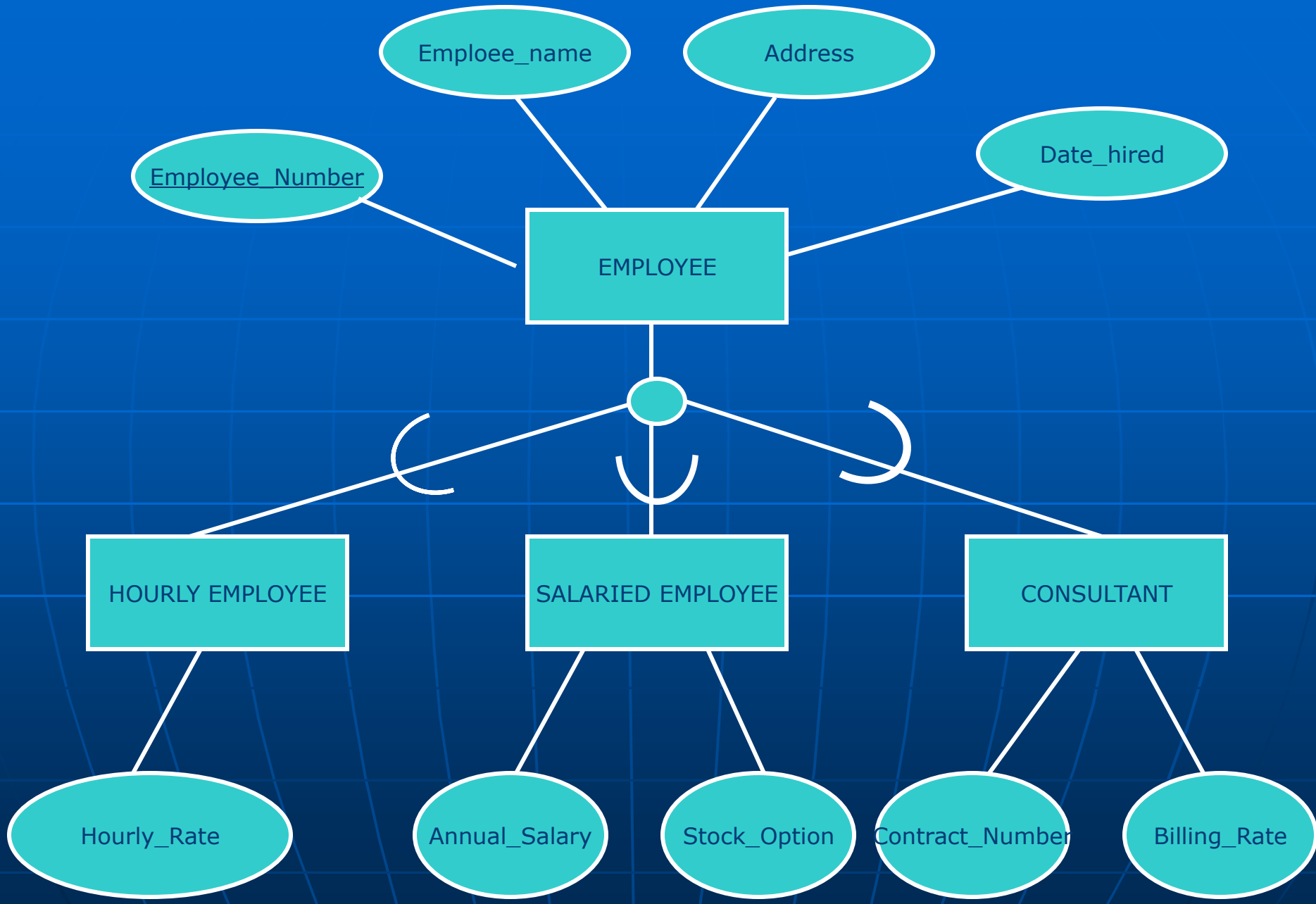

Attributes shared by all entities
General entity type
SUPERTYPE
Direction of join
Specialized versions of Supertype
SUBTYPE1
SUBTYPE2
Attributes unique to subtype1
Attributes unique to subtype2

In order to develop the Data Model Concept for the EMPLOYEE example , where we have three different kinds of employees , we have three choices :

1. Define a single entity type called EMPLOYEE , this approach has the disadvantage that EMPLOYEE would have to contain all of the attributes for the three types of employees . for an instance of an hourly – employee attributes such as annual - salary and contract - no would be null or not used when taken to development environment , programs that use this entity type would necessarily be quiet complex to deal with the many variations.

2. Define a separate entity type for each of the three entities , this approach would fail to exploit the common properties of employees , and users would have to be careful to delect the correct entity when using the system .

3. Define a Supertype called EMPLOYEE with Subtypes for : HOURLY- EMPLOYEE, SALARIED – EMPLOYEE and CONSULTANT , this approach exploit the common properties of all employees , yet recognized the distinct properties of each type .


Emploee_name
Address
Employee_Number
Date_hired
EMPLOYEE
HOURLY EMPLOYEE
SALARIED EMPLOYEE
CONSULTANT
Hourly_Rate
Annual_Salary
Stock_Option
Contract_Number
Billing_Rate

Attributes Inheritance

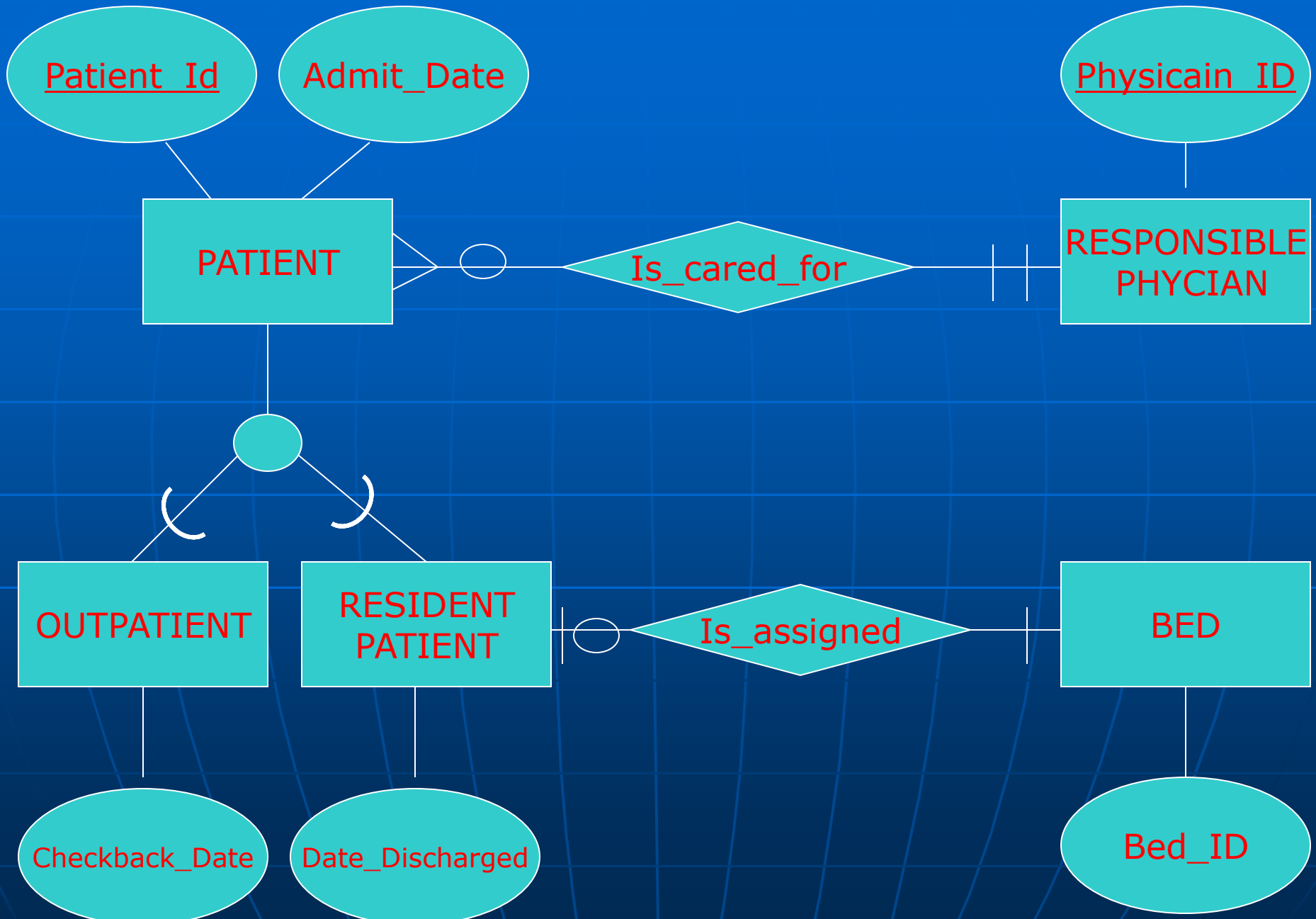
- A property that subtype entity inherit values of all attributes of the Supertype .
- Any attribute in the Supertype does not need to re-declare it in the subtype , its value came from the Supertype .

Q: when to use the Supertype / Subtype Relationship ?

You should consider using subtypes when either (or both) of the following conditions are present :

1. There are attributes that apply to some (but not all) of the instances of any entity type . ex : EMPLOYEE entity type
2. The instances of subtype participate in a relationship unique to that subtype .

eg : the hospital entity type PATIENT


Patient_Id
Admit_Date
Physicain_ID
PATIENT
Is_cared_for
RESPONSIBLE PHYCIAN
OUTPATIENT
RESIDENT PATIENT
Is_assigned
BED
Checkback_Date
Date_Discharged
Bed_ID

Representing Specialization and Generalization

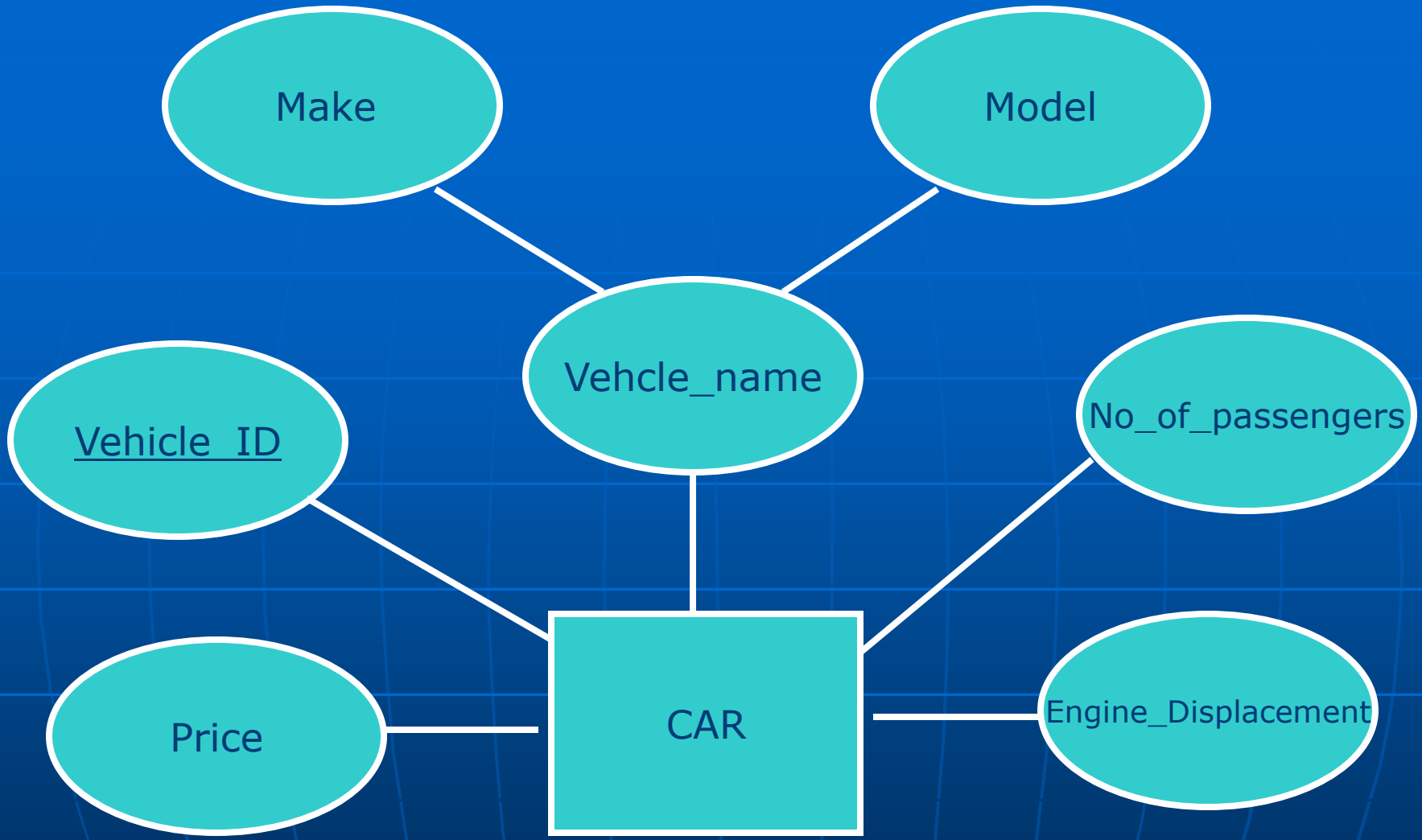
There are two processes that serves as mental models in developing Supertype/ Subtype relationships.

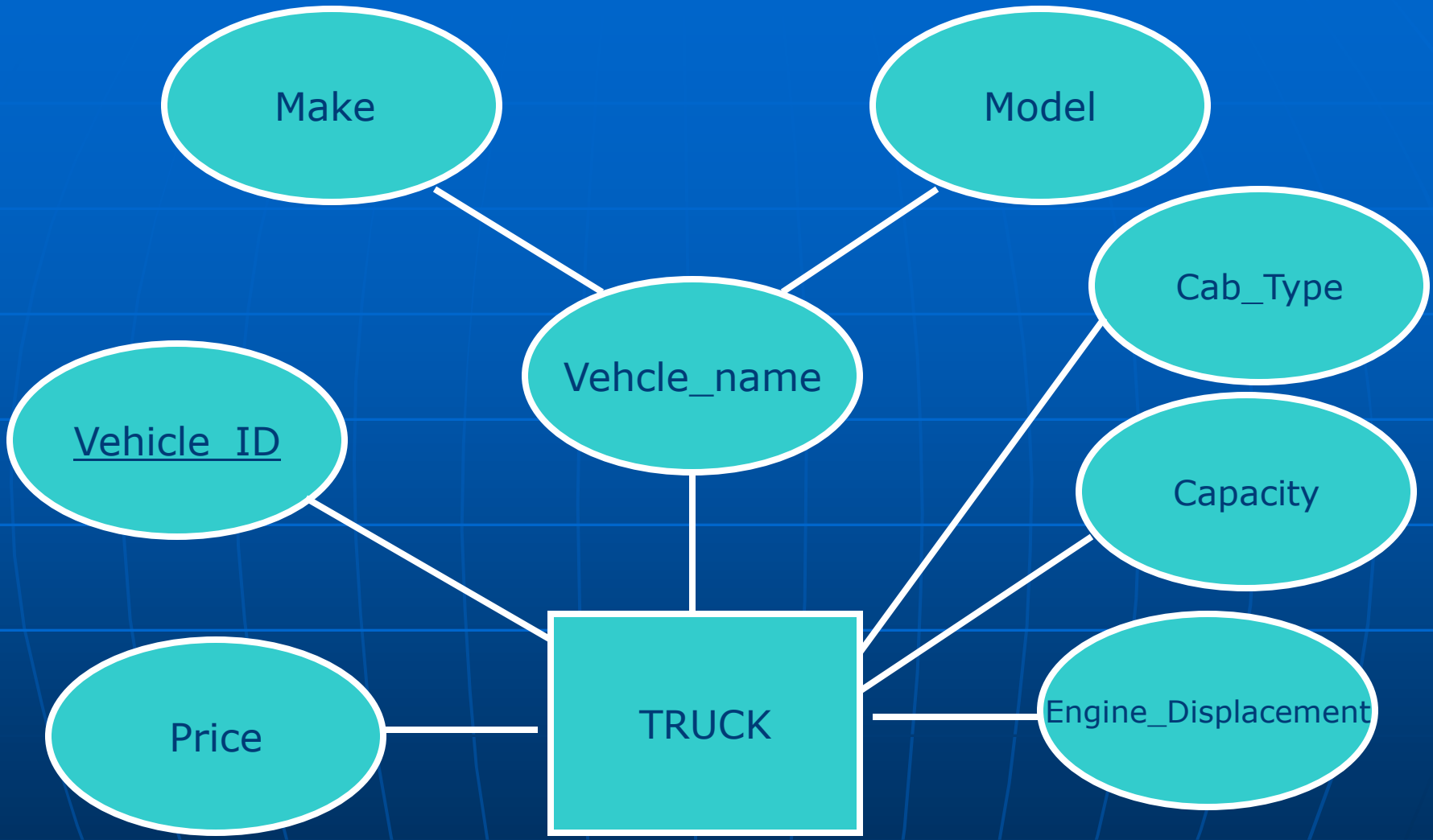
1. **Generalization**: it is a bottom – up process of defining a more general entity type from a set of more specialized entity type.

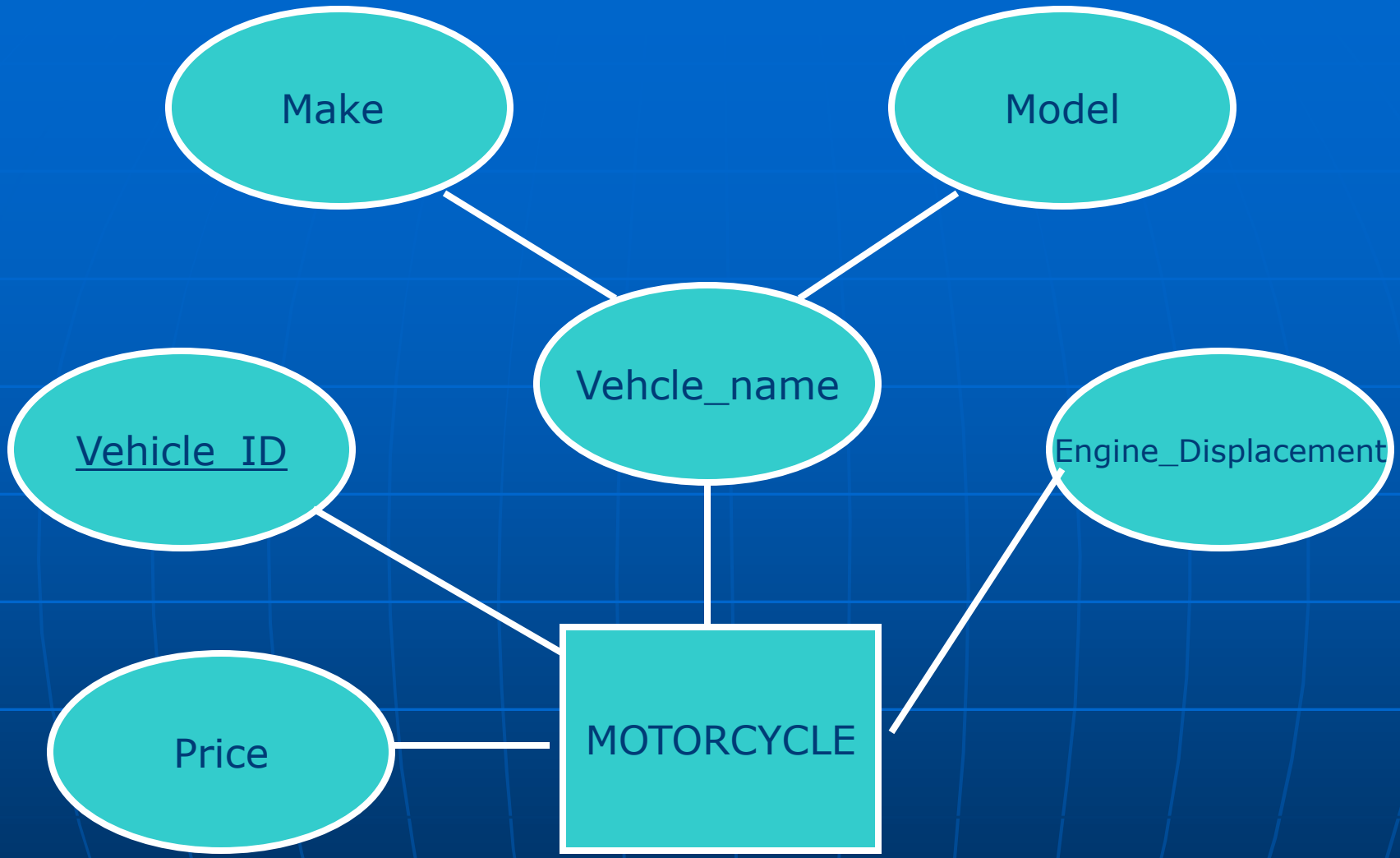
Examples for it: CAR, TRUCK, MOTORCYCLE.

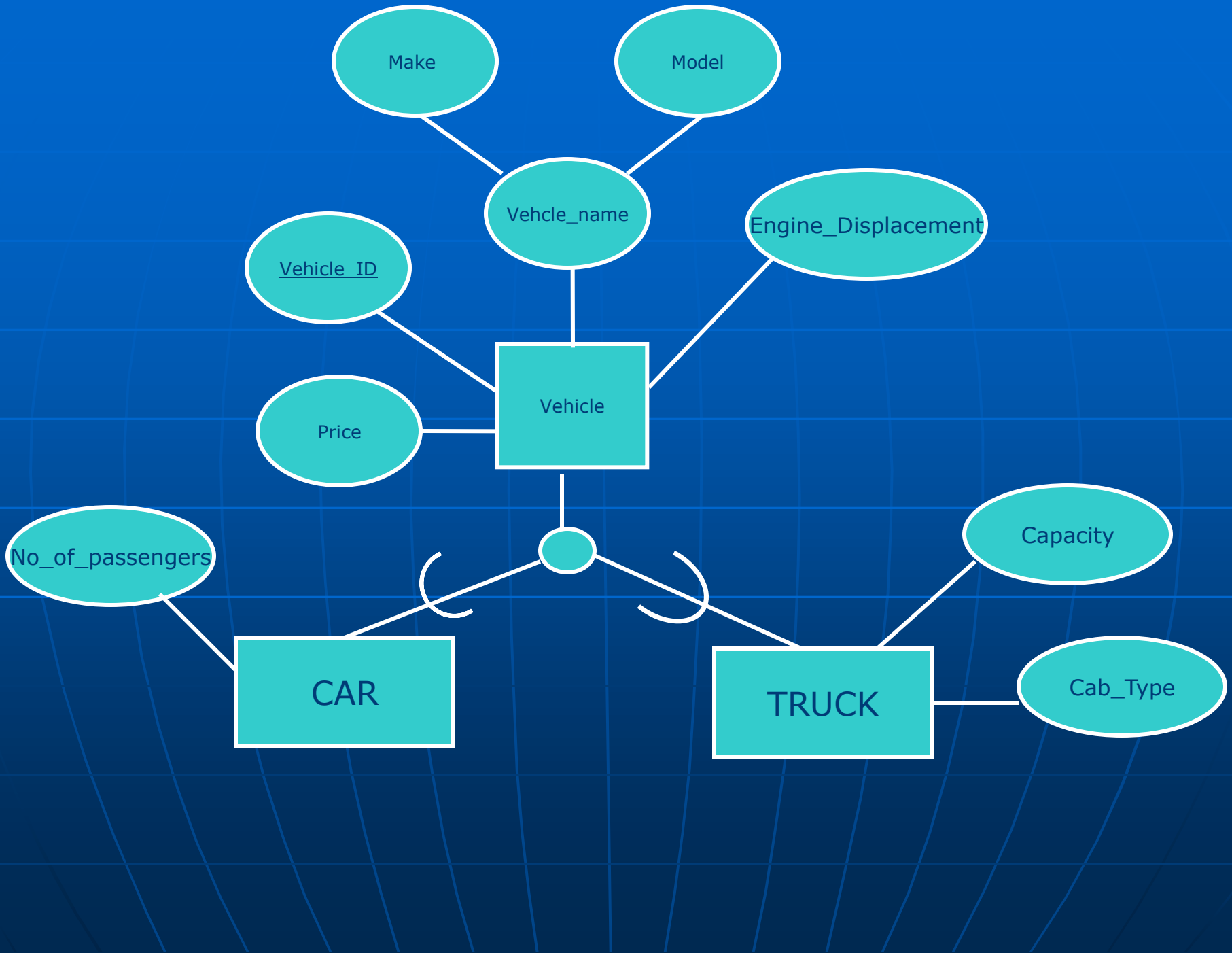
2. **Specialization:** it is atop – down process f defining one or more subtype relationship.

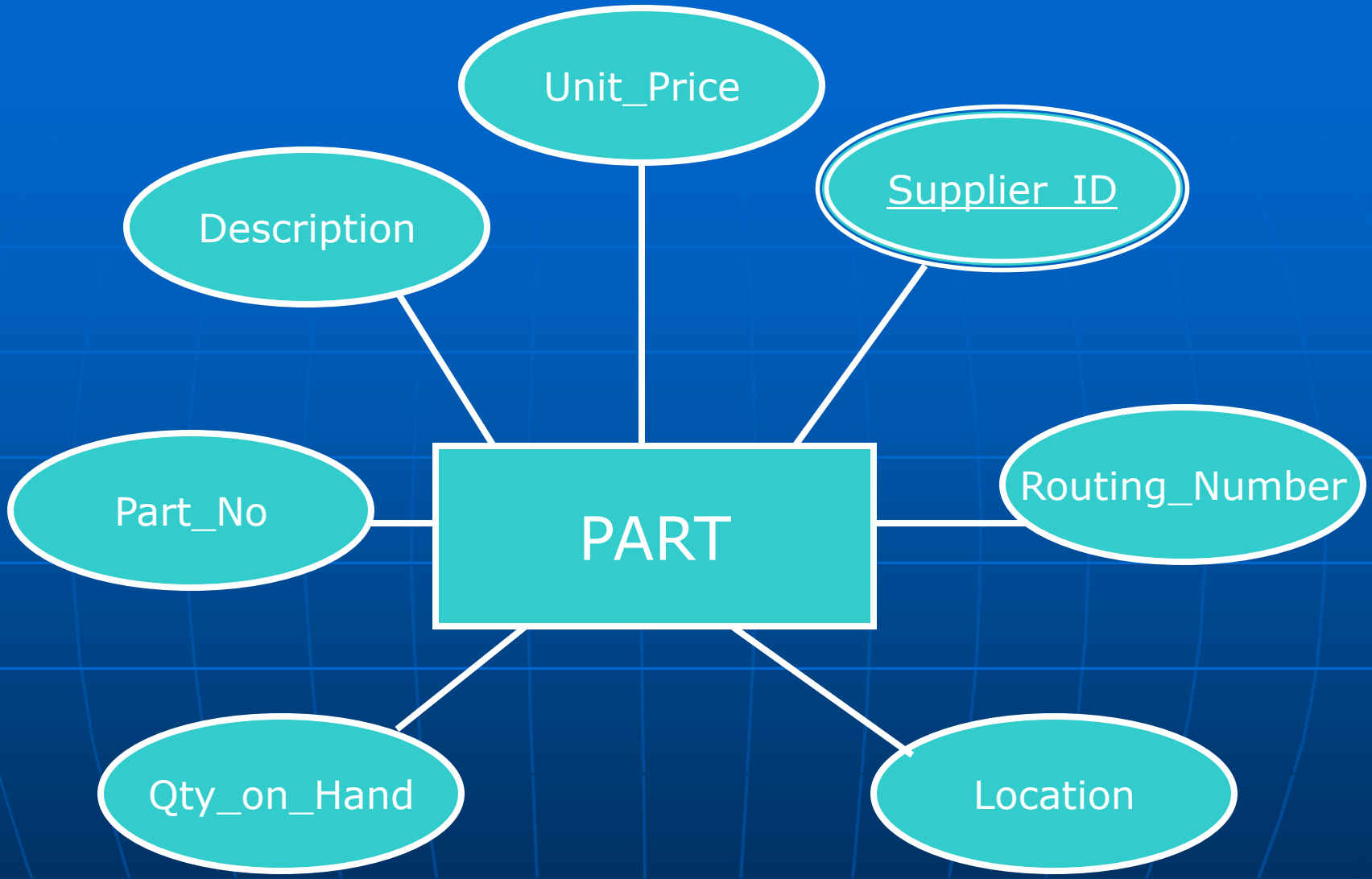
Examples for it: PART

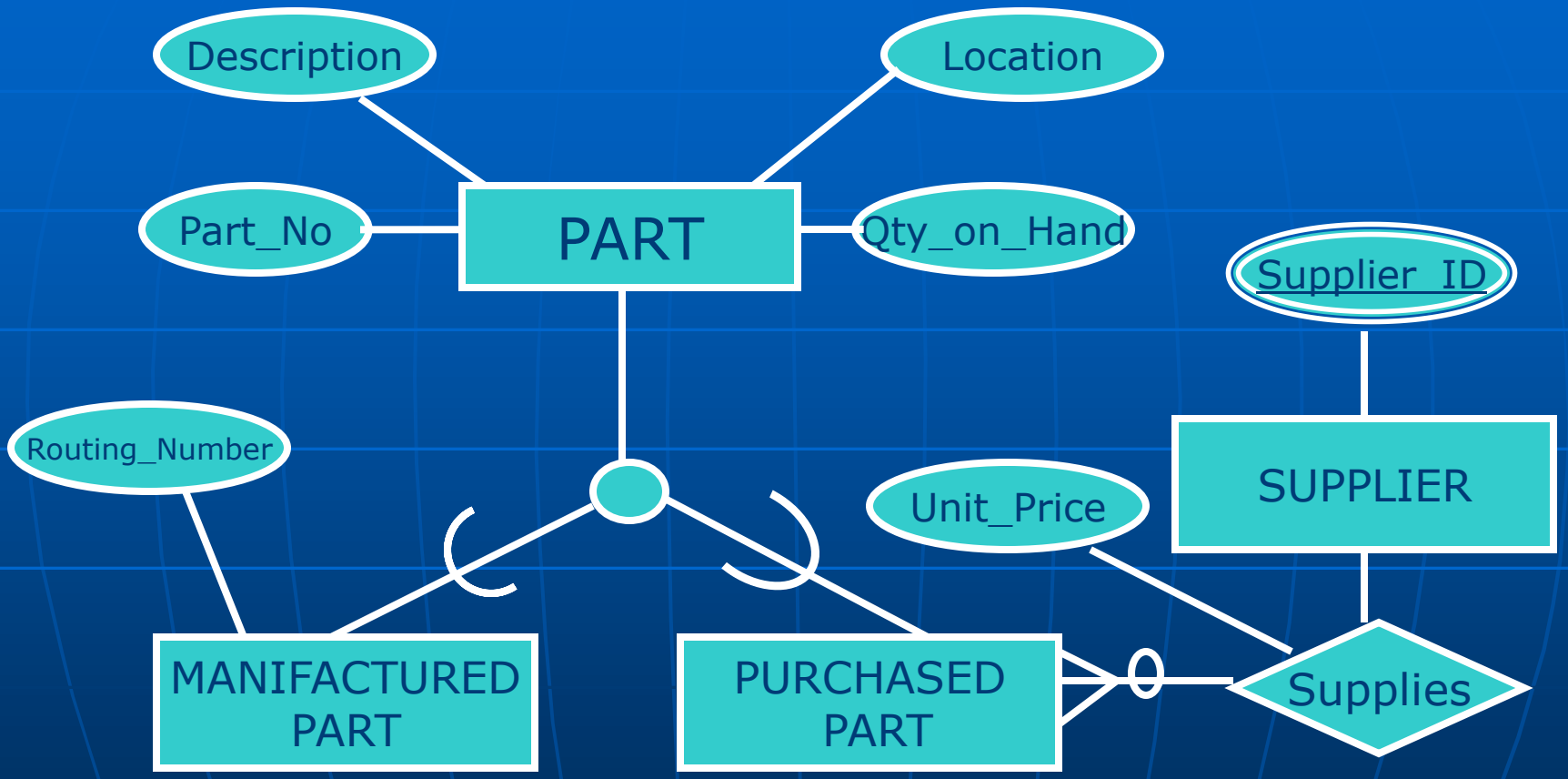

Make
Model
Vehcle_name
Vehicle_ID
No_of_passengers
Price
CAR
Engine_Displacement


Make
Model
Cab_Type
Vehcle_name
Vehicle_ID
Capacity
TRUCK
Price
Engine_Displacement


Make
Model
Vehcle_name
Vehicle_ID
Engine_Displacement
Price
MOTORCYCLE


Make
Model
Vehcle_name
Engine_Displacement
Vehicle_ID
Vehicle
Price
No_of_passengers
Capacity
CAR
TRUCK
Cab_Type


Unit_Price
Description
Supplier_ID
Part_No
PART
Routing_Number
Qty_on_Hand
Location


Description
Location
Part_No
PART
Qty_on_Hand
Supplier_ID
Routing_Number
SUPPLIER
Unit_Price
MANIFACTURED PART
PURCHASED PART
Supplies

Specifying Constraints in Supertype/Subtype Relationships

We introduce notation to represent constraints on Supertype / subtype relationship . These constraints allow us to capture some of the important business rules that apply to these relationships.

The two most important types of constraints are :

1. Completeness Constraints
2. Disjointness Constraints

1. Specifying Completeness Constraints

Completeness Constraints : a type of constraints that address the question wheather an instance of Supertype must be also a member of at least one subtype .

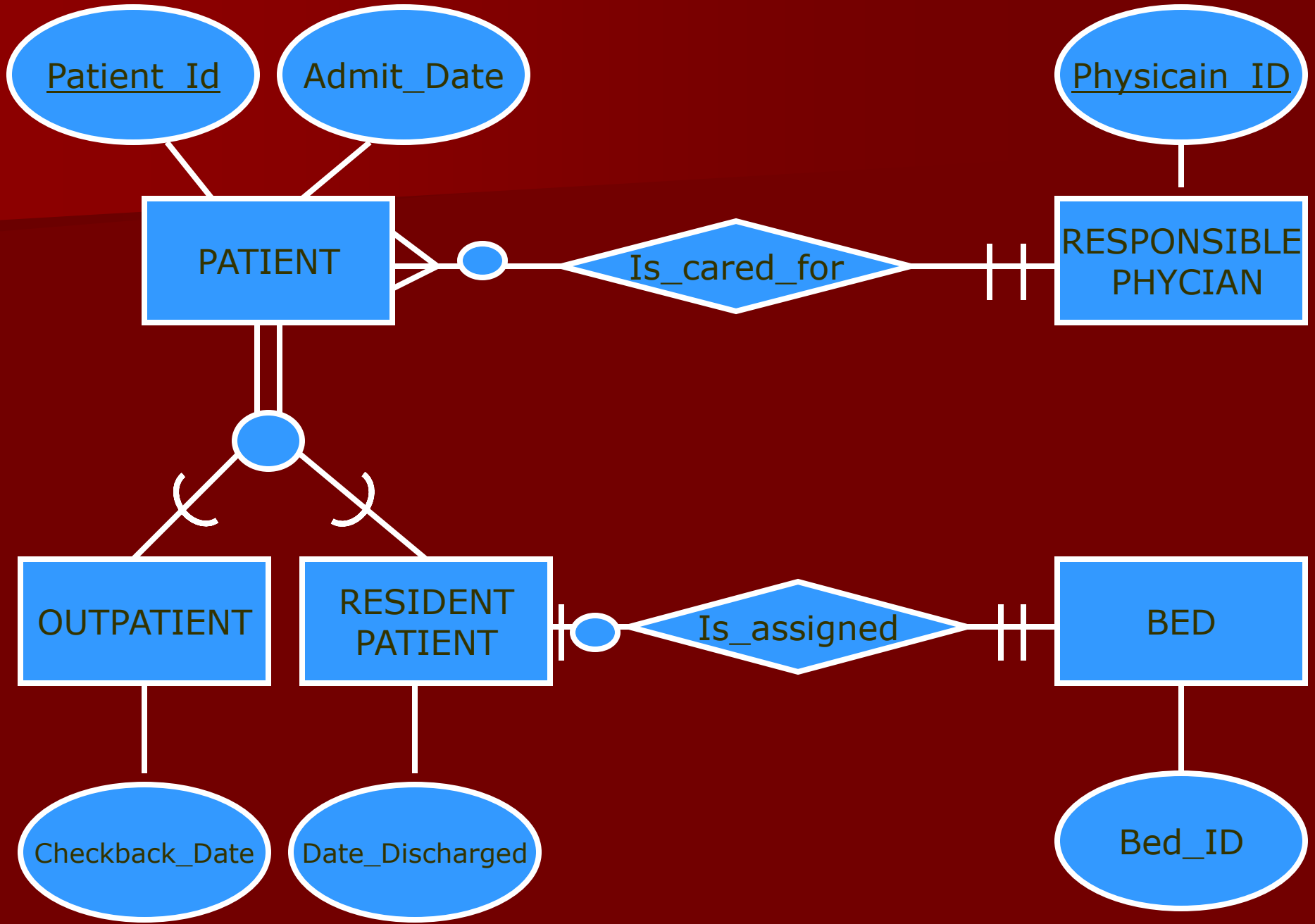
*It has two possible rules :

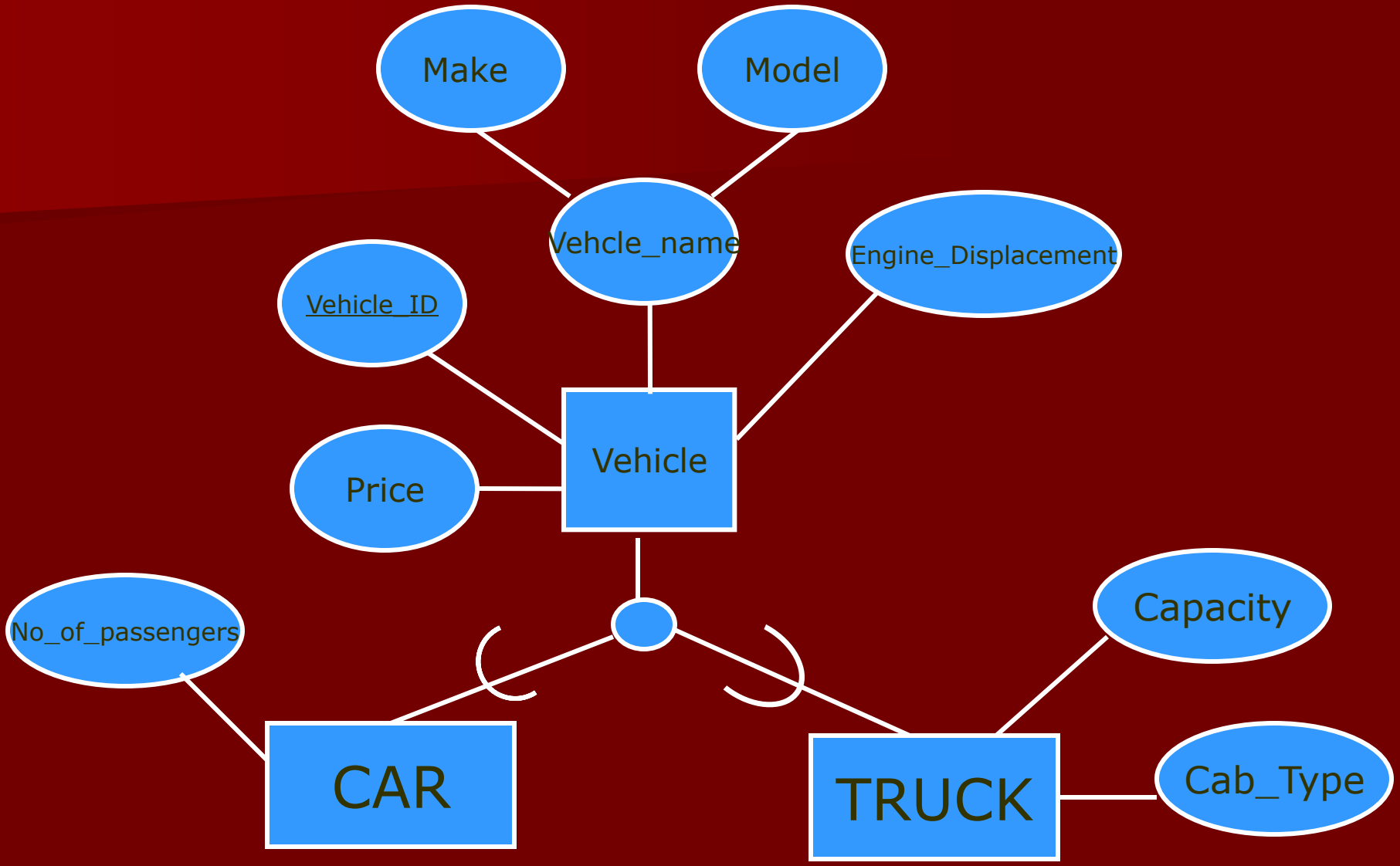
A. Total specialization rule: specifies that each entity instance of super type must be member of some subtype in the relationship, Ex: EMPLOYEE,

- *Represented by Double line.*

B. Partial specialization rule: specifies that an entity instance of super type is allowed not to belong to any subtype. Ex: VEHICLE

- *Represented by Single line.*


Patient_Id
Admit_Date
Physicain_ID
PATIENT
Is_cared_for
RESPONSIBLE PHYCIAN
OUTPATIENT
RESIDENT PATIENT
Is_assigned
BED
Checkback_Date
Date_Discharged
Bed_ID


Make
Model
Vehcle_name
Engine_Displacement
Vehicle_ID
Vehicle
Price
No_of_passengers
Capacity
CAR
TRUCK
Cab_Type

2.Specifying Disjontness Constraints

Disjointness Constraints : a constraint that addresses the question whether an instance of Supertype may simultaneously be member of two (or more)subtypes . It has two possible rules :

A. the disjoint rule : specifies that if an entity instance (of the same Supertype) ,is a member of one subtype , it ***cannot simultaneously*** be a member of any other subtype.

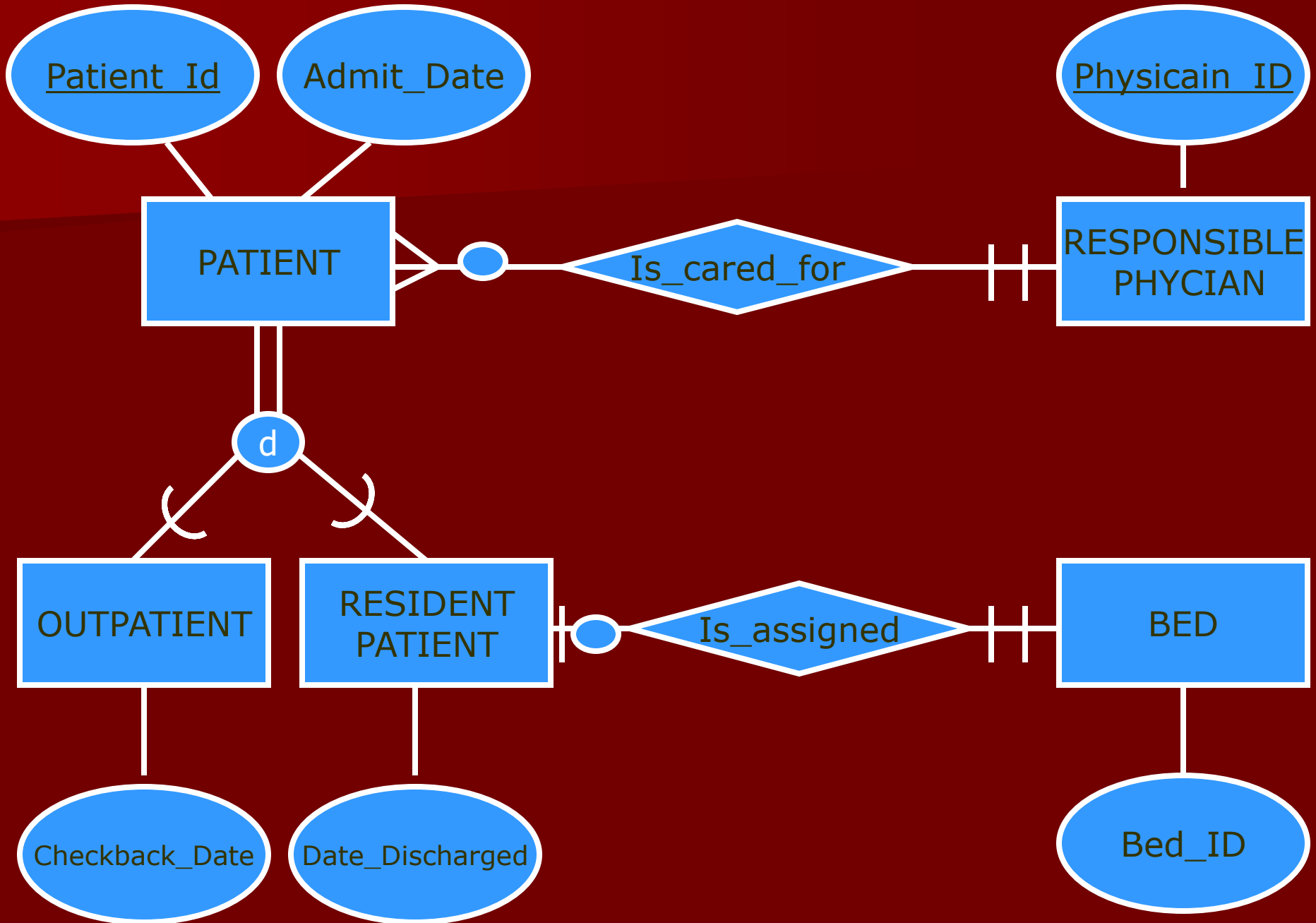
Ex: PATIENT

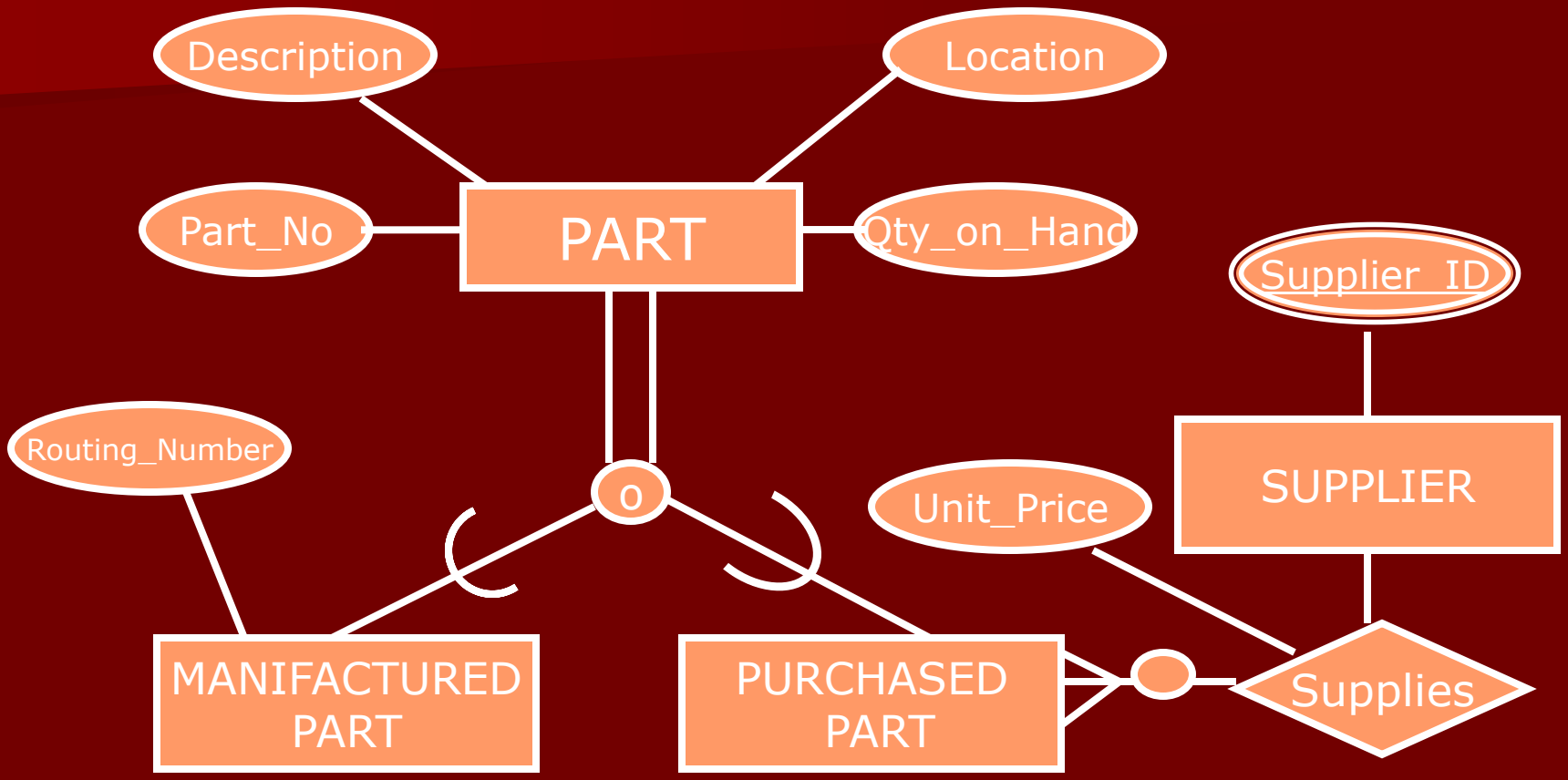
- *Represented by letter (d) inside the circle that joint Supertype with subtype.*

B. The overlap rule: specifies that any entity instance ***can simultaneously*** **be a member of two or more subtypes.**

Ex: PART .

- *Represented by letter (o) inside the circle that joint Supertype with subtype.*


Patient_Id
Admit_Date
Physicain_ID
PATIENT
Is_cared_for
RESPONSIBLE PHYCIAN
d
OUTPATIENT
RESIDENT PATIENT
Is_assigned
BED
Checkback_Date
Date_Discharged
Bed_ID


Description
Location
Part_No
PART
Qty_on_Hand
Supplier_ID
Routing_Number
O
SUPPLIER
Unit_Price
MANIFACTURED PART
PURCHASED PART
Supplies

Defining Subtype Discriminators

A problem occurred when new instance added to the Supertype , so we'll discuss some rules , one of them are the subtype discriminators .

Subtype Discriminators: is an attributes of Supertype whose values determine the target subtype or subtypes. It has two types :

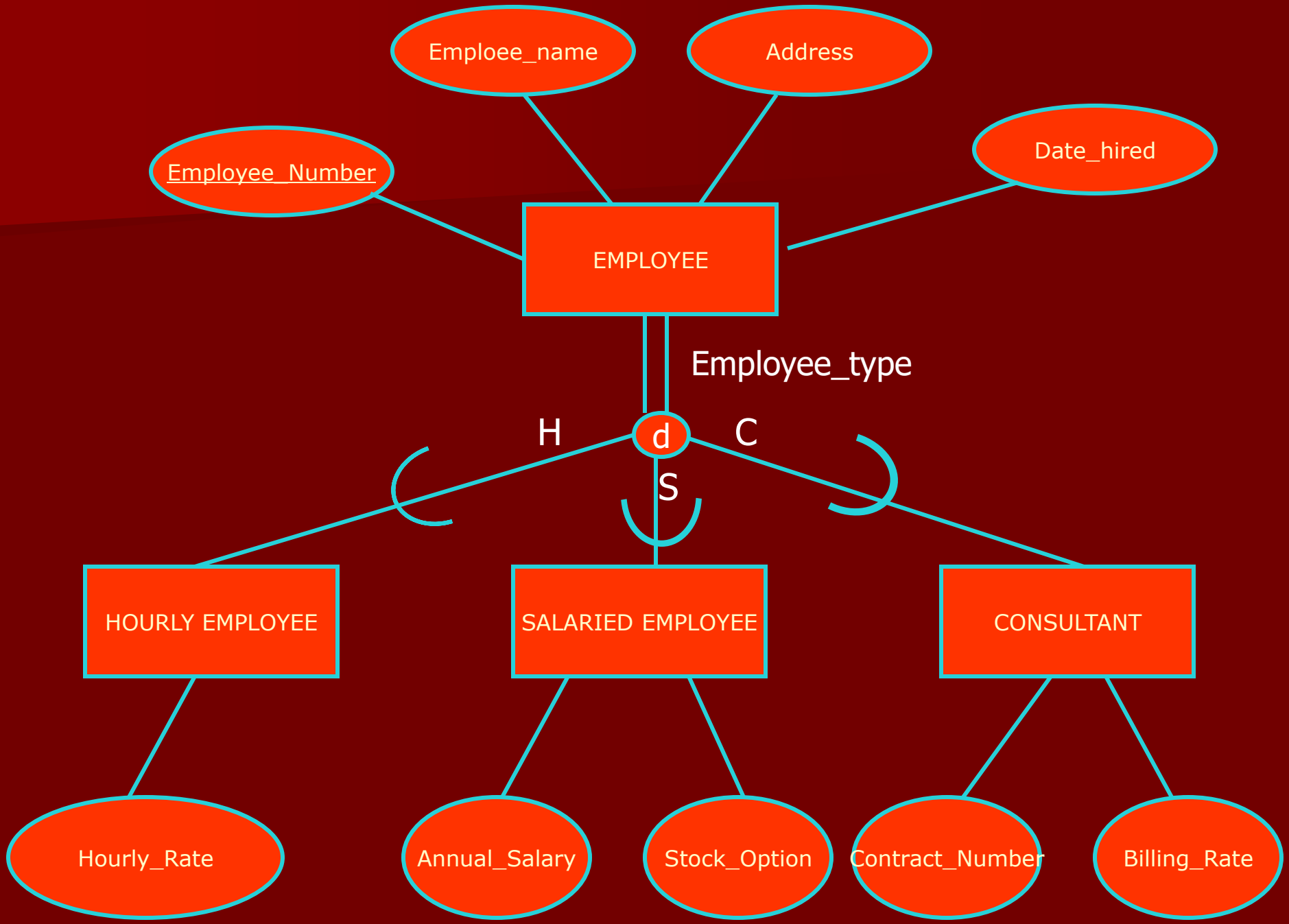
1. Disjoint Subtypes:

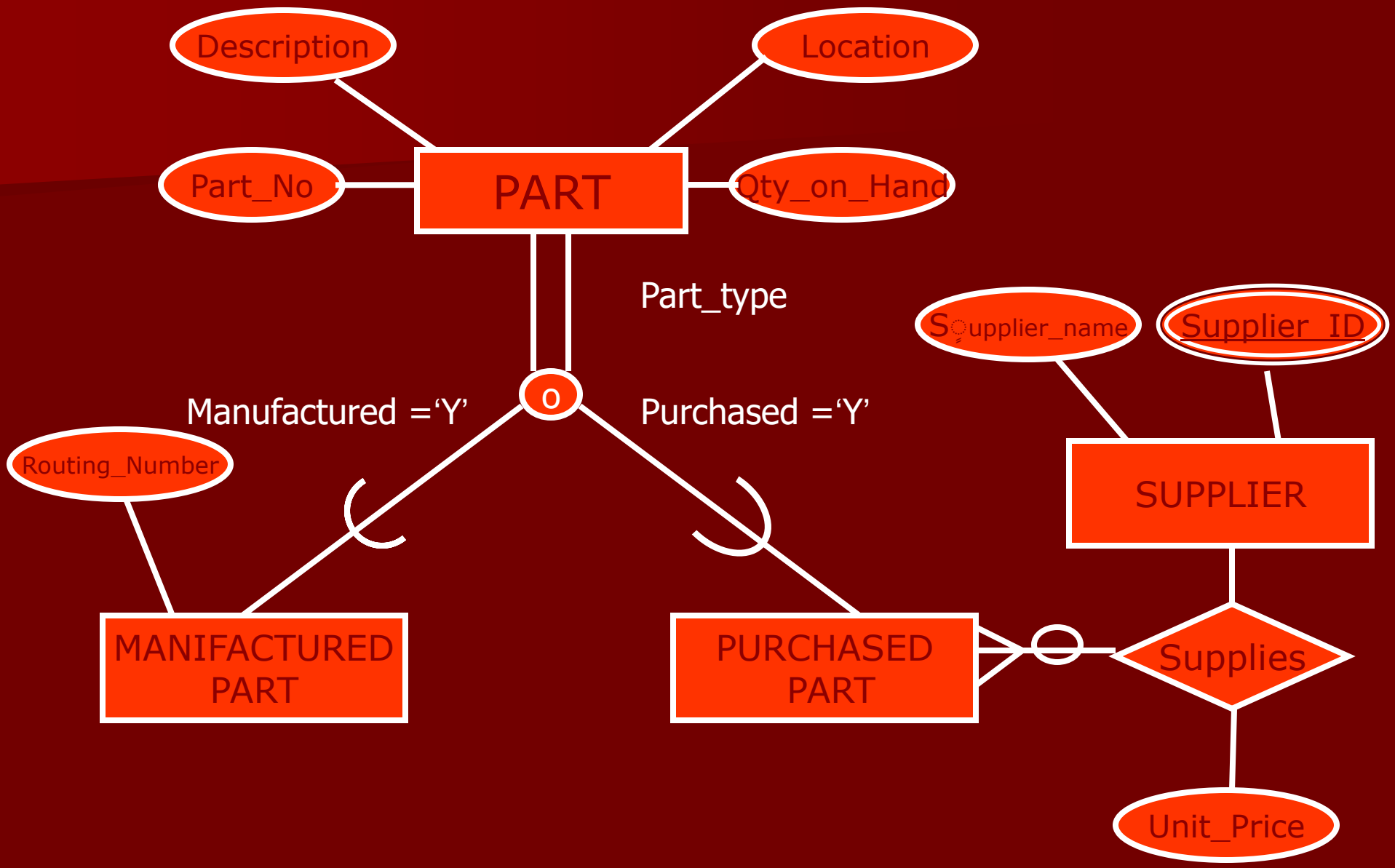
Ex: EMPLOYEE , here a new attribute Employee_Type added to the Supertype , the condition put to the left side of the line .

2. Overlap Subtype:

Ex: PART , here a new attribute Part_Type added , which is a composite attribute , represented by logical expression .

Type of part	Manufactured?	Purchased ?
Manufactured only	‘Y’	‘N’
Purchased only	‘N’	‘Y’
Manufactured& Purchased	‘Y’	‘Y’

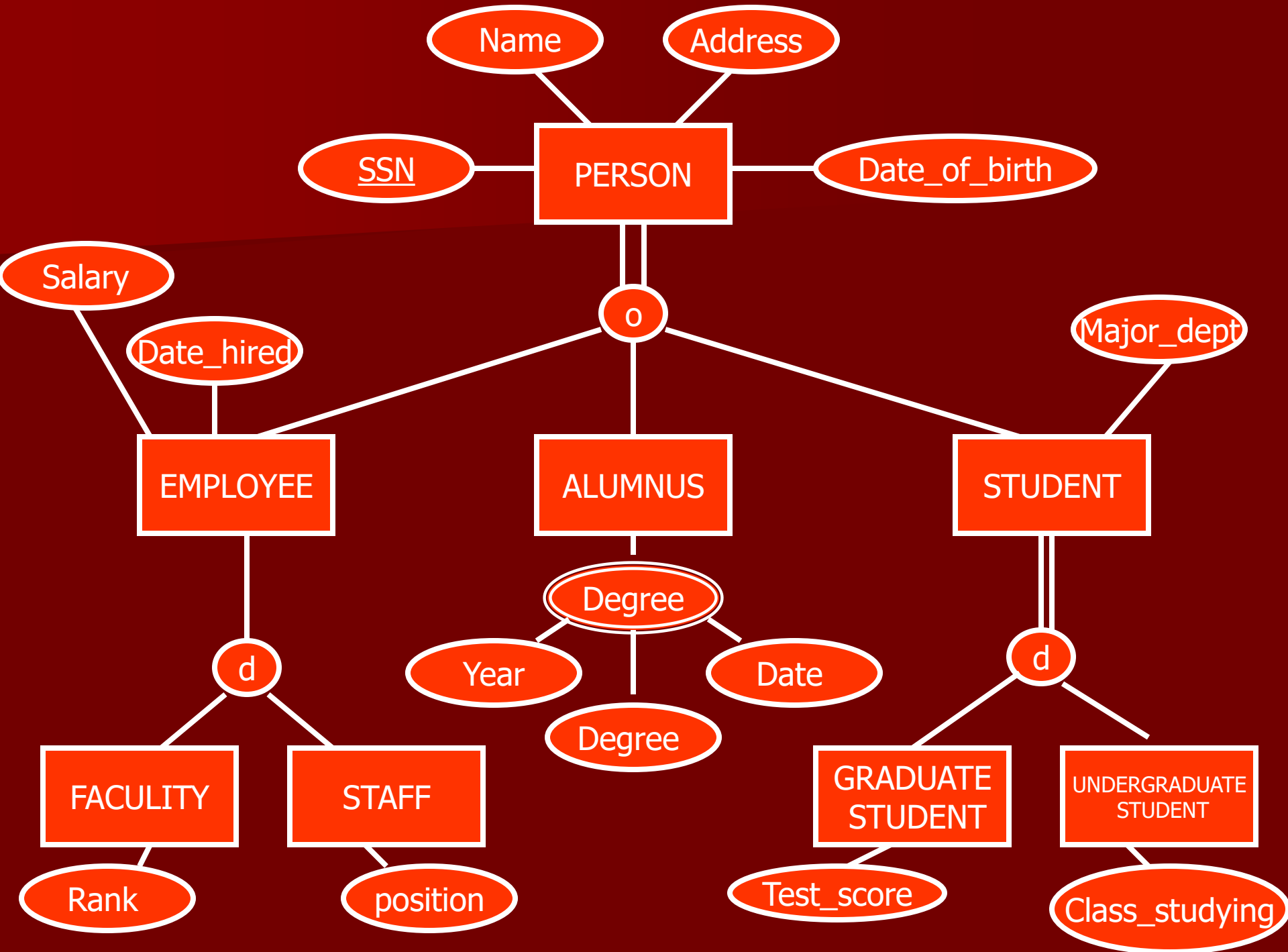

Emploee_name
Address
Employee_Number
Date_hired
EMPLOYEE
Employee_type
H
d
C
S
HOURLY EMPLOYEE
SALARIED EMPLOYEE
CONSULTANT
Hourly_Rate
Annual_Salary
Stock_Option
Contract_Number
Billing_Rate


Description
Location
Part_No
PART
Qty_on_Hand
Part_type
Supplier_name
Supplier_ID
Manufactured ='Y'
o
Purchased ='Y'
Routing_Number
SUPPLIER
MANIFACTURED PART
PURCHASED PART
Supplies
Unit_Price

Supertype/Subtype Hierarchy

a hierarchal arrangement of Supertypes and subtype where each subtype has only one Supertype .

- Attribute in the root are for all its subtypes, also the subtype if it has Supertype attributes. They will be to all the subtypes.
- Subtype inherit all the attributes in the Supertype connected directly (or indirectly) with it .


Name
Address
SSN
PERSON
Date_of_birth
Salary
o
Date_hired
Major_dept
EMPLOYEE
ALUMNUS
STUDENT
Degree
d
Year
Date
d
Degree
FACULITY
STAFF
GRADUATE STUDENT
UNDERGRADUATE STUDENT
Rank
position
Test_score
Class_studying

Some Solved Examples about ER-Model & EER-Model:

- Example(1): Draw the ER-Model for a

" COMPANY" which has the attributes (CName, CType, CId, CAddress, CProduct_name ,CAgent)

that have Two relations, the first one is (Purchasing) with "MATERIAL" which has the attributes (MName, MSupplier,MMan_Date MExp_Date, MExp_time ,MUnit_Price).

The second relation is (Marketing) with

"PRODUCT" which has the attributes(PName ,PId , PUnit_Price,PUsage).

- Note that there is an attribute Marketing_date applied on the relationship (Marketing)

Solution :

A: There are three entity types (COMPANY , MATERIAL ,PRODUCT)

The attributes for COMPANY are : CName, CType, CId, CAddress, CProduct_name ,CAgent

1. CAddress is a composed attribute, at least (Buildind_no and Street_no)

2. The COMPANY has many Agents , so CAgent is a multivalued attribute.

3.CId is the primary key for entity type "COMPANY"

The attributes for MATERIAL are:(MName, MSupplier, MMan_Date, MExp_Date, MExp_time, MUnit_Price)

The MSupplier is a multivalued attribute since we can have more than one supplier for the MATERIAL.

MExp_time is calculated from (MMan_date – Mexp_Date) , so it is derived attribute.

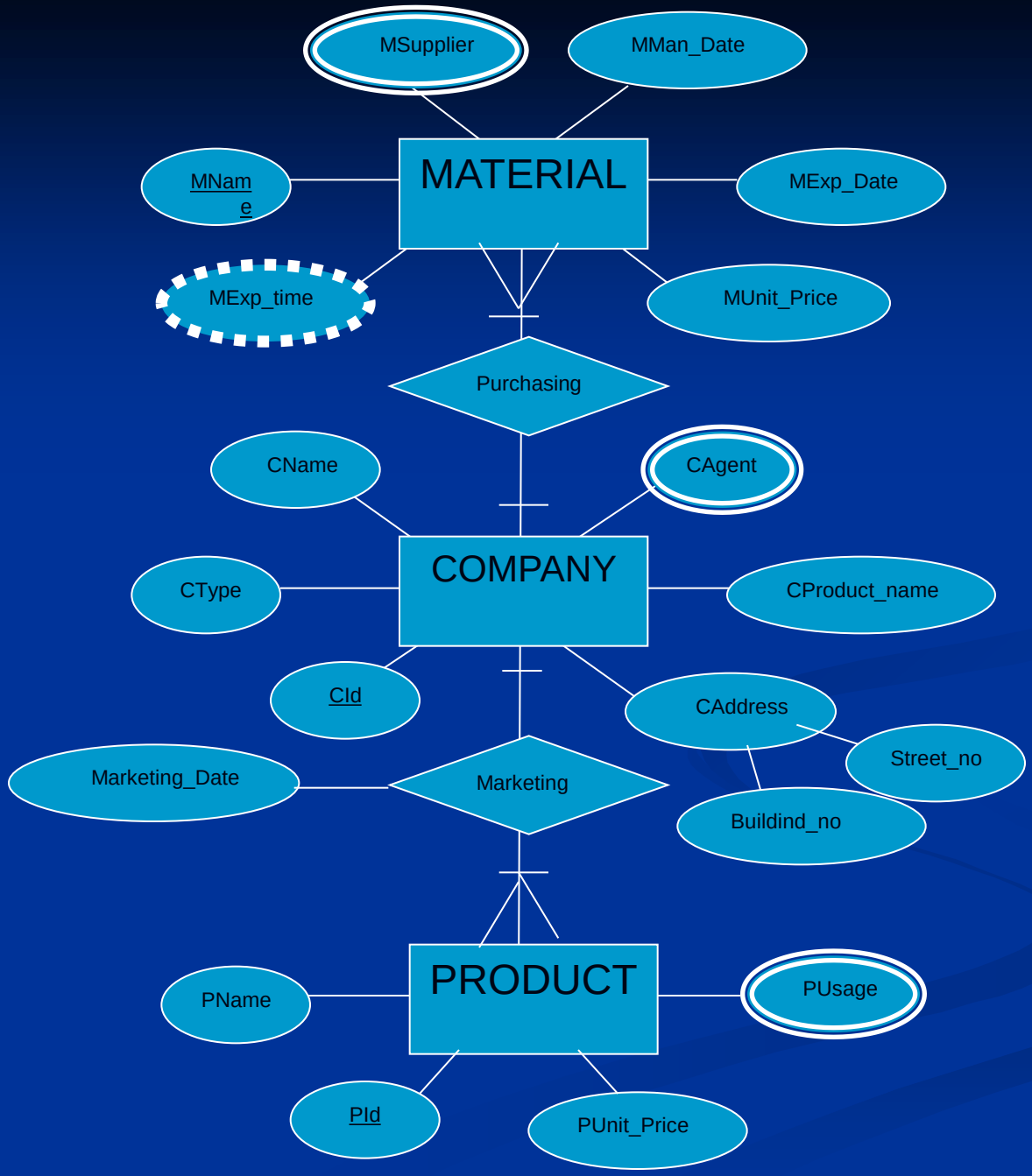
MName is the primary key for material

The attributes for PRODUCT are (PName ,PId , PUnit_Price,PUsages)

1. We can expect there are many usages for one product , so is is a multivalued attribute.

2.the primary key for the product is PId.

- B: there are tow relations (Purchase & Marketing). Marketing has an attribute (Marketing_Date), the relation from COMPANY to MATERIAL is one –to-many, and from COMPANY TO PRODUCT is also one –to-many, Since the COMPANY can buy many MATERIALs , and sell many PRODUCTs, all of them are mandatory because the COMPANY in this example should buy MATERIAL to produce a PRODUCTs, and the all the relations are binary.


MSupplier
MMan_Date
MATERIAL
MNam
e
MExp_Date
MExp_time
MUnit_Price
Purchasing
CName
CAgent
COMPANY
CType
CProduct_name
CId
CAddress
Street_no
Marketing_Date
Marketing
Buildind_no
PRODUCT
PName
PUsage
PId
PUnit_Price

Example (2):

Suppose you have a Mathematical PROBLEM to solve, you need VARIABLEs to use, and OPERATORs. How can you represent that in ER-Diagram?

If you know that the attributes for

PROBLEM (Problem_No, Problem_Degree, Problem_body).

VARIABLE (Variable_Name, VData_Type).

OPERATOR are (Operator_Symbol, Operator_Operands).

The relation among the Three entity types is "Solve" which have the attribute (No_of_Variables)

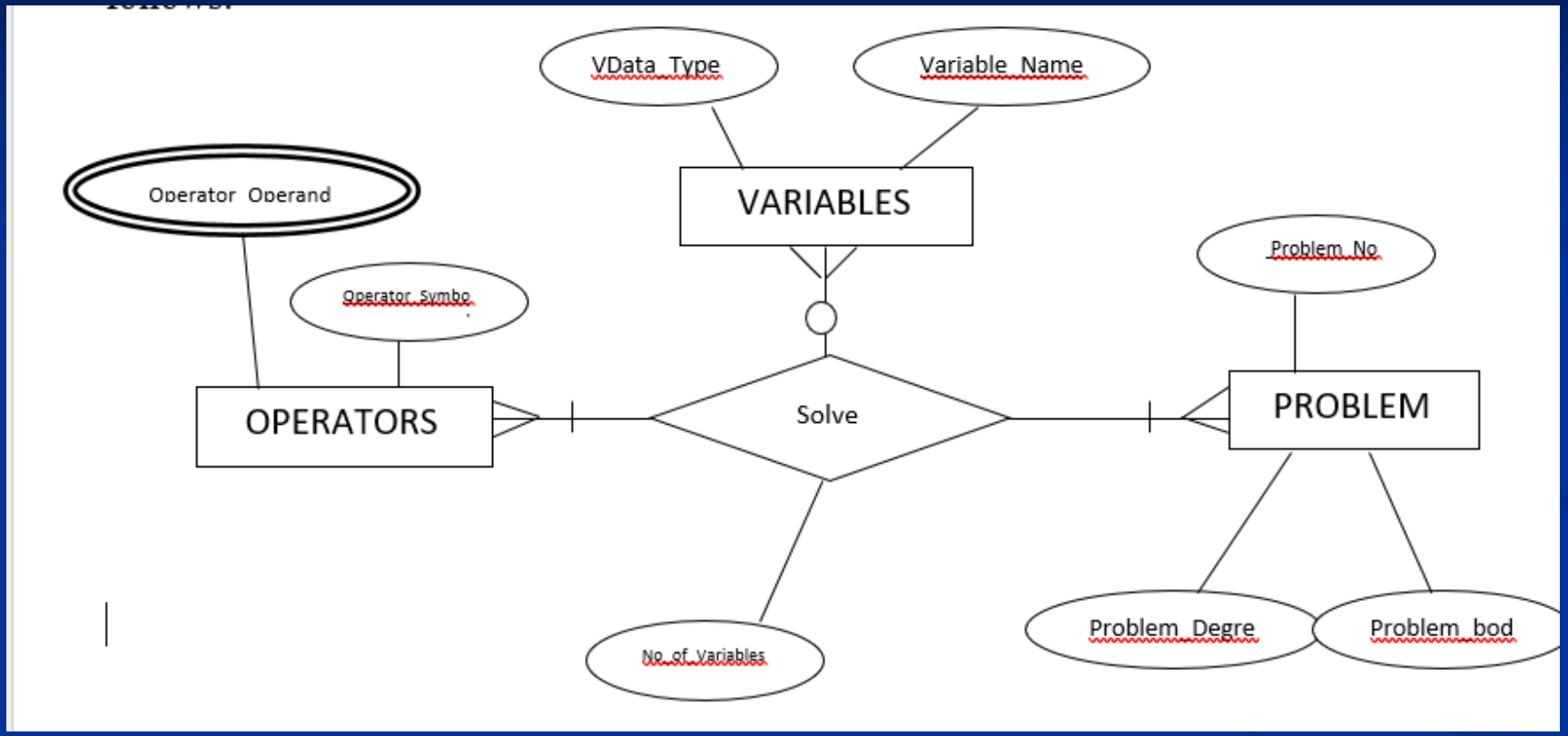
The Solution:

A: we have three entity types: PROBLEM, VARIABLE, and OPERATOR

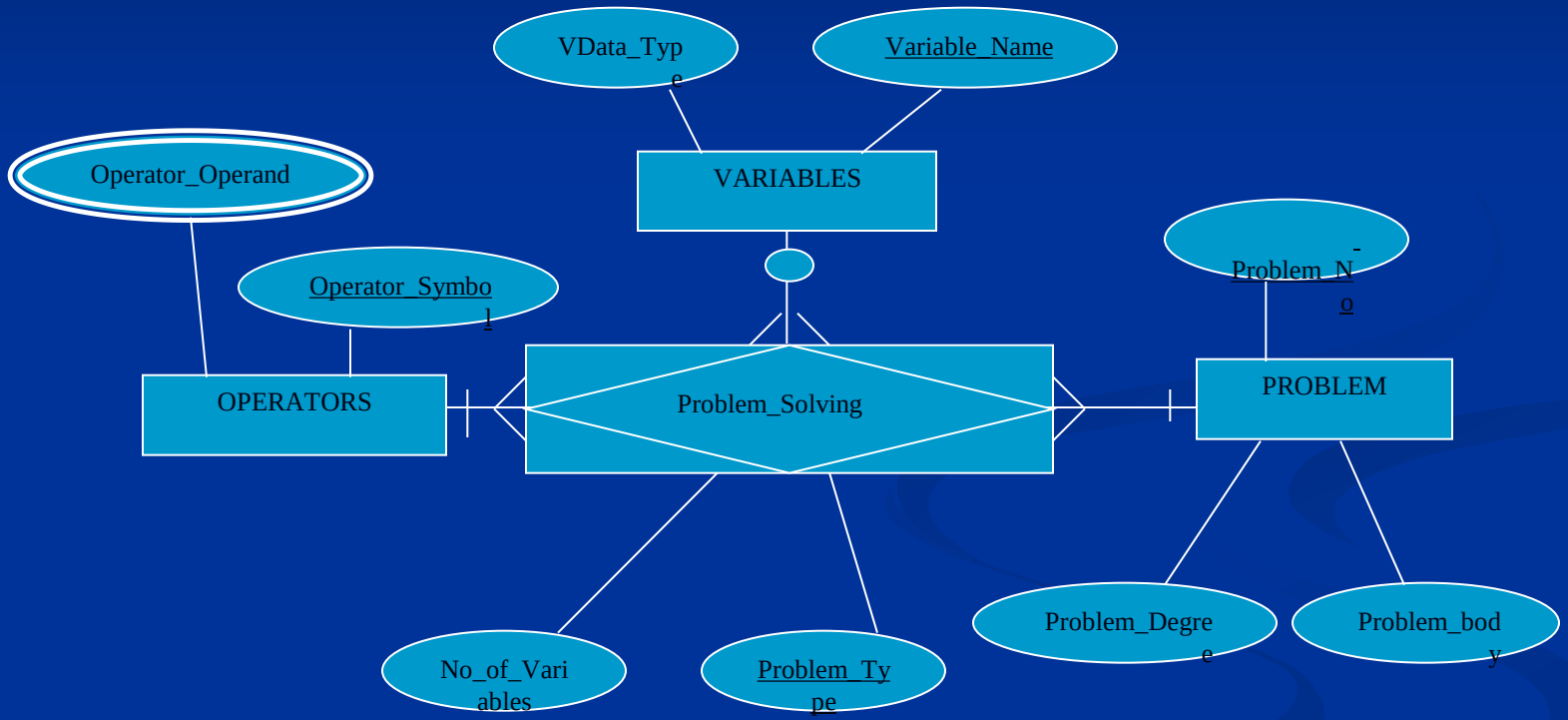
1. For the PROBLEM, Problem_no is the primary key.
2. For the VARIABLE Variable_Name is the primary key,and Operator_operand may have more than one so it is a multivalued attribute
3. For the OPERATOR Operator_Symbol is the primary key.

B: We have one relationship (Solve)

Here we have a ternary relationship, all of them are many to many, mandatory for OPERATOR and PROBLEM, optional for VARIABLE, and the diagram as follows:


VData_Type
Variable_Name
VARIABLES
Operator_Operand
Operator_Symbo
OPERATORS
Solve
Problem_No
PROBLEM
No_of_Variables
Problem_Degre
Problem_bod

Since we can Add a new attribute to the Relationship Problem_Name which is the primary key, we can get an associative entity (PROBLEM_SOLVING) as follows :



Example(3) : if you have a TOYS shop , which sell Dolls , Robots and Childs Guns ,could you represent them as separated entities? Merge them in one general entity type. Then make a relation ship between the child who buying the toy and the shop. Also, make a relation between the shop and the main store which supply the store by toys. Use EER-Diagram to represent your model

The solution:

1.We can recognize three entity types for the toys (DOLLS , ROBOTS, GUNS) and three other entities SHOP , CHILD and MAIN_SORE.

2. The attributes for :

The DOLL are(Name,Unit_Price, Serial_No,Material,Size,Age)

The ROBOTS are (Name Unit_Price, Serial_No,Material,Size Age,No_of_batteries)

The GUN are (Name Unit_Price, Serial_No,Material,Size Age,No_of_Bullets)

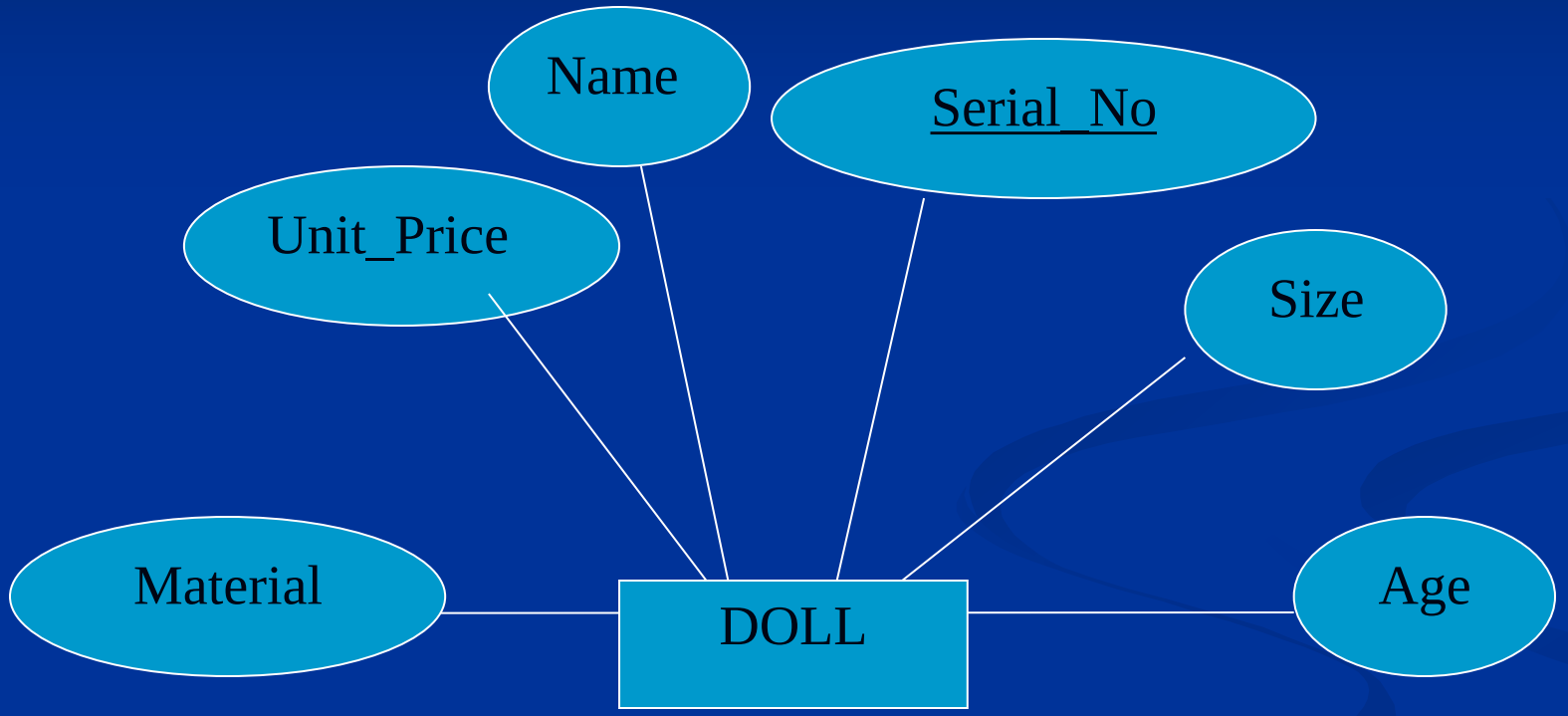
The SHOP(SName,SAddres,SOwner,SSupplier)

The CHILD are(CName,CBirth,CAge)

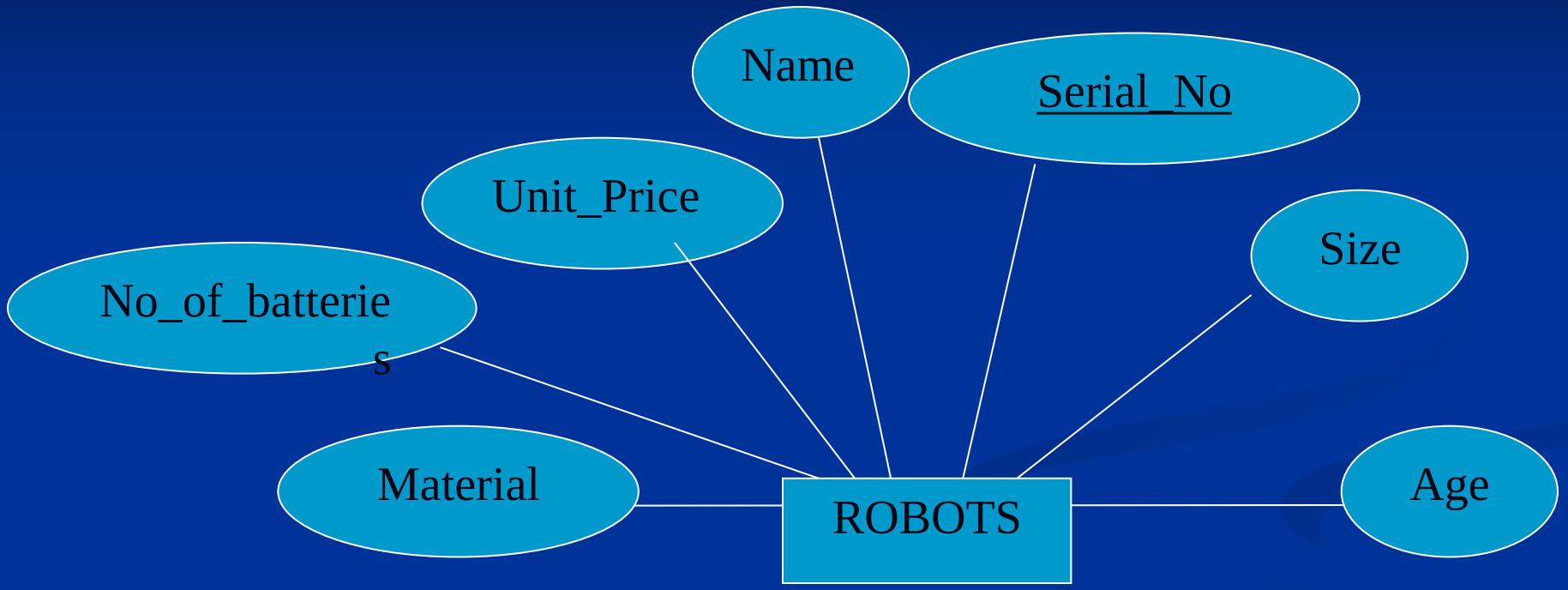
The MAIN_STORE are(MSName,Agent ,MSAddress)

- 4. The attributes:
  - Material, SOwner,SSupplier, Agent are multivalued
  - Cage is derived
  - SAddres, ,MSAddress are composed
  - Serial_no SName, CName , MSName are primary keys
- 5. The relations are buying between the CHILD and the TOYS have one attribute"Cost", Between SHOP and TOY is Contains. Between SHOP and MAIN_STORE is Supplies having one attribute "Bill" which is composed of (Total_Cost and Date)

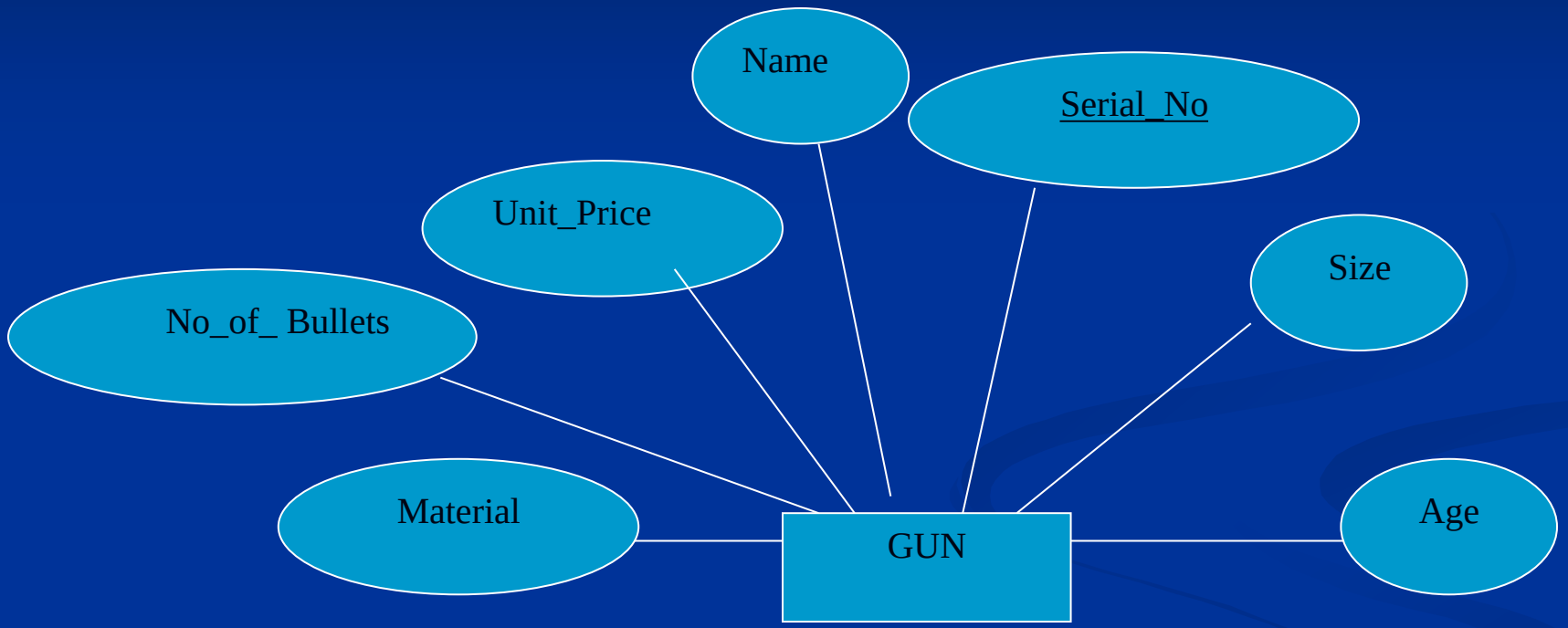
For the DOLL the diagram is:



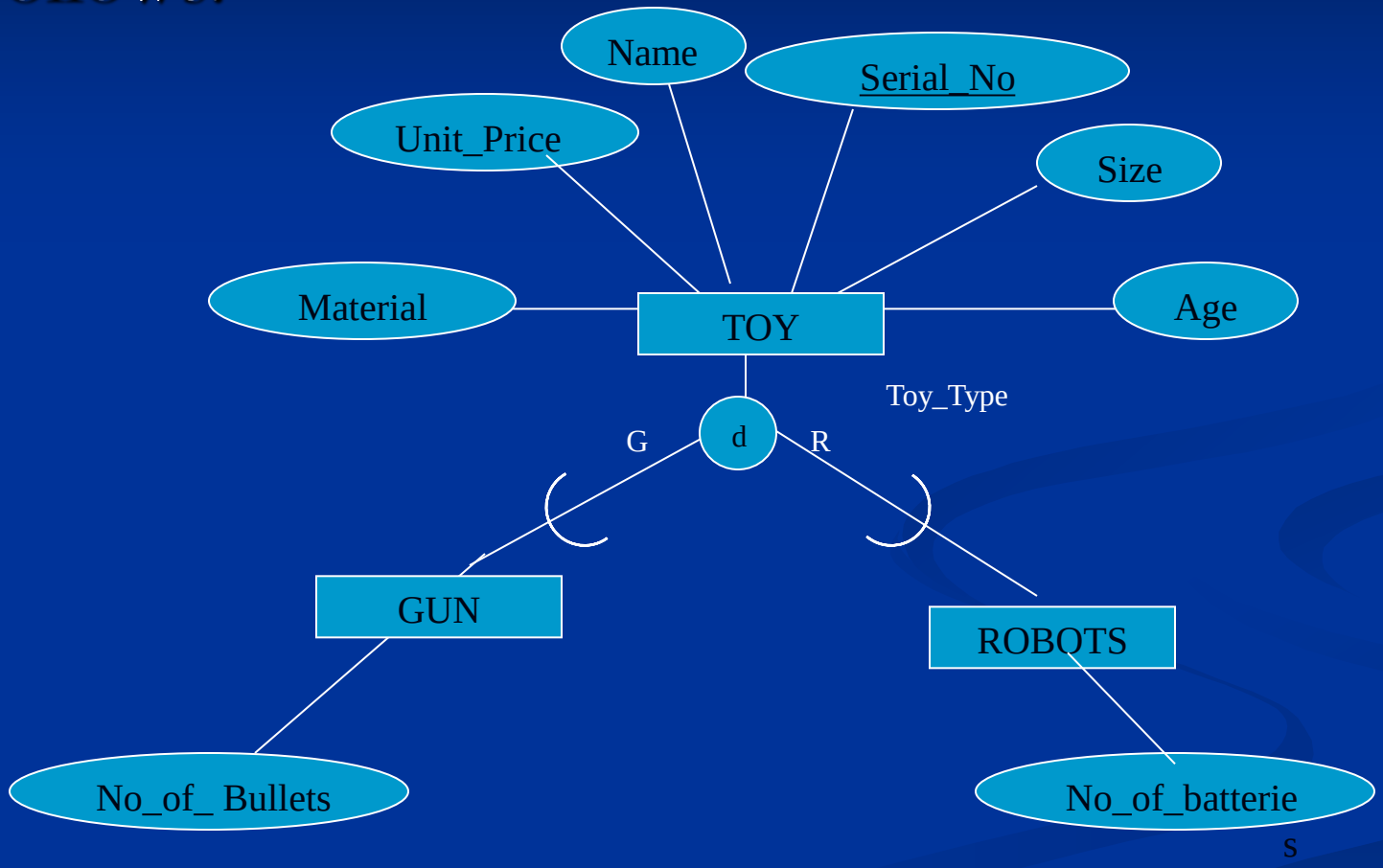
And for the ROBOTS



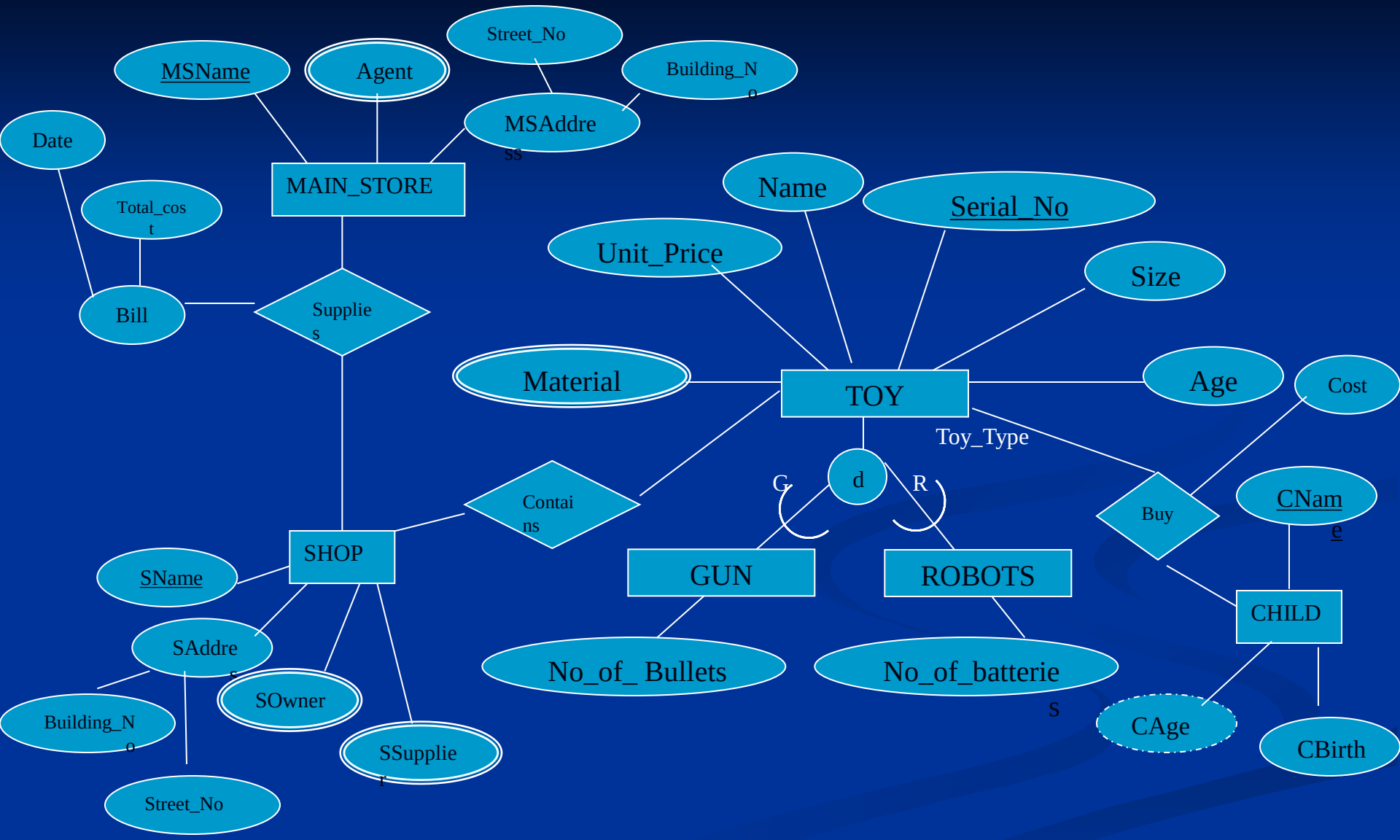
So for the GUNS :



- By using Generalization we can setup a super entity type TOY has two subtypes ROBOTS and GUN as Follows:



- Note that there is no mention for DOLLS because its attributes are the general attributes for the super entity type TOY. Also we have a partial specialization because the toy may be a doll and do not need to be a subtype, and disjoint rule since the toy could not be a Robot or a gun in the same time.


MSName
Agent
Street_No
Building_No
MSAddress
Date
MAIN_STORE
Total_cost
Bill
Supplies
Name
Serial_No
Unit_Price
Size
Material
TOY
Age
Cost
Toy_Type
Contains
d
G
R
Buy
CName
SHOP
SName
GUN
ROBOTS
CHILD
SAddress
SOwner
No_of_ Bullets
No_of_batteries
Building_No
SSupplier
CAge
CBirth
Street_No