DATABASE (1)

3RD CLASS

COMPUTER SCIENCE DEPARTMENT

2nd Lecture – Entity Relationship Model:

Sunday 29th of September 2024

LECTURER :

DR. RAYAN YOUSIF ALKHAYAT

Entity Relationship (ER) Model:

ER Model is a logical representation of the data for an organization or for a business area. Within this model the database can be modeled as:

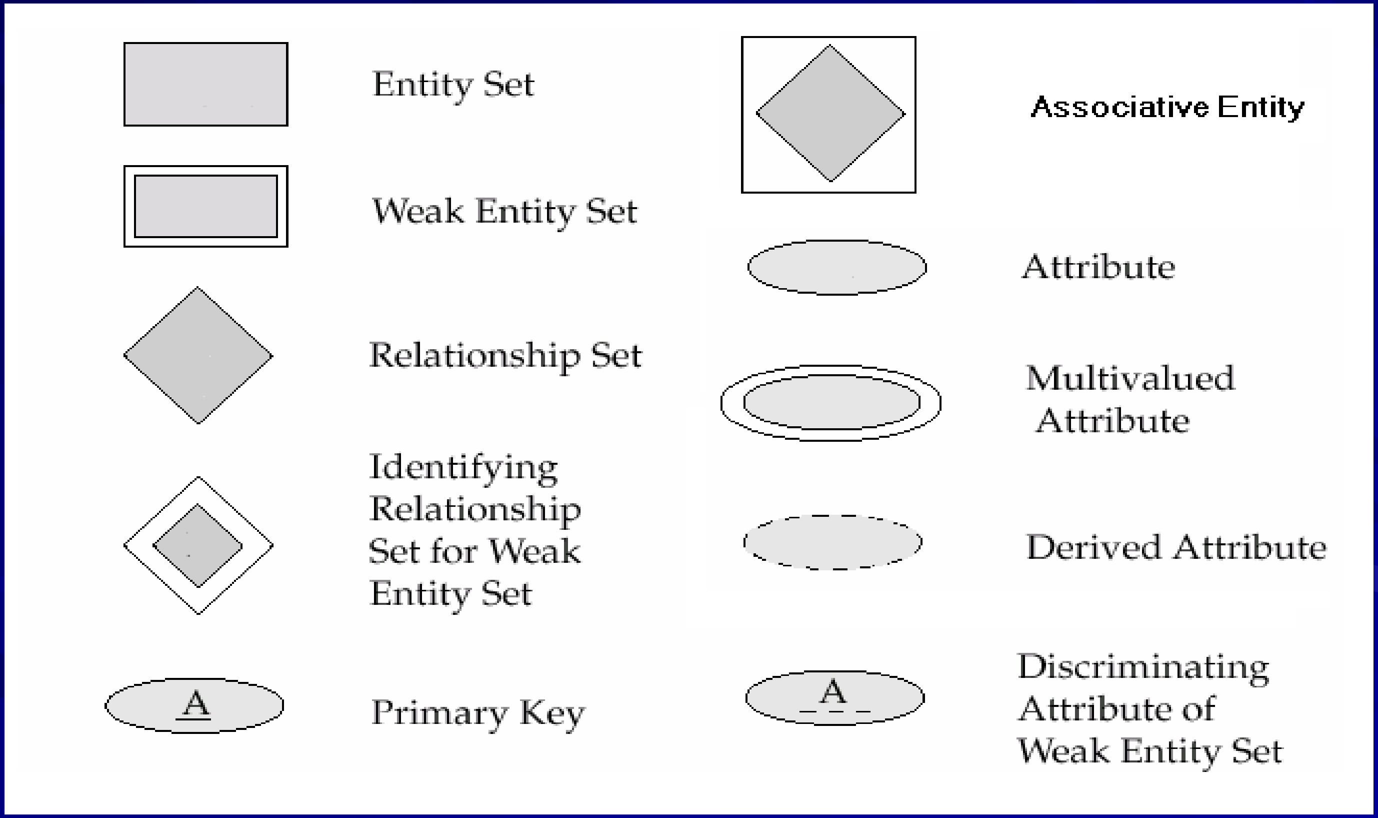
- **1. Collection of Entities.**
- **2. Relationship between Entities.**

Entity Relationship (ER) Diagram

- **ER- Diagram is the graphical representation of an Entity-Relationship Model.**

Entity Relationship (ER) Diagram

The symbols used to draw ER diagram are:



Entity Set

Weak Entity Set

Relationship Set

Identifying Relationship Set for Weak Entity Set

Primary Key

Associative Entity

Attribute

Multivalued Attribute

Derived Attribute

Discriminating Attribute of Weak Entity Set

Entity Relationship (ER) Diagram

- **Notably that:**

➢ Each entity type has a set of attributes associated with it.

➢ In naming attributes, we use an initial capital letter followed by lower case letters, and if attributes consist of two words ,we use underscore character '_' to connect the words and we start each word with capital letter.

➢ An attribute is associated with exactly one entity or relationship.

Entity Relationship (ER) Diagram

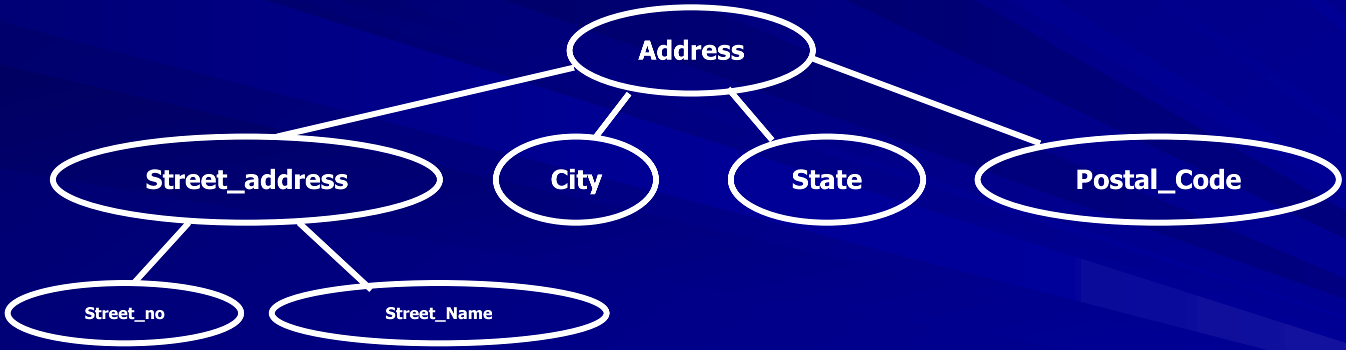
- **As an example, an Entity is STUDENT with attributes: Student_Id ,Student_Name , Address and Phone_Number**

- **Below are two entity instances with attribute values:**

Student_Id = 1209 **Student_Name = Sufyan Salim Mahmood** **Address = Mosul – Alkafaat** **Phone_Number = +96460817543**	**Student_Id = 1190** **Student_Name = Mohammad Zaki Hasan** **Address = Mosul – AlnabiSheet** **Phone_Number = +96460813241**

Attribute Types :

- **1- Simple Attribute :** An attribute that cannot be broken down into smaller components like all attributes associated with STUDENT.
- **2- Composed Attributes**: an attribute that can be broken down into components parts like address.



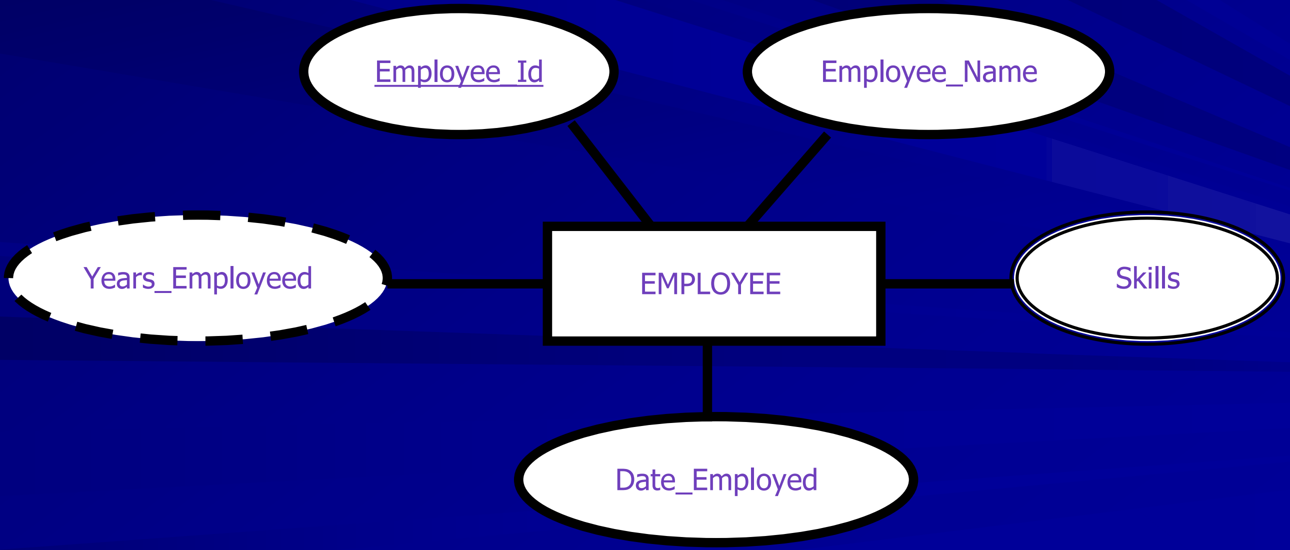
Attribute Types :

- 3-**Single Valued Attribute :** An attribute that take single one value for a given entity instance, like Student_Id

- **4- Multi Valued attribute :** An attribute that may take more than one value for a given entity instance like : EMPLOYEE may have more than one Skill.

- **5- Stored Attribute :** An attribute whose value is fixed and stored .

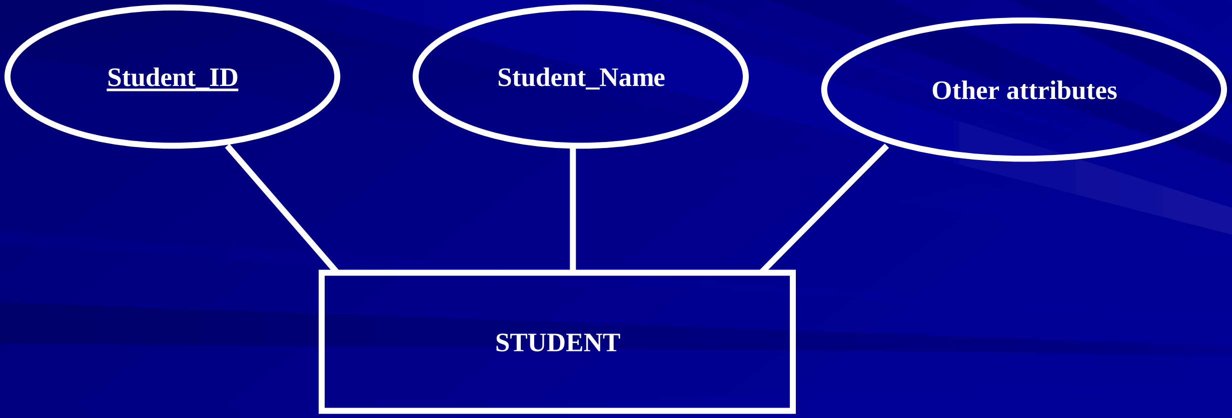
- **6-Derived Attribute :** An attribute whose values can be calculated from related stored attribute values .



Identifiers

- *** Identifier:** an attribute or combination of attributes that uniquely identifies individual instances of an entity type, we express it by underline its name.
- **Note:** If the entity type has one attribute, and it is an identifier, this entity considered "illegal".
- **** Composed Identifier** : An identifier that consists of composite attribute

a. Simple Key attribute :



Student_ID

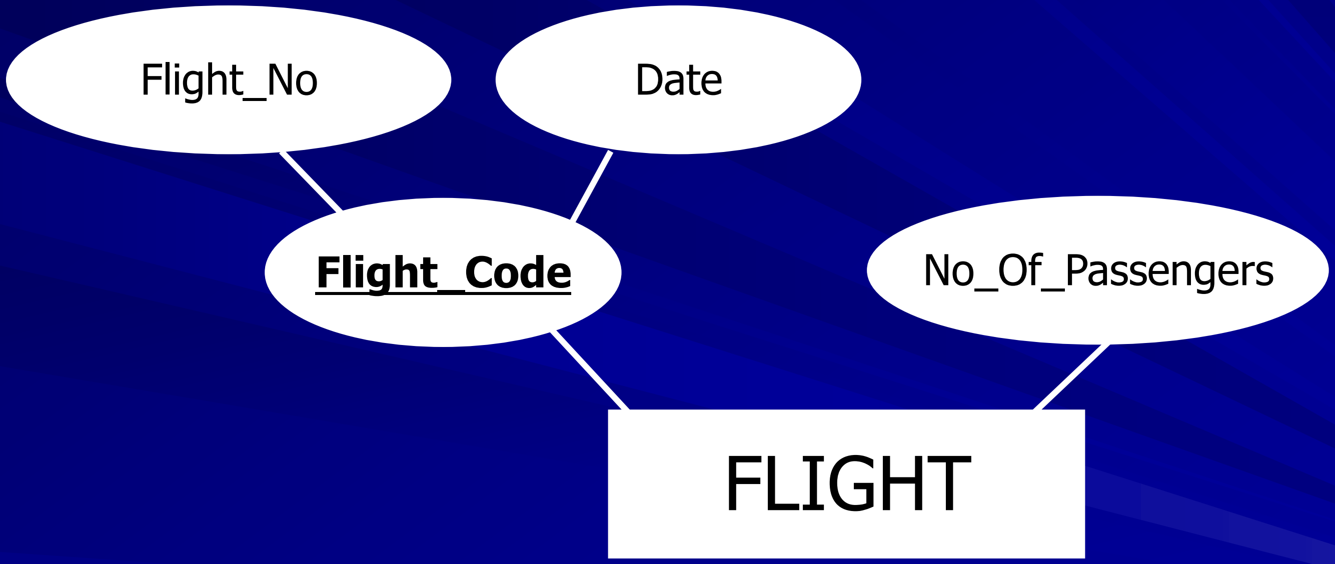
Student_Name

Other attributes

STUDENT

Identifiers

a. Composite Key Attribute :



Entity Types:

- There are three types of entities:

1. **Strong Entity** : an entity that exists independently of other entity types. It has identifier that is an attribute or combination of attributes that uniquely distinguish each occurrence of that entity.

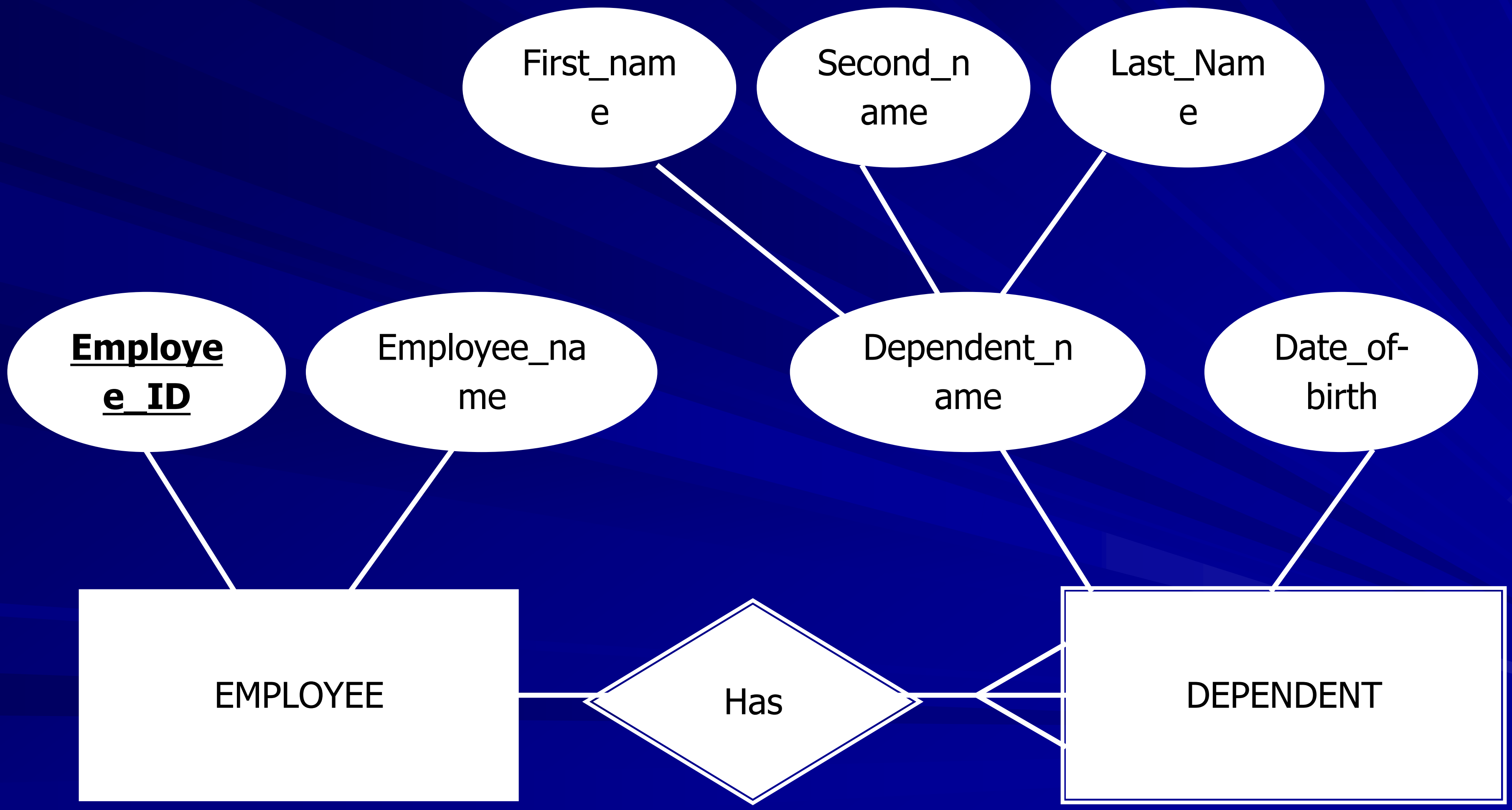
Entity Types:

2. Weak Entity : an entity whose existence depend on some other entity type . It does not have its own identifier. But have an attribute that serves as partial identifier.

- **Full identifier** for weak entity = ***partial identifier + identifier of its owner***
- **Identifying Owner (**owner entity**) the entity on which the weak entity type depends.**
- **Identifying relationship:** the relationship between a weak entity and its owner.

3. Associative Entity.

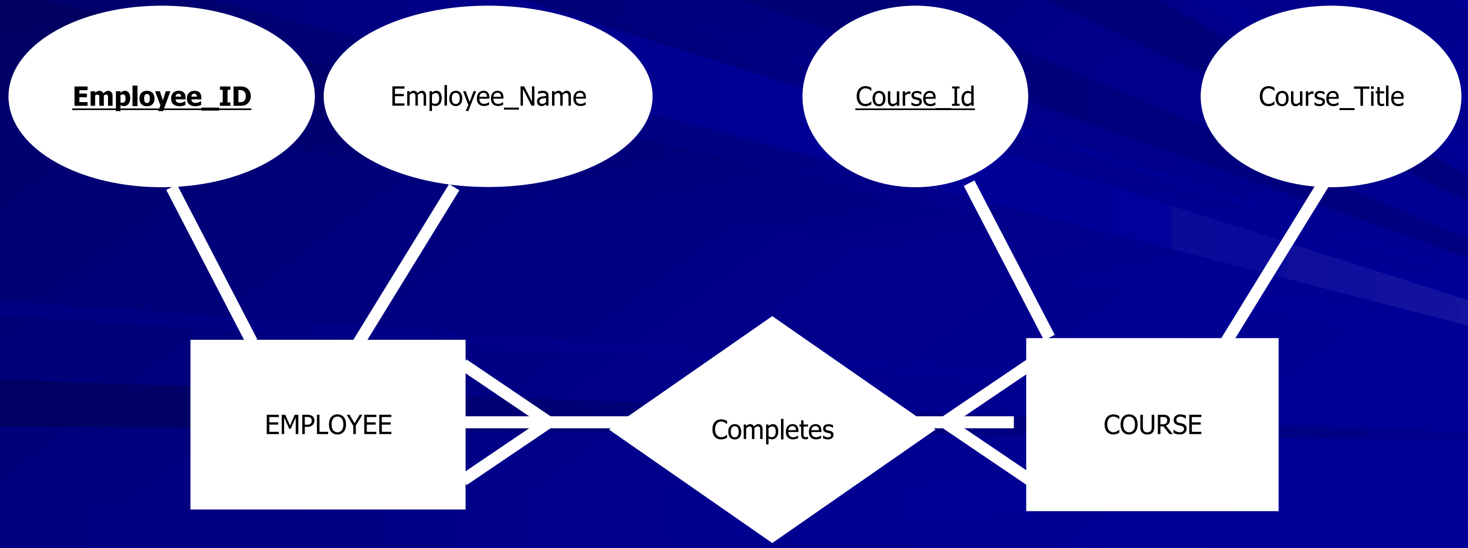
Entity Types:



Relationship:

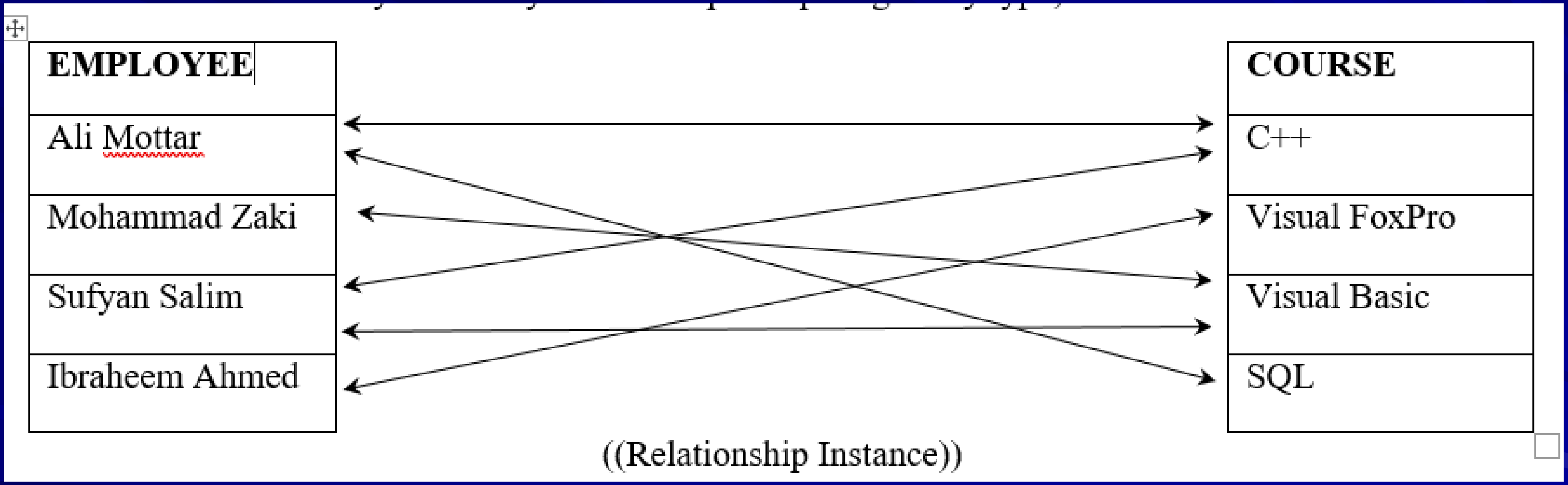
A meaningful association between entities.

The relationship name is created using a short descriptive verb phrase and should be in present tense, which is meaningful to the user in naming the relationship.



Relationship:

- **Relationship Instance** : An association between (or among) entity instances ,where each relationship instance includes exactly one entity from each participating entity type,



EMPLOYEE
Ali Mottar
Mohammad Zaki
Sufyan Salim
Ibraheem Ahmed

COURSE
C++
Visual FoxPro
Visual Basic
SQL

((Relationship Instance))

Relationship:

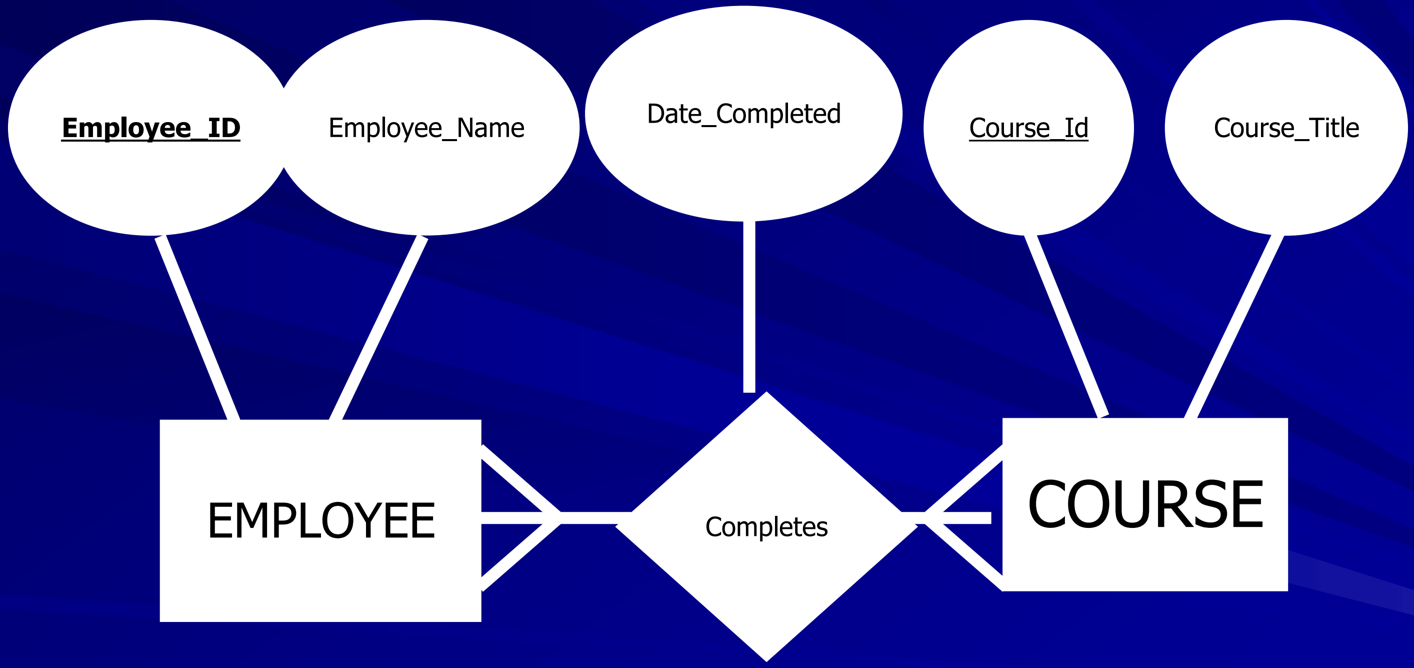
Attributes on Relationships:

- From the previous figure we can consider a table , if we want to add an attribute (Date_Completed), it will be:

EMPLOYEE	COURSE	Date_Completed
Ali Mottar	C++	12-12-2021
Ali Mottar	SQL	12-7-2022
Mohammad Zaki	Visual Basic	10-5-2022
Sufyan Salim	C++	21-7-2022
Sufyan Salim	Visual Basic	2-8-2022
Ibraheem Ahmed	Visual FoxPro	22-8-2022

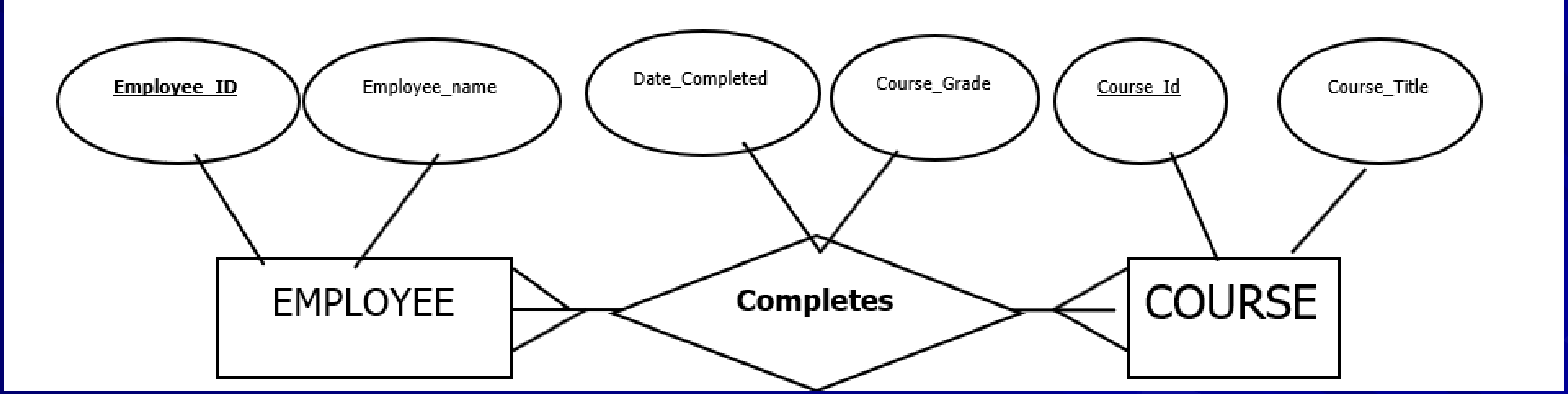
Relationship:

As shown in the following E-R diagram



Relationship:

- Also, we can add another attribute, like the grade of the course (Course_Grade) then the diagram will be:



Associative Entities:

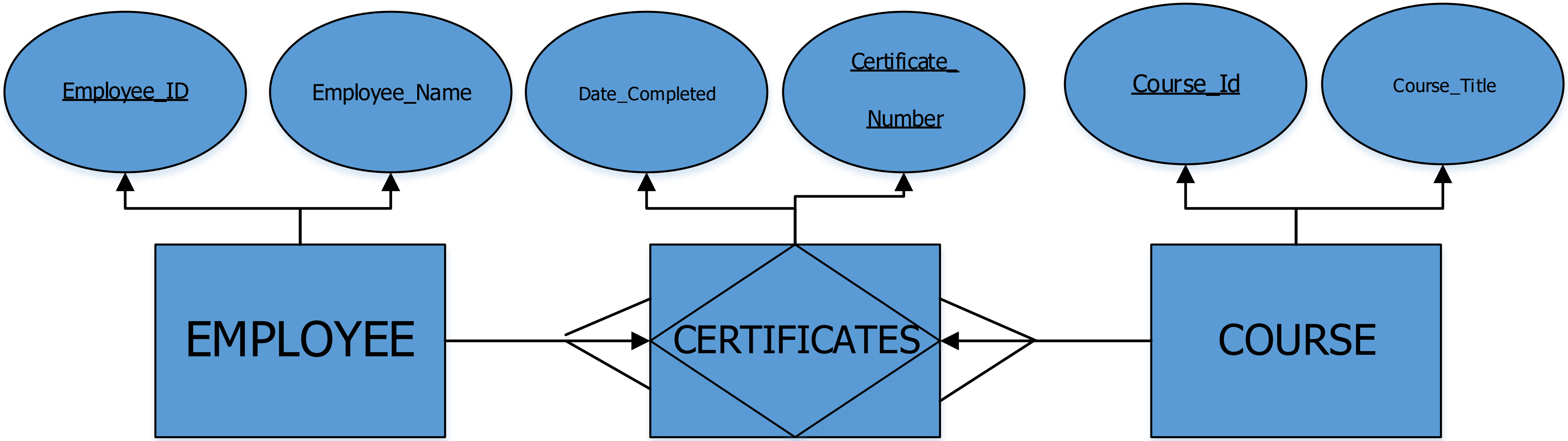
- **Associative Entity** is an entity that associates the instances of one or more entity types and contains attributes that are peculiar to the relationship between those entity instances.
- Associative entity sometimes referred to as [gerunds] since the relationship name [verb] is usually converted to an entity name that is noun (an –ing form of the verb).

How do we know whether to covert a relationship to an associative entity type?

❖ Following are four conditions that should exists:

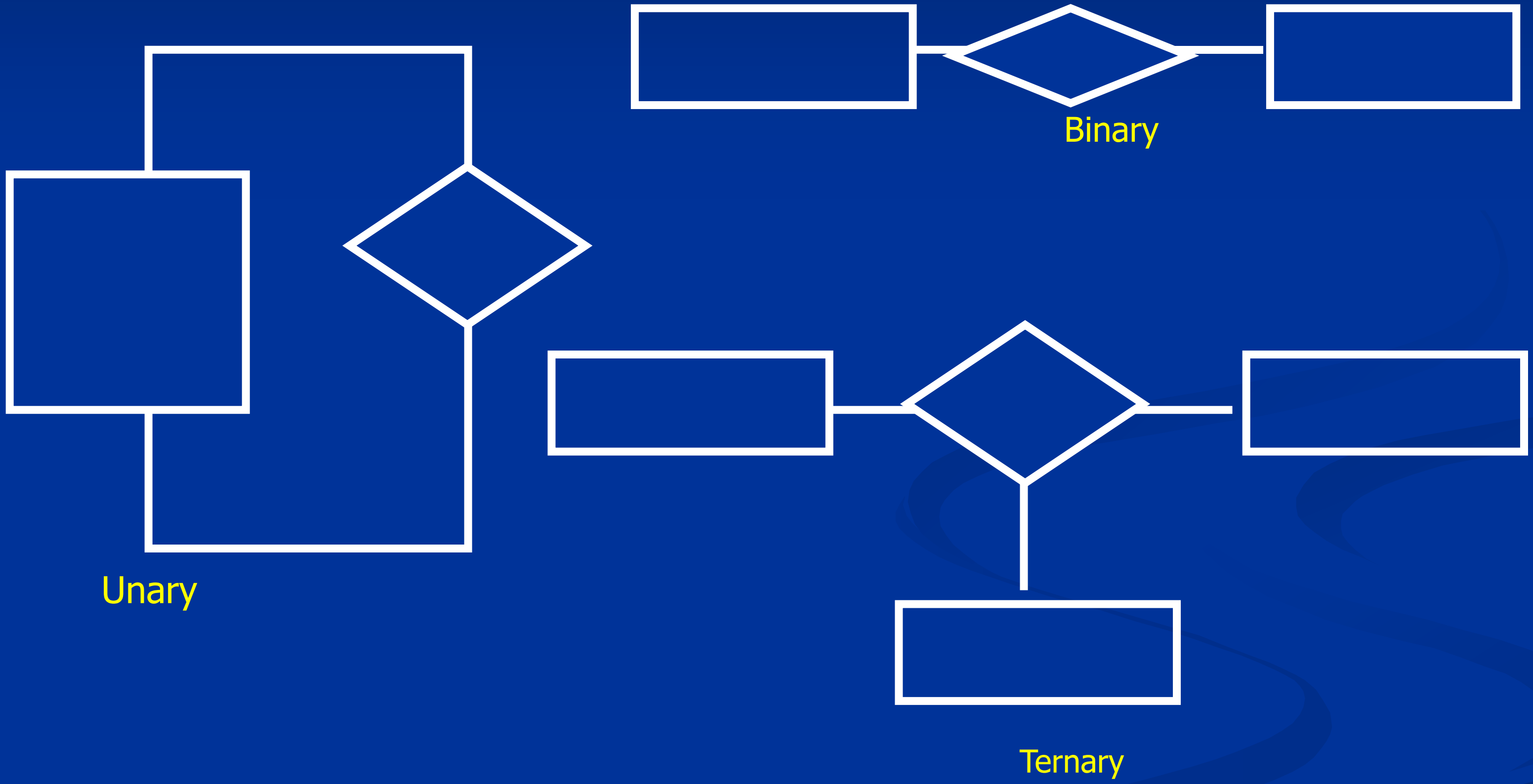
1. All the relationships for the participating entity type are (Many) relationship.
2. The resulting associative entity type has independent meaning to end user, and preferably can be identified with a single attribute identifier .
3. the associative entity has one or more attributes in addition to the identifier .
4. the associative entity participates in one or more relationships independent of the entities related in the associated relationship.

How do we know whether to covert a relationship to an associative entity type?

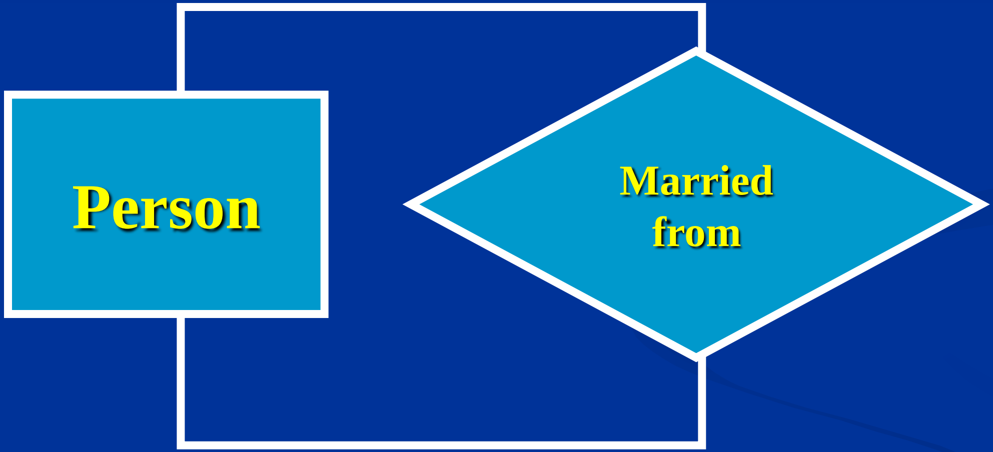


Relationship Degree:

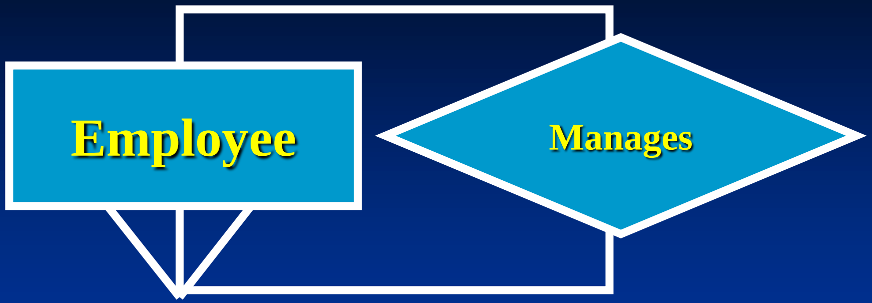
Refers to number of entity sets that participate in a relationship set.



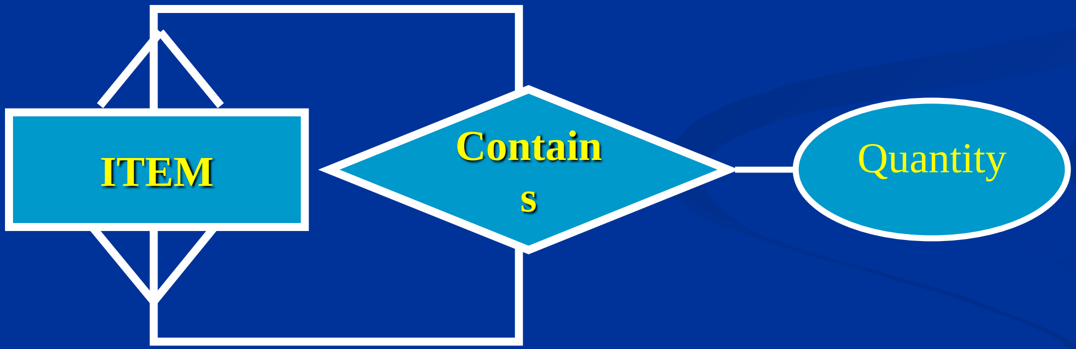
Unary relationship : the relationship between the instances of a single entity type (recursive relationships)



One – to -One

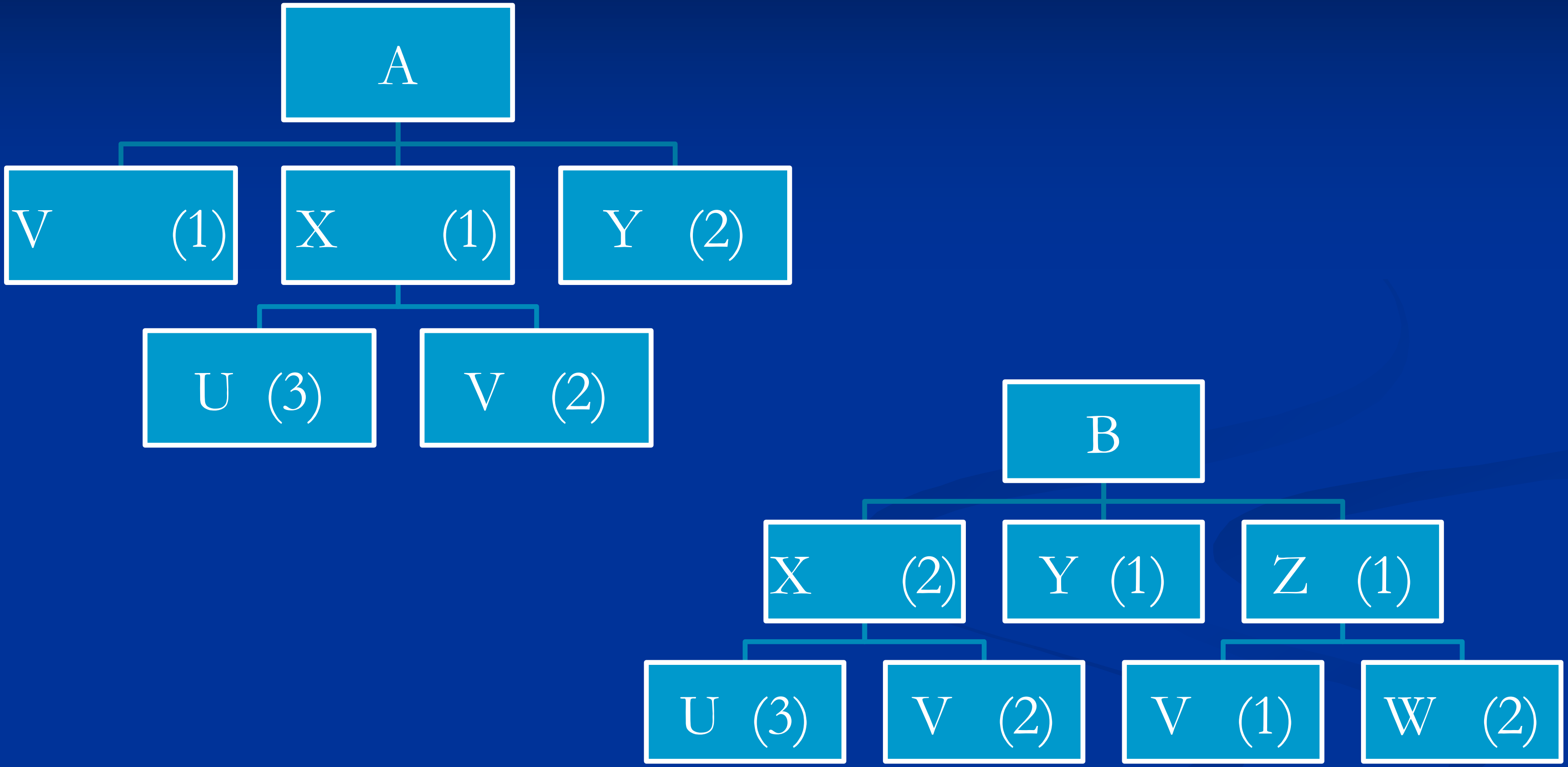


One – to -Many



Many – to – Many

Two Instances for Unary Many- to -Many



A

V (1)

X (1)

Y (2)

U (3)

V (2)

B

X (2)

Y (1)

Z (1)

U (3)

V (2)

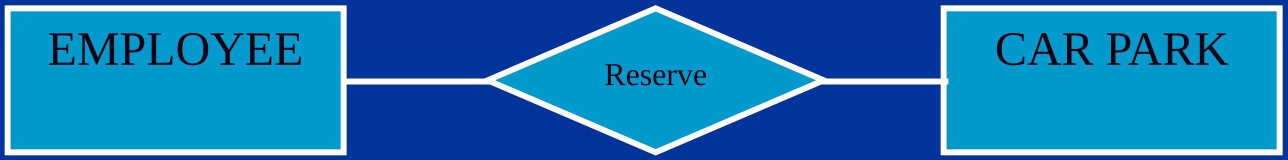
V (1)

W (2)

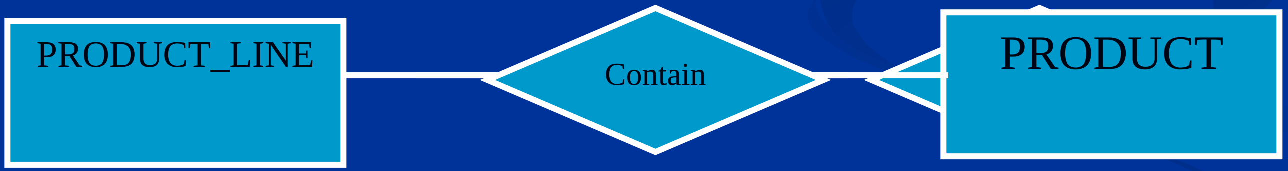
. Binary relationship :the relation involve two entities, i.e. the relation between the instances of two entity type

Generally, most relationship in a database system are binary.

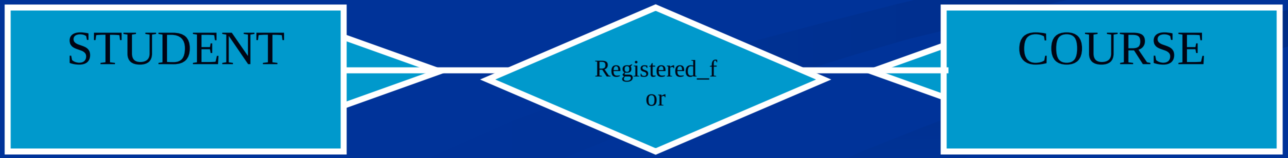
One – to – One



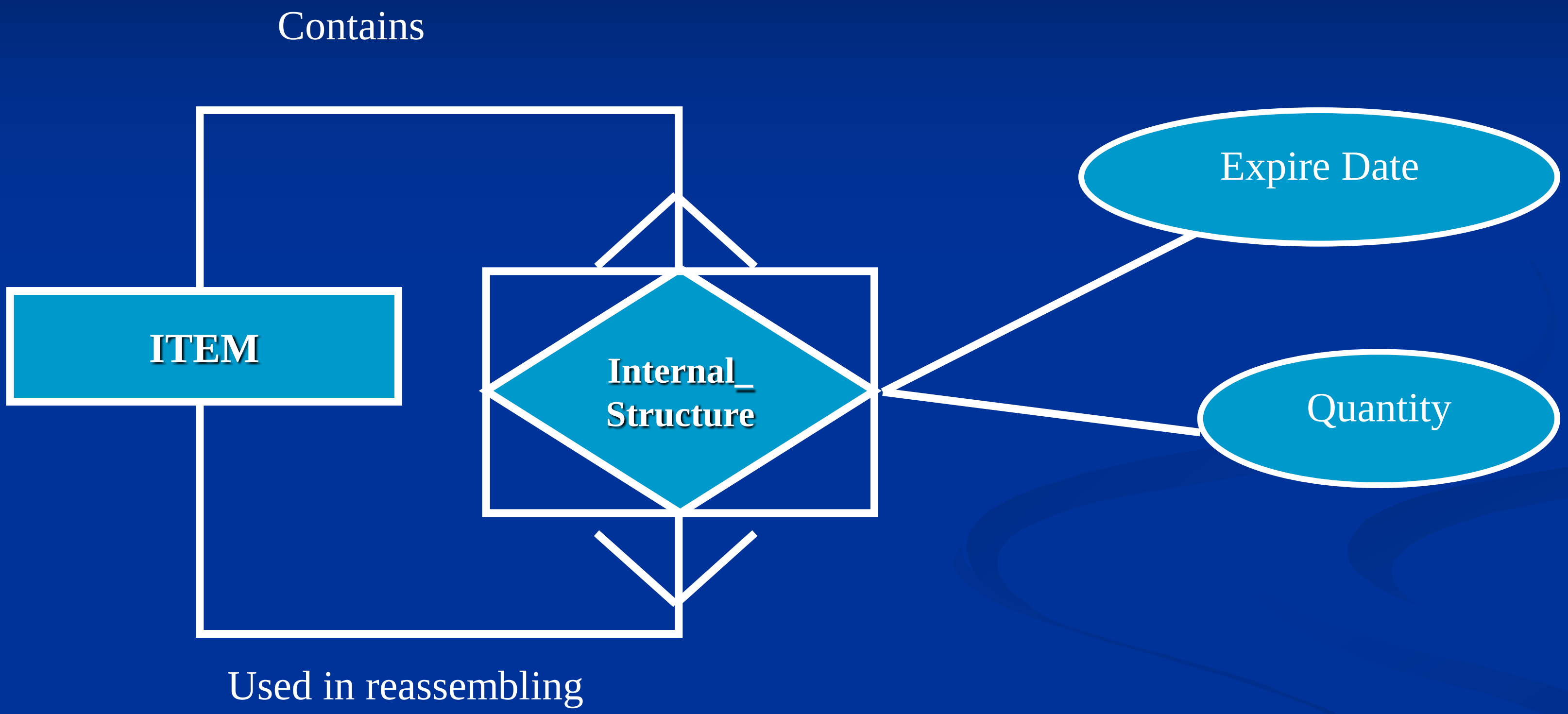
One – to - Many



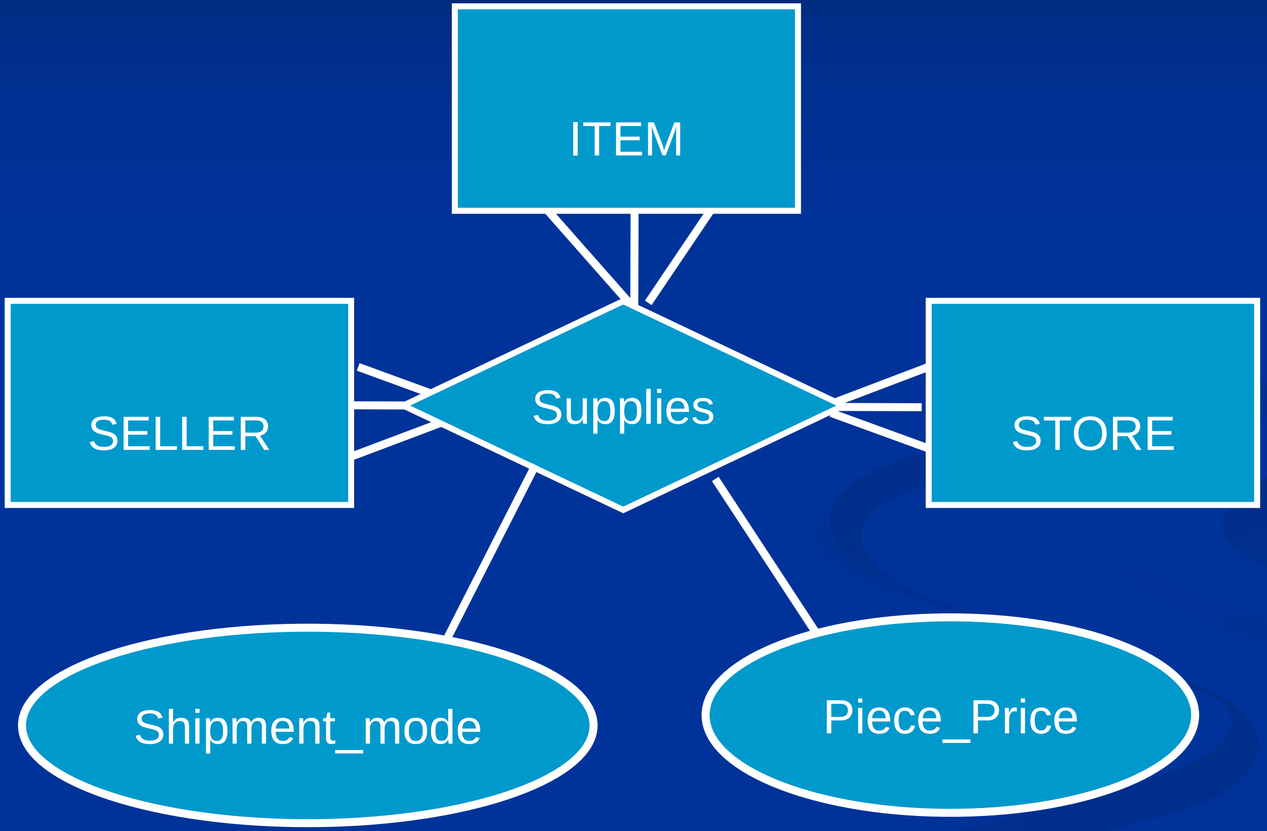
Many – to Many

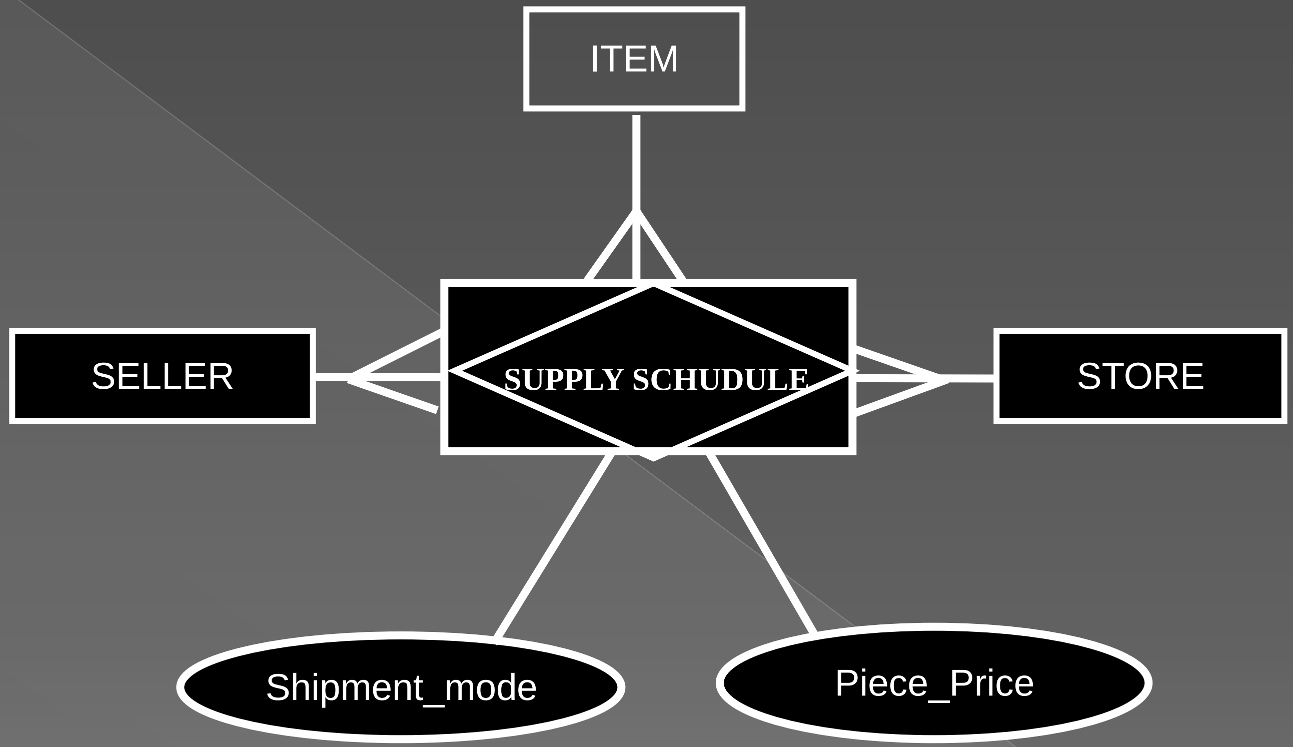


Associative Entity



Ternary relationship sets: involve more than two entity sets. I.e.: simultaneous relationship among the instances of three entity types at least





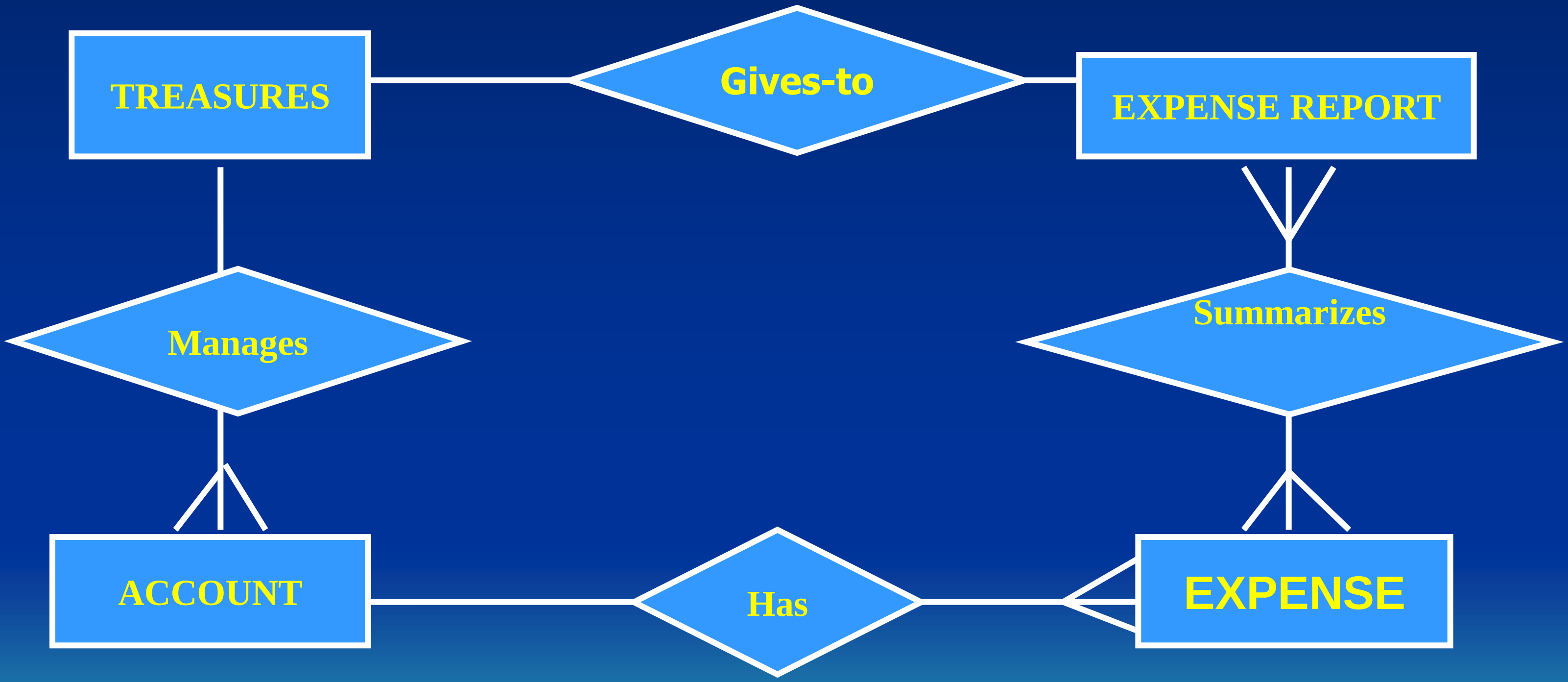
Associative Entity

- E.g. Suppose employees of a bank may have jobs (responsibilities) at multiple branches, with different jobs at different branches. Then there is a ternary relationship set between entity sets *employee, job and branch*

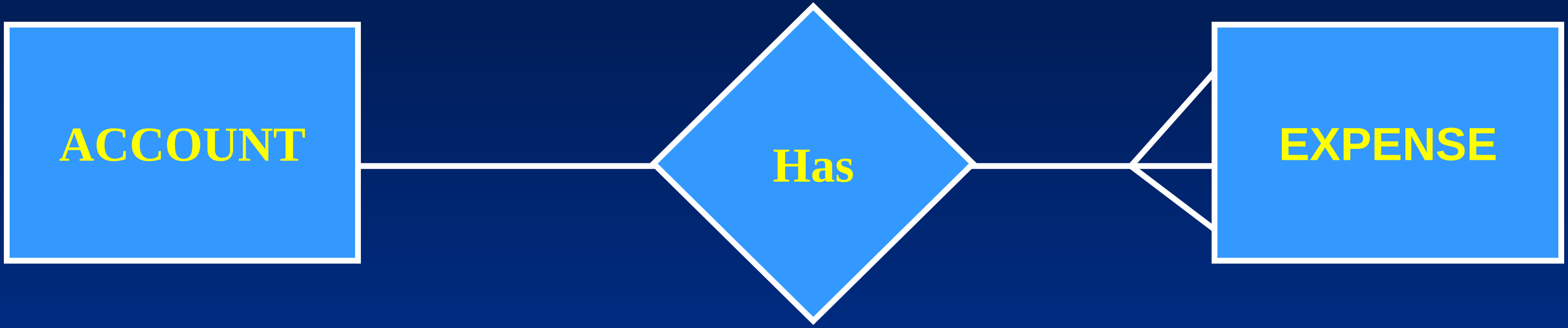
- ((Relationships between more than two entity sets are rare. Most relationships are binary.))

Entity Type versus System Input, Output, or Users

Example of inappropriate entities:



a.System user (Treasurer) and output (expense report) shown as entities

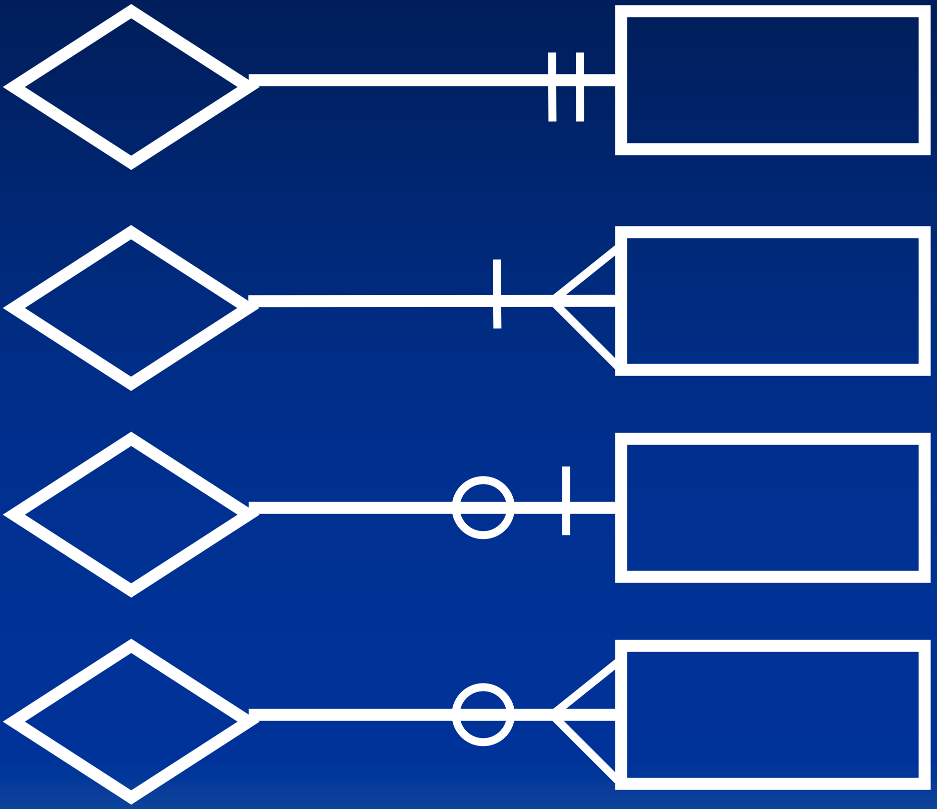


E-R Model with only necessary entities.

Cardinality Constraints :

Specifies the number of instances of one entity that can (or must) be associated with each instance of another entity :

Relationship Cardinality



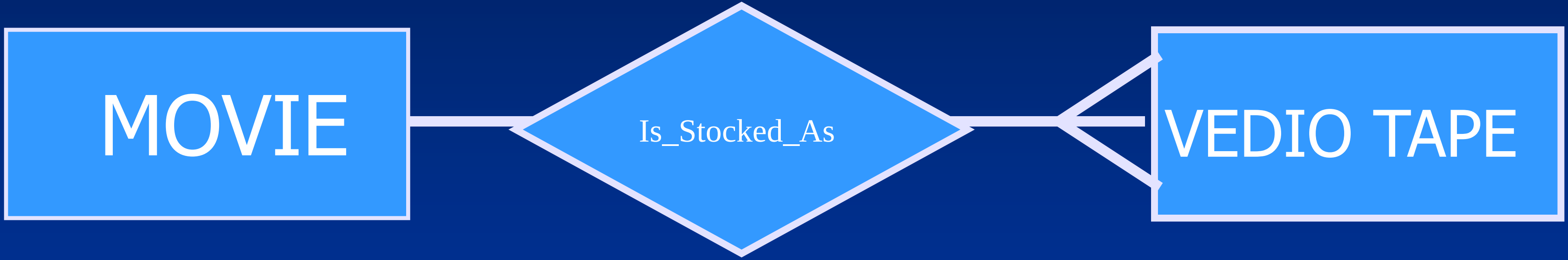
Mandatory One

Mandatory Many

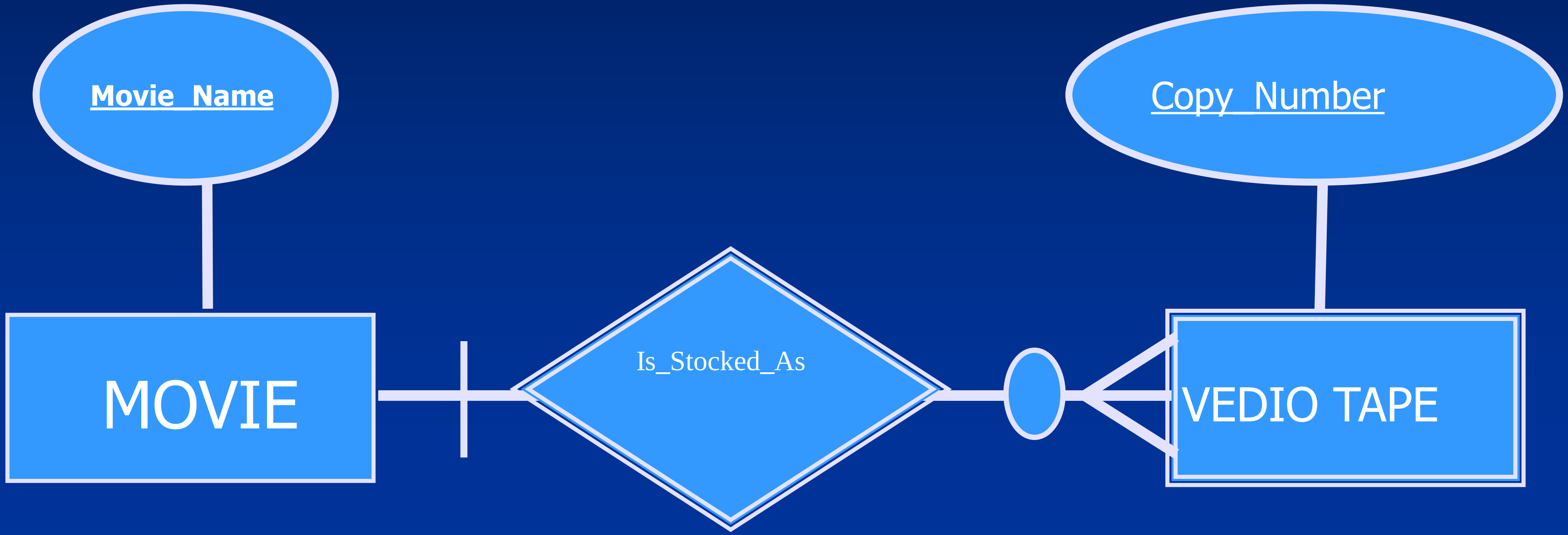
Optional One

Optional Many

A. Basic relationship:



B. Relationship with cardinality constrains



- Minimum Cardinality:

The minimum number of instances of one entity that may be associated with each instance of another entity. When the minimum number of participants is zero, we say that the entity is an *optional participant* in the relationship.

- Maximum Cardinality :

The maximum number of instances of one entity that may be associated with a single occurrence of another entity.

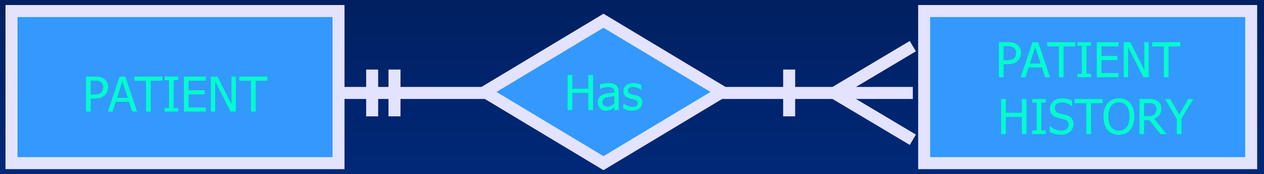
**Note :

- If the maximum cardinality is zero , participation is *optional* ,if the minimum cardinality is one , participation is *mandatory* .
- If the minimum and maximum are both one , this is called *mandatory one* cardinality

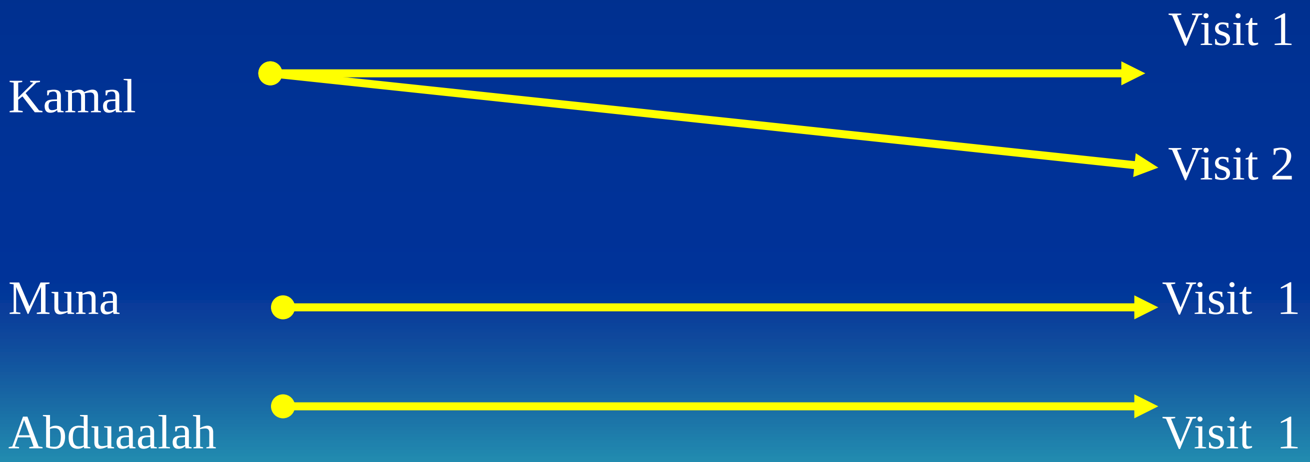
Some Examples :

- PATIENT has PATIENT HISTORY , each patient has one or more patient history , each instance of PATIENT HISTORY “belongs to” exactly one PATIENT.
- EMPLOYEE is assigned to PROJECT each PROJECT has at least one EMPLOYEE assigned to it (some projects have more than one). Each EMPLOYEE may or may not be assigned to any existing PROJECT, or may assigned to one or more PROJECT.
- PERSON IS_Married _to PERSON. This is an optional zero or one cardinality in both directions. Since a person may or may not be married.

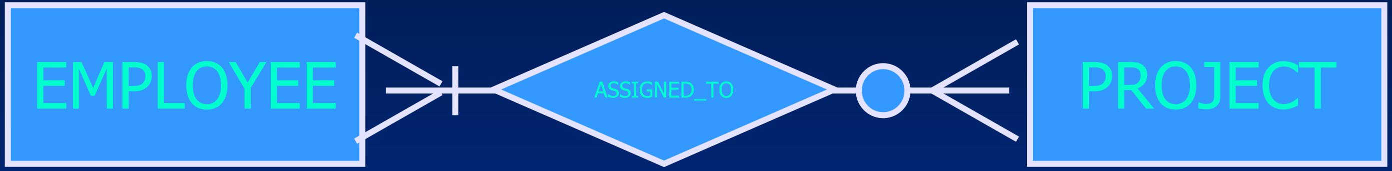
Examples of Cardinality Constrains



Mandatory Cardinalities.



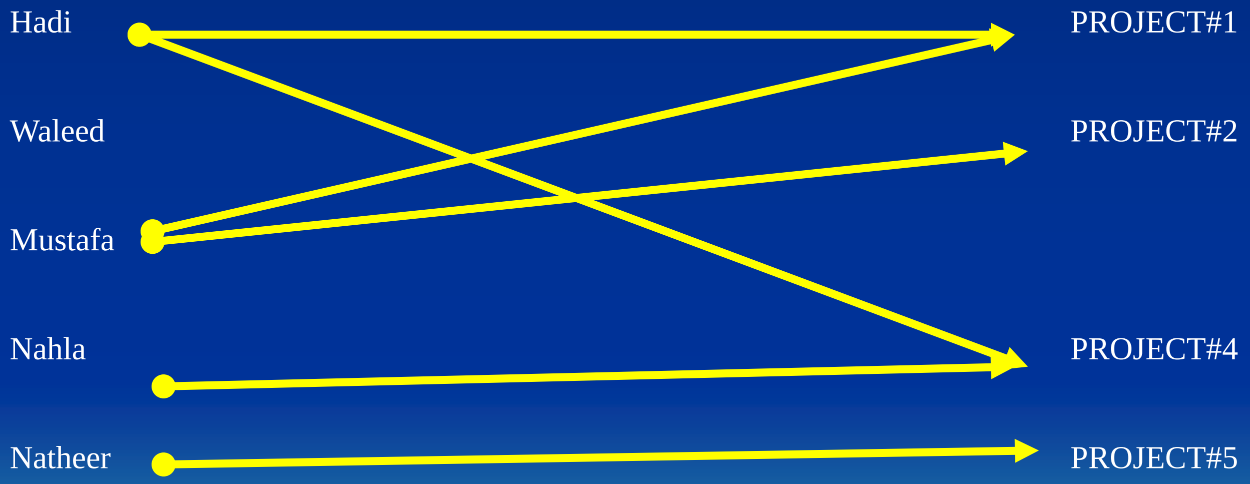
One optional , one mandatory Cardinalities..



EMPLOYEE

ASSIGNED_TO

PROJECT



Hadi

Waleed

Mustafa

Nahla

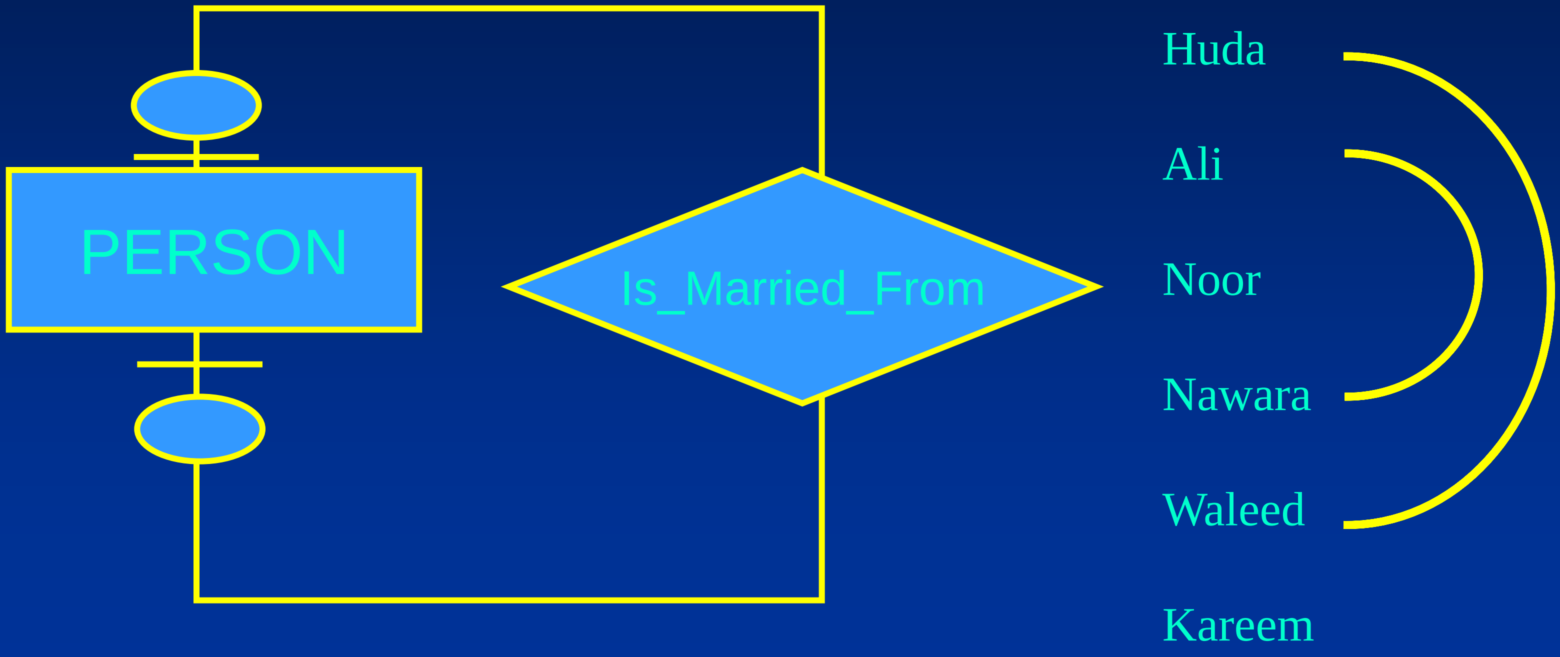
Natheer

PROJECT#1

PROJECT#2

PROJECT#4

PROJECT#5



PERSON

Is_Married_From

Huda

Ali

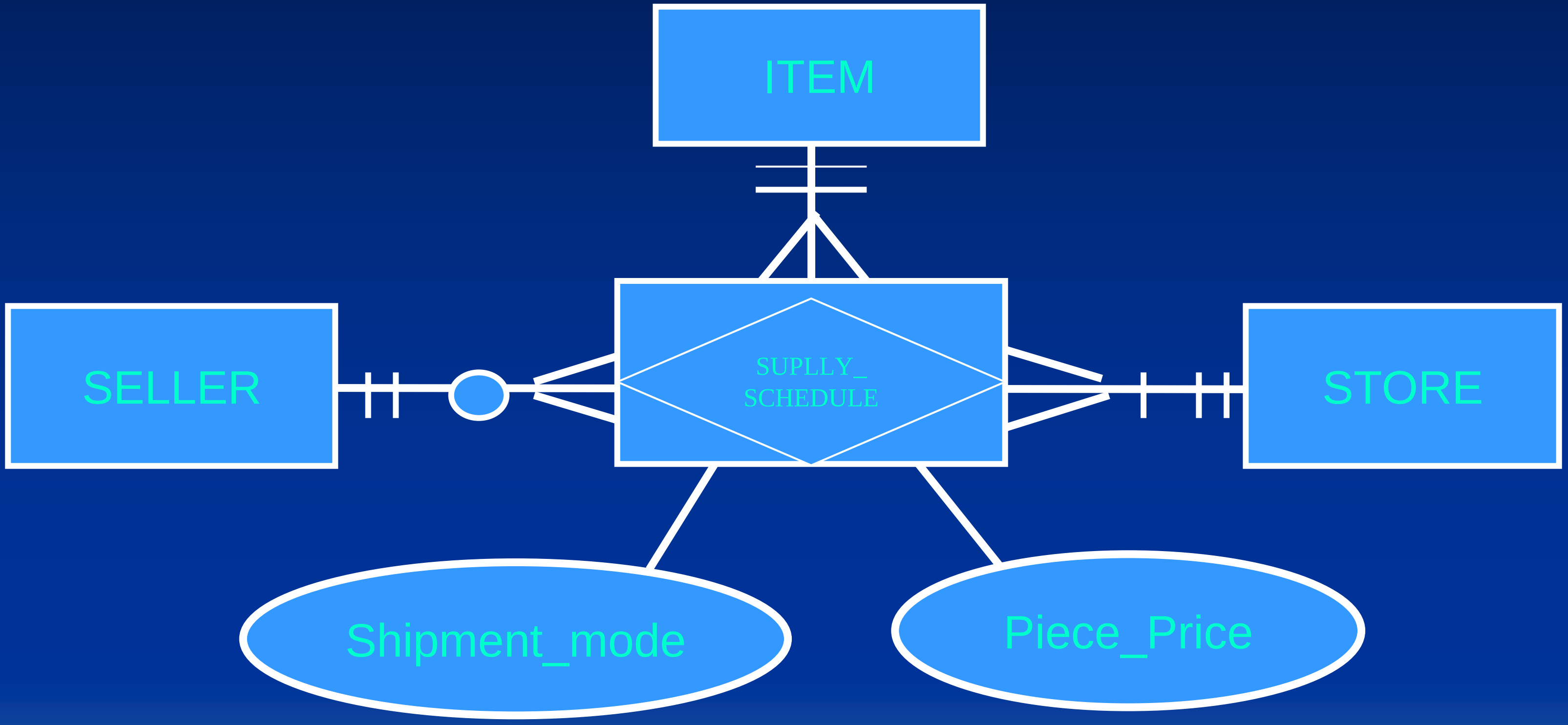
Noor

Nawara

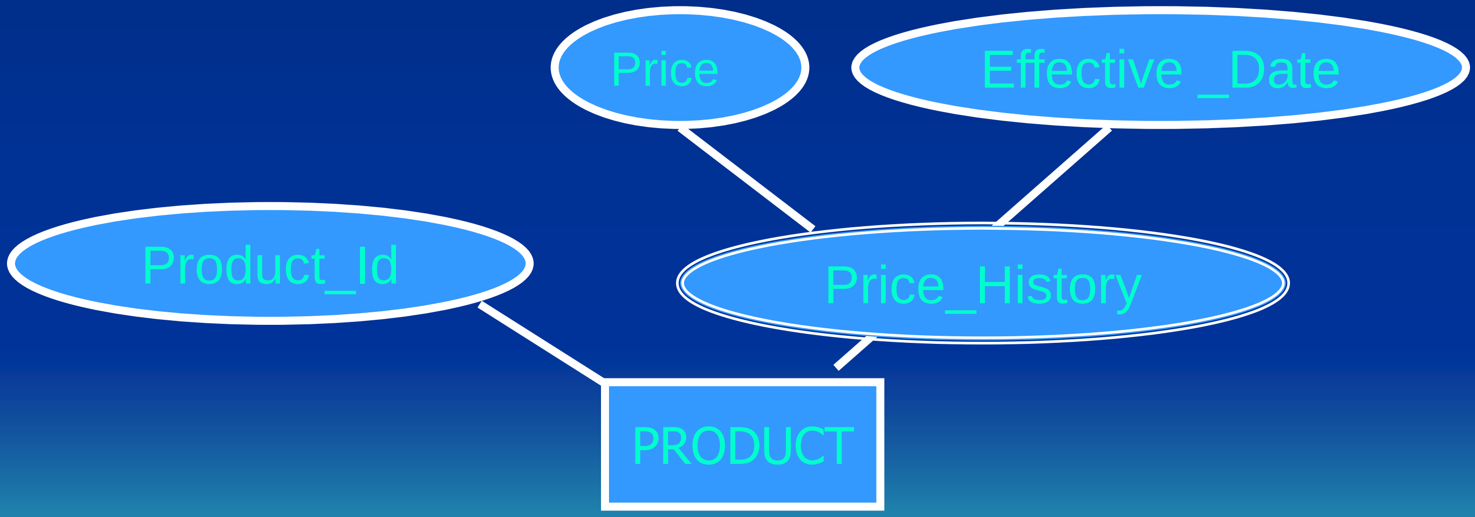
Waleed

Kareem

A ternary relationship (Add cardinality constraints to this giagram):

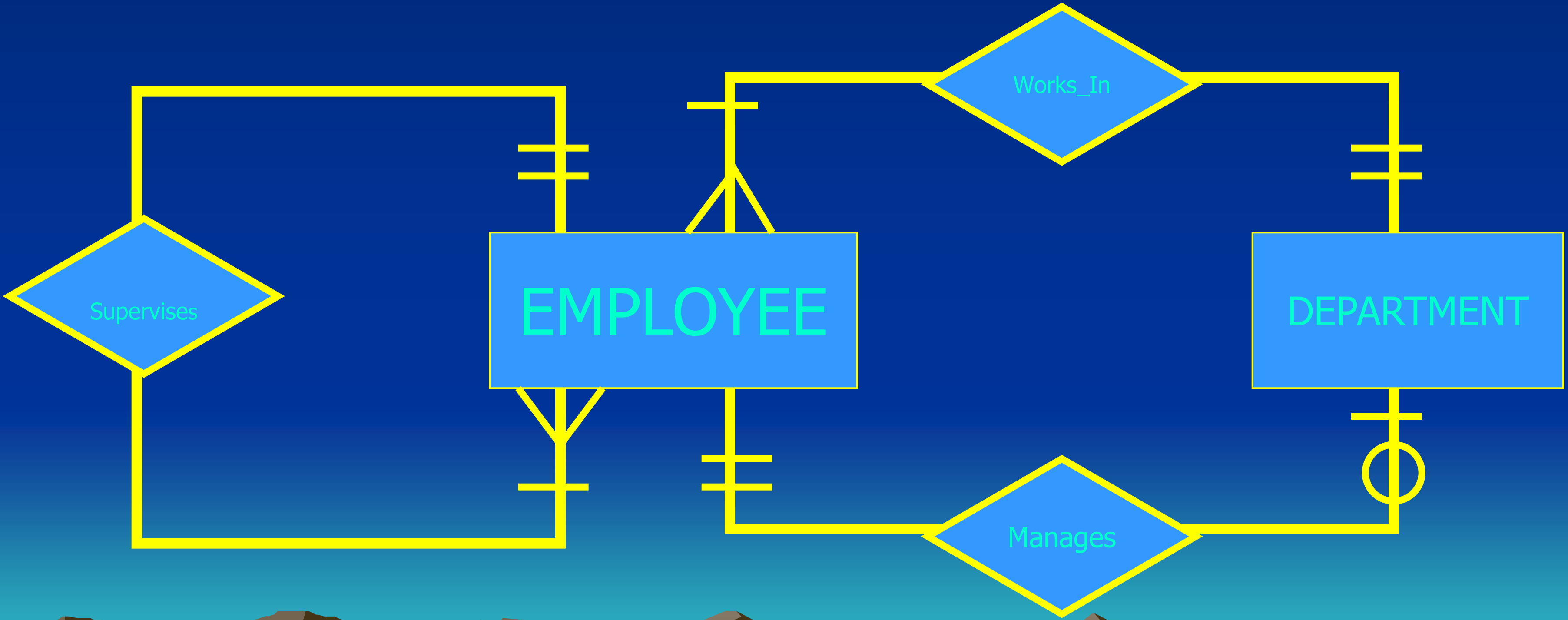


- Time Stamp :
- A time value that associate with a data value .
- A time stamp may be associated with any data value that changes over time when ew need to maintain a history of those data values.

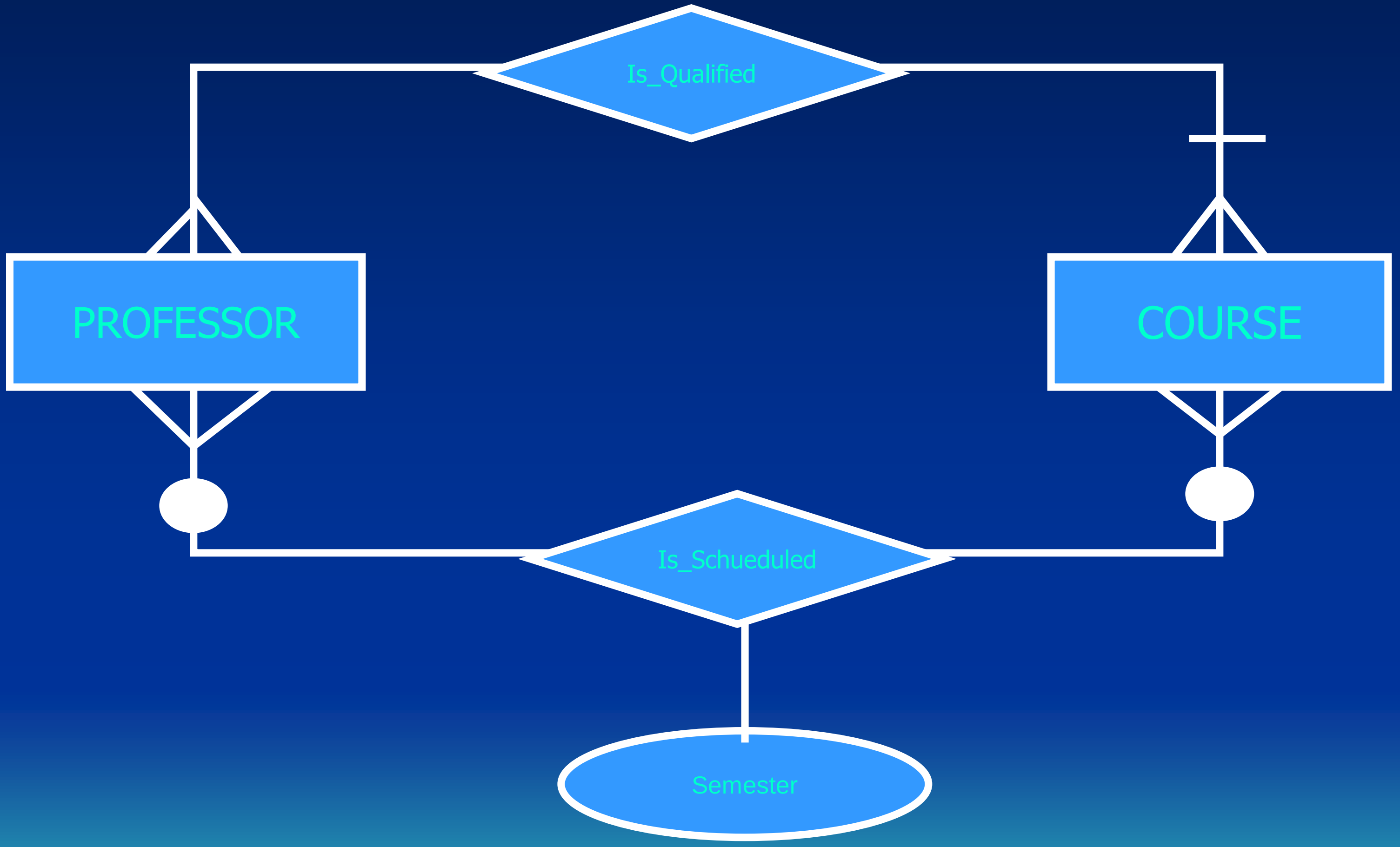


Multiple Relationships:

- An organization of more than one relationship between the same entity type.
- Examples:
- a. Employees and departments:



b. Professors and Courses:


Is_Qualified
PROFESSOR
COURSE
Is_Schueduled
Semester