CS 220
Database Systems

Fall 2019

Lecture 7
E-R Modelling

Conceptual Database Design

Conceptual database design involves modeling the collected information at a high-level of abstraction without using a particular data model or DBMS.

High-level modeling languages for conceptual design.

Unified Modeling Language (UML) syntax

Entity–Relationship (ER) Model

Entity–Relationship (ER) model, is a popular high-level conceptual data model.

Entity-relationship modeling is a top-down approach to database design that models the data as entities, attributes, and relationships.

The ER model conveys knowledge at a high-level (conceptual level) which is suitable for interaction with technical and non-technical users

E-R Model can later be converted into the desired logical model (e.g. relational model)

Entity – Relationship (ER) Model

The ER model defines entities and relationships by including
properties of entities called attributes
constraints on entities, relationships, and attributes

The diagrammatic notation associated with the ER model, known as **ER diagrams**

Entity

An entity is an object or concept of interest which are identified as having an independent existence.

Entity is a thing or object in the real world with an independent existence.

Entity does not always have to be a physical real-world object such as a person or department, it can be an abstract concept such as a project or job.

Entity Type is the collection of entities with similar characteristics

Entity

Entity is represented by rectangles with the name of the entity in the rectangle.

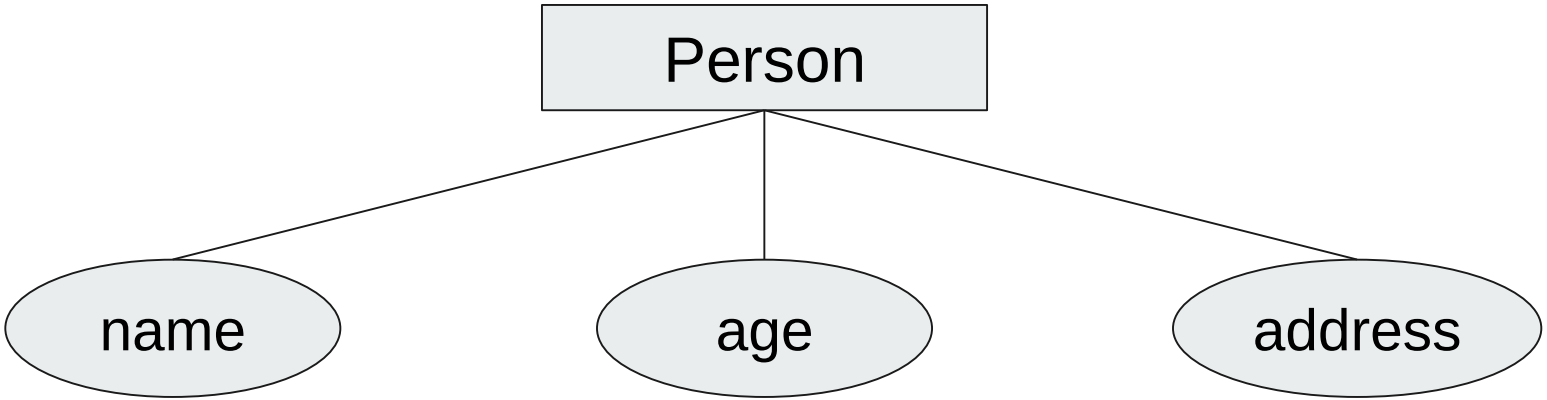
*Entity Name is normally a singular noun
**First Letter of each word in Entity name is capitalized

Attributes

Attributes are particular properties that describe an entity

Person entity may have attributes *name*, *age, address, salary*

Entity is represented by ellipses

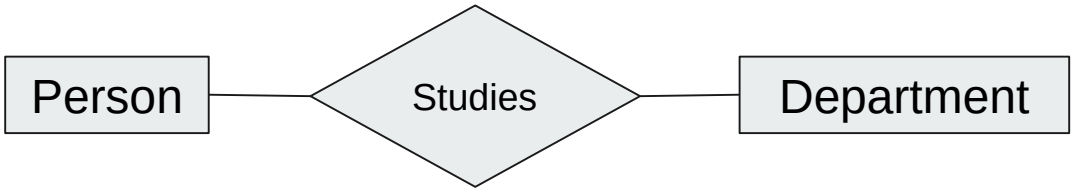


Relationship

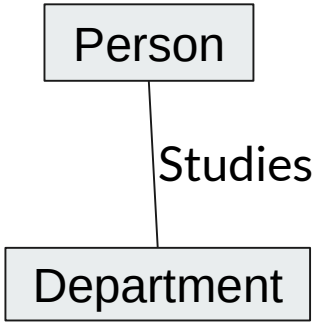
Relationship shows associations among entities.
Each relationship has a name that describes its function.

There can be more than one relationship between two entity types

- Rhombus is used to set up relationships between two or more entities
- Sometimes represented as a labeled edge between the two entities
  - The label is applied only in one direction so an arrow indicates the correct way to read it.



OR



Relationship Type is a collection of similar relationships

Entity Name

Entity

Person,place,object,event or concept about which data is to be maintained

Example: Car, Student

Jack

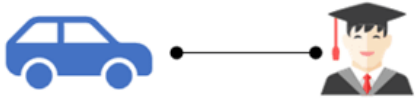
Attribute Name

Attribute

Property or characteristic of an entity

Example: Color of car Entity

Name of Student Entity

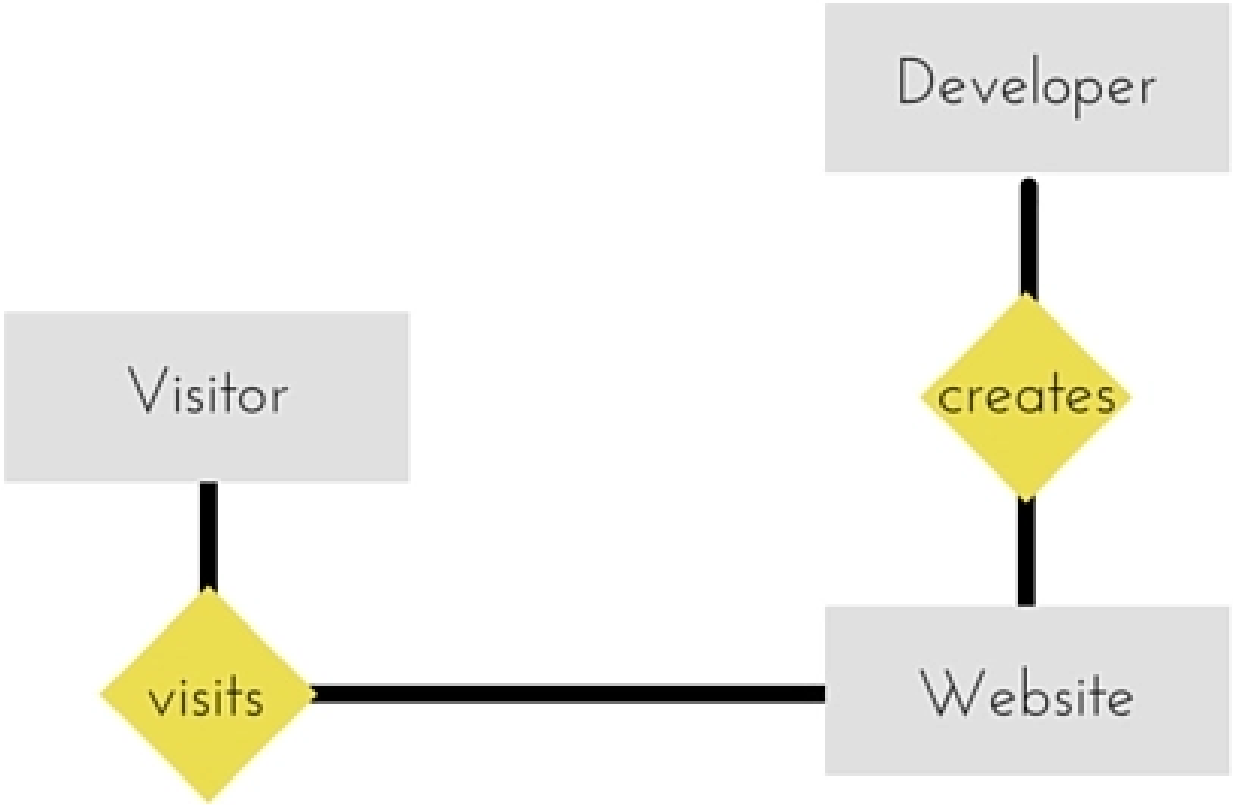


Relation

Verb Phrase

Association between the instances of one or more entity types

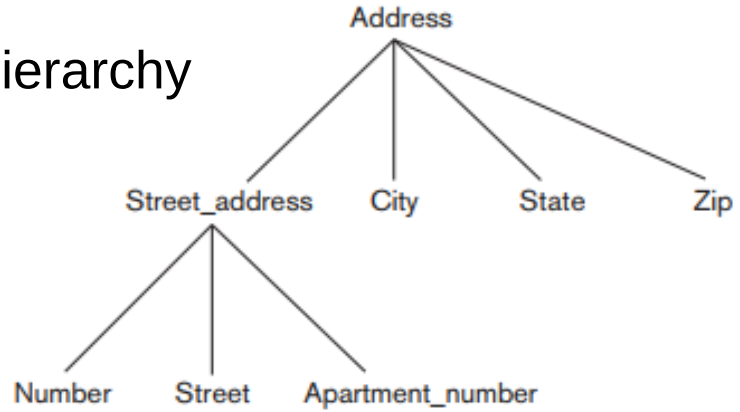
Example: Blue Car Belongs to Student Jack


Developer
creates
Visitor
visits
Website

More on Attributes

Simple (Atomic) Attributes: Values are atomic

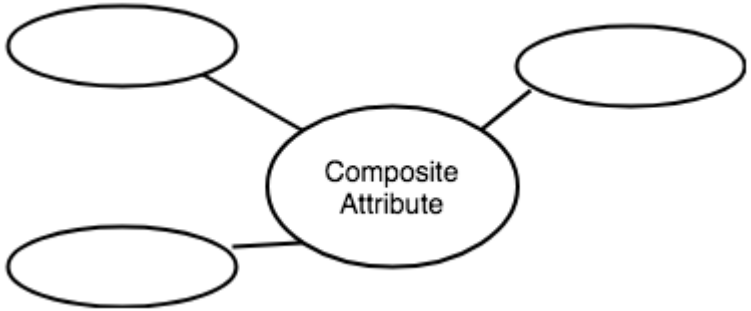
Composite Attributes: Attributes that form a hierarchy



Composite attributes are useful to model situations in which a user sometimes refers to the composite attribute as a unit but at other times refers specifically to its components

More on Attributes

Composite Attributes:



If the composite attribute is referenced only as a whole, there is no need to subdivide it into component attributes.

If there is no need to refer to the individual components of an address (Zip Code, street, and so on), then the whole address can be designated as a simple attribute.

More on Attributes

Single Valued Attributes: Single Values for a particular entity

e.g. Age, Color of cars

Multivalued Attributes: Multiple values for attributes

Colors attribute for a car

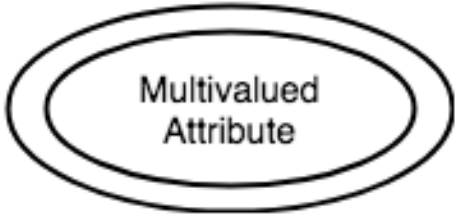
What if car has two or more colors?

College_degrees attribute for a person

Some individuals may have 1 college degree,

Others may have 2 or more degrees

Represented using a **Double Ellipse**



More on Attributes

Multivalued Attributes:

A multivalued attribute may have lower and upper bounds to constrain the number of values allowed for each individual entity.

For example, the Colors attribute of a car may be restricted to have between one and two values, if we assume that a car can have two colors at most

More on Attributes

Stored Attributes:

Value of attributes exists independently
e.g. Date_of_Birth

Derived Attribute:

When value of an attribute is derived from another attribute
e.g. Age is derived from Date_of_Birth

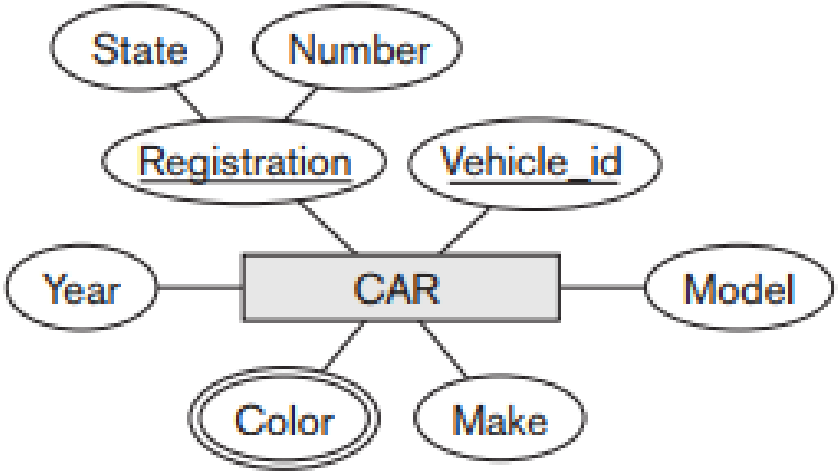
Represented using **dotted ellipse** is created inside the main ellipse

More on Attributes

Key Attributes:

Values are distinct for each individual entity

Represented by underlining the name



Creating an ER-Model

Information:

The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

A department controls a number of projects, each of which has a unique name, a unique number, and a single location.

Creating an ER-Model

Identifying Entities:

The company is organized into **departments**. Each department has a unique name, a unique number, and a particular **employee** who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations.

A department controls a number of **projects**, each of which has a unique name, a unique number, and a single location.

Creating an ER-Model

More Information:

The database will store each employee's name, Social Security number,2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. It is required to keep track of the current number of hours per week that an employee works on each project, as well as the direct supervisor of each employee (who is another employee).

The database will keep track of the dependents of each employee for insurance purposes, including each dependent's first name, sex, birth date, and relationship to the employee

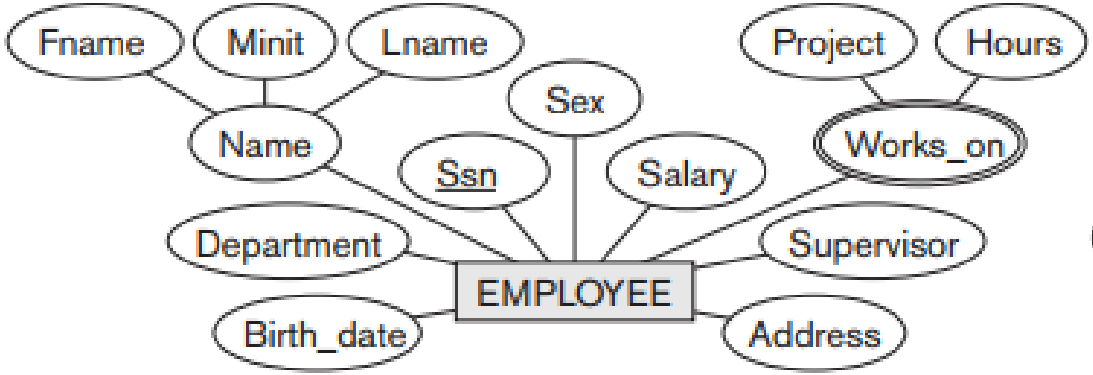
Creating an ER-Model

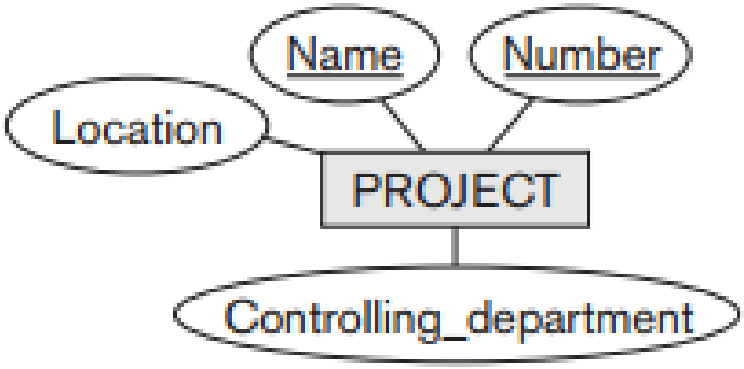
Identifying Entities:

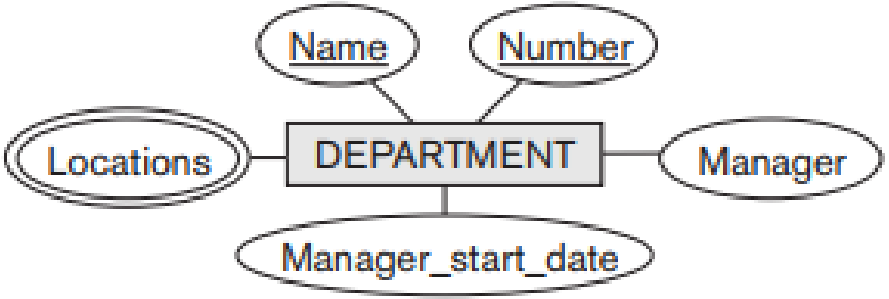
The database will store each employee's name, Social Security number,2 address, salary, sex (gender), and birth date. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. It is required to keep track of the current number of hours per week that an employee works on each project, as well as the direct supervisor of each employee (who is another employee).

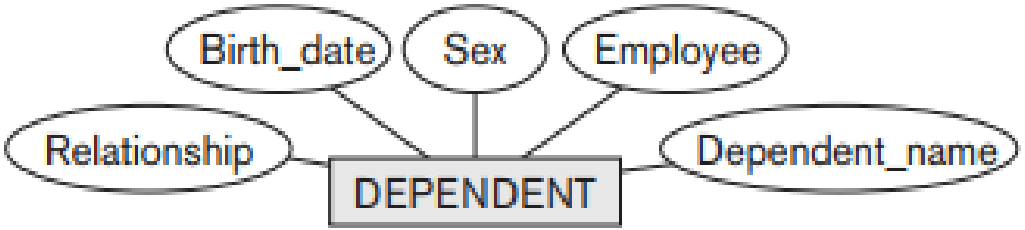
The database will keep track of the **dependents** of each employee for insurance purposes, including each dependent's first name, sex, birth date, and relationship to the employee

Entities & Attributes Identified









More on Relationships

Degree of Relationship:

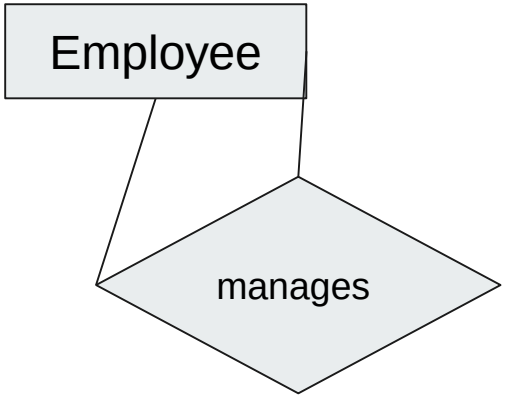
Number of Entity Types participating in a relationship

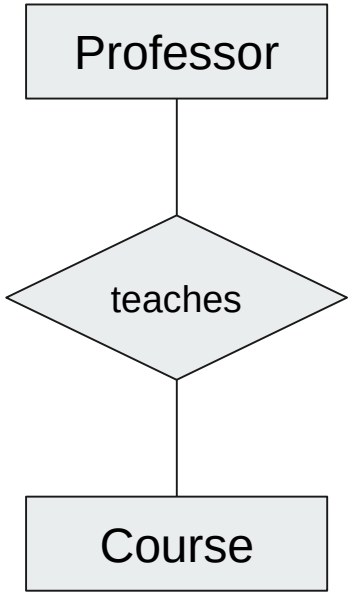
Unary Relationship: Relationships of degree one are unary
e.g. Employee manages Employee
Binary Relationship: Relationships of degree two are binary
E.g. a PROFESSOR teaches one or more CLASSes
Ternary Relationship: Relationships of degree three are ternary
E.g booking by Travel Agency for a Tourist in a Hotel

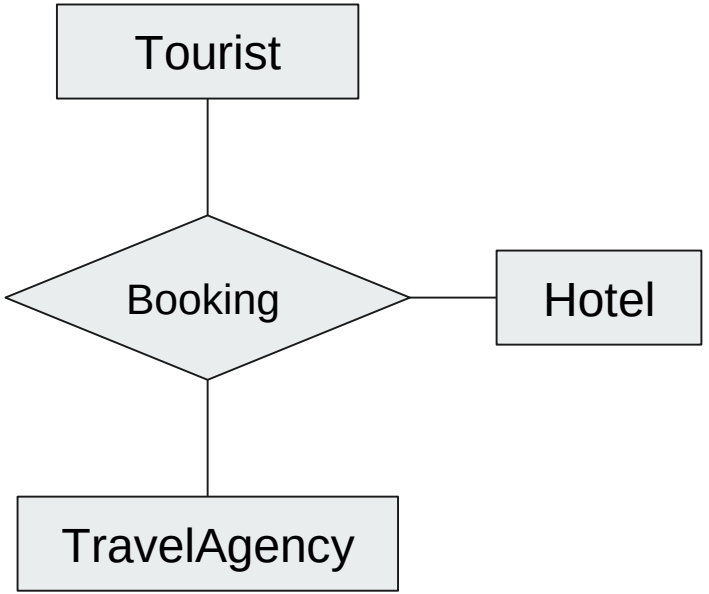
*Relationships of arbitrary degree N are called n-ary

More on Relationships

Degree of Relationship:







Unary

Binary

More on Relationships

Recursive Relationship:

When an Entity is related with itself it is known as Recursive Relationship

Naturally, such a condition is found within a unary relationship

More on Relationships

Cardinalities

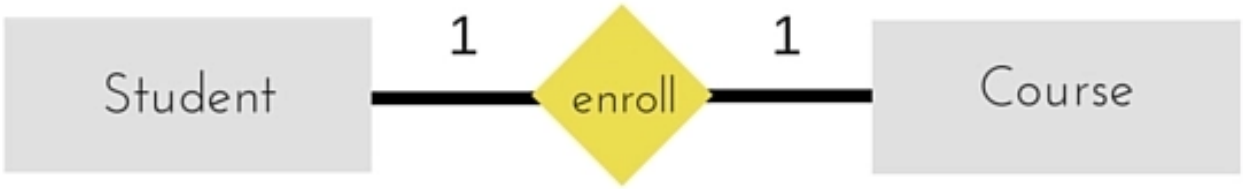
Every relationship type can be characterized in terms of its cardinalities, which specify the minimum or maximum number of relationship instances that an individual entity can participate in

4 type of Cardinalities

One to One
One to Many
Many to One
Many to Many

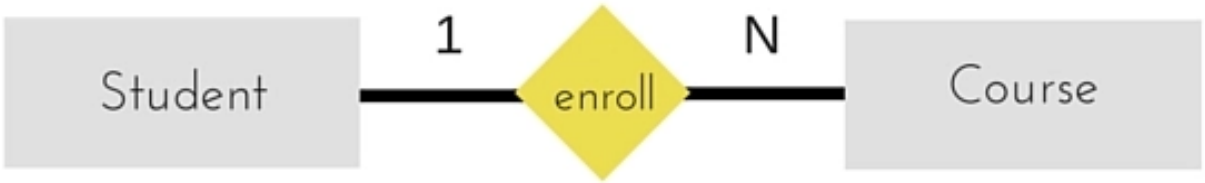
More on Relationships

Cardinalities - One to One



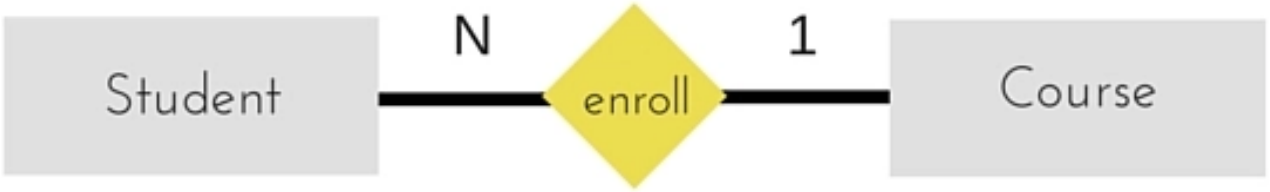
More on Relationships

Cardinalities - One to Many



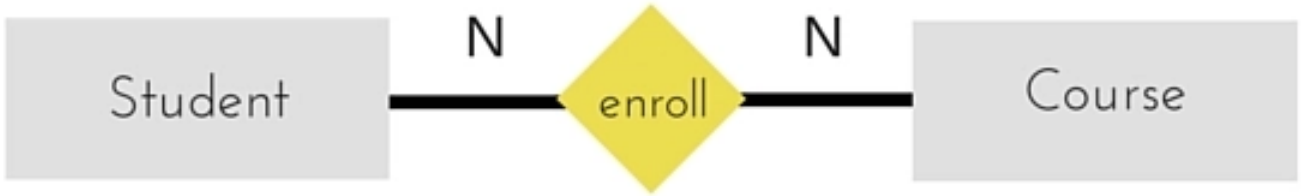
More on Relationships

Cardinalities - Many to One

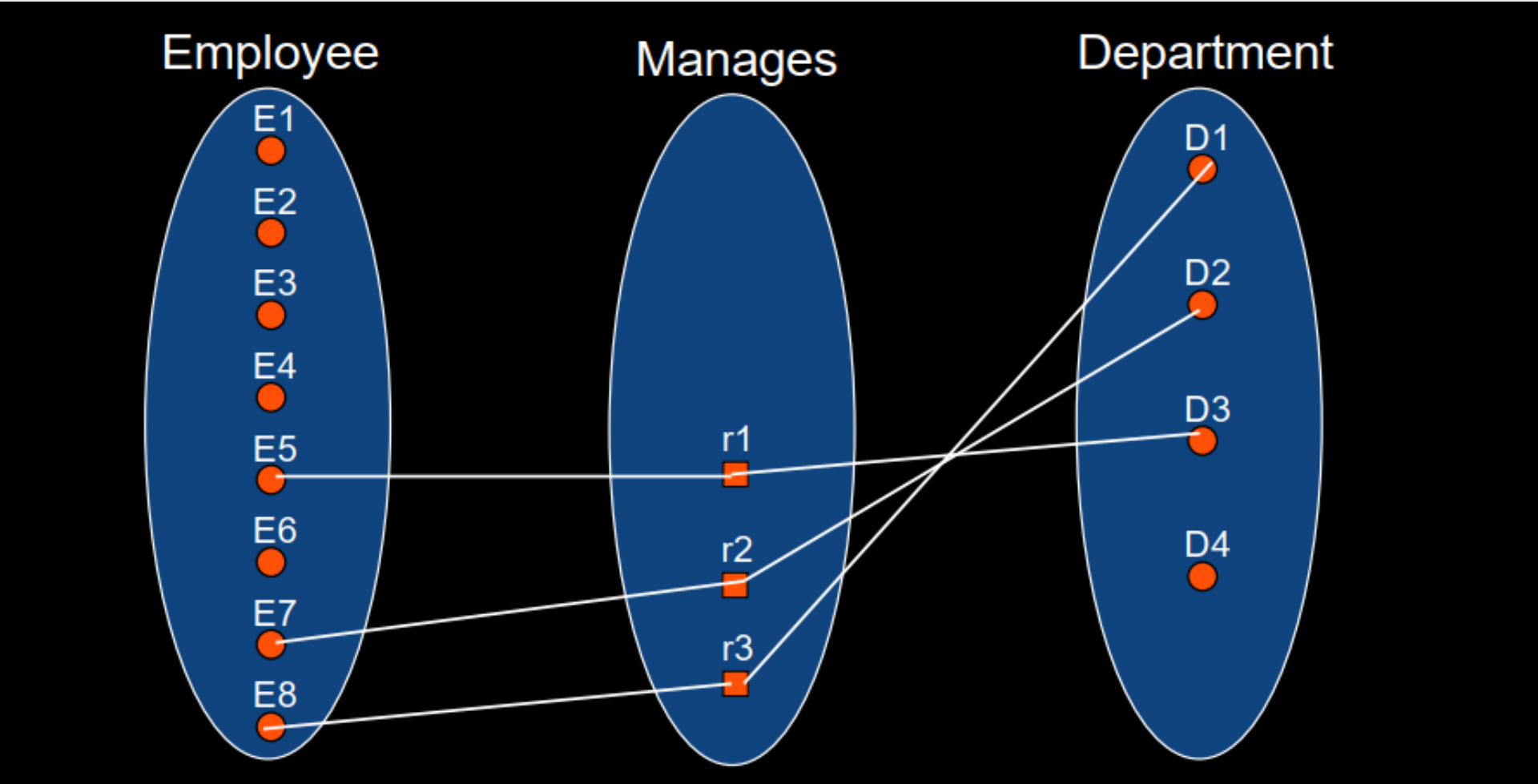


More on Relationships

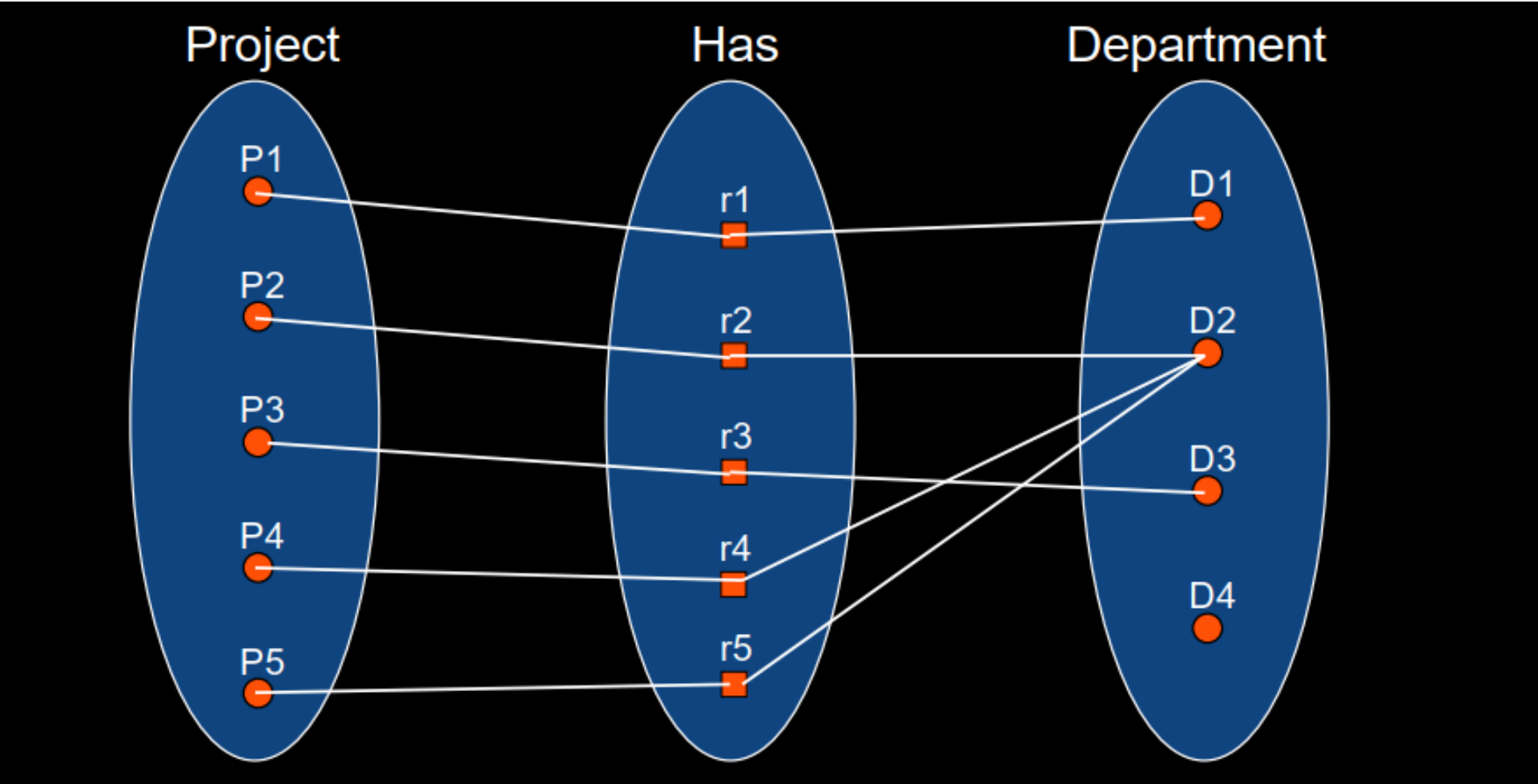
Cardinalities - Many to Many



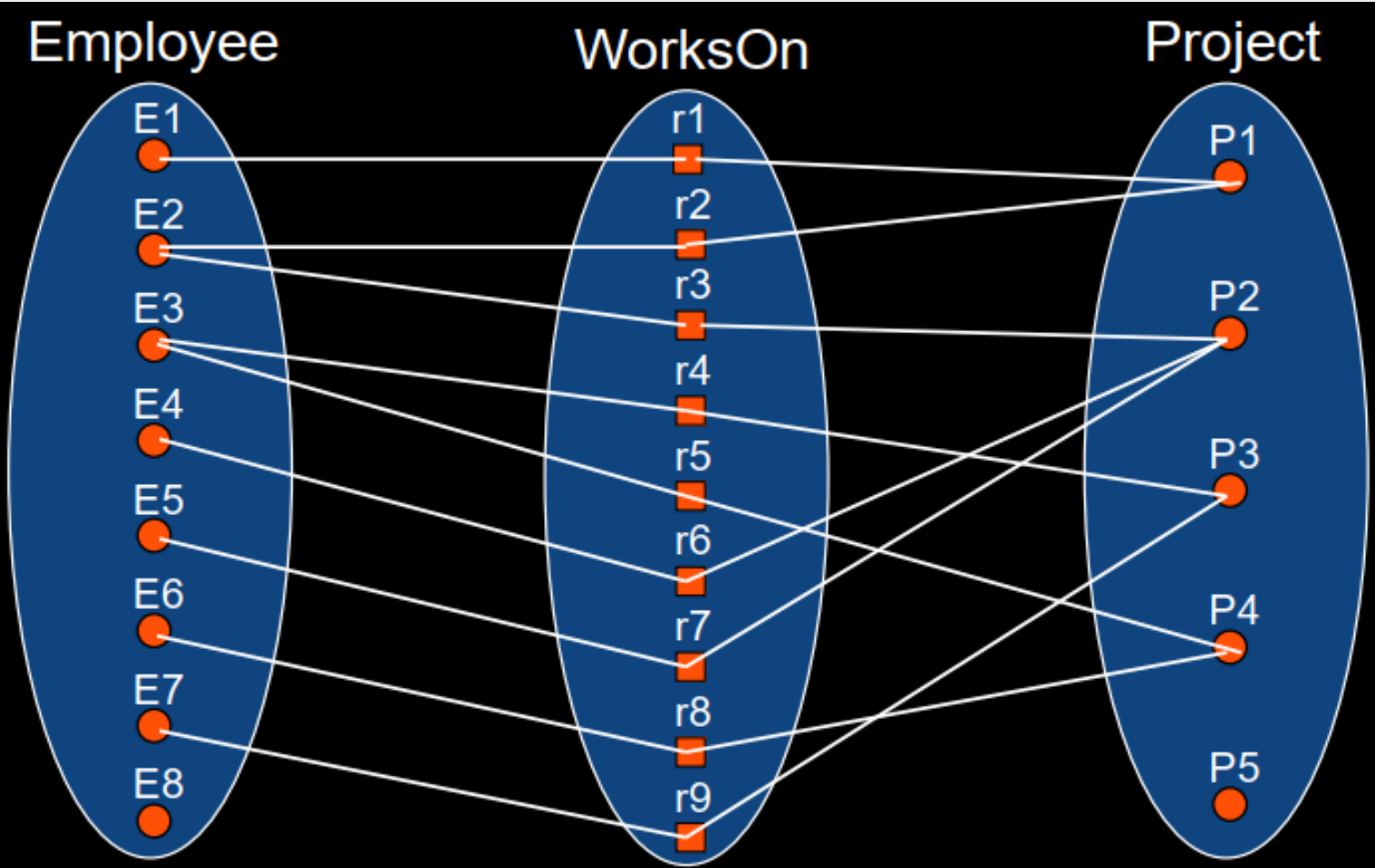
Visualizing Relationships - 1:1



Visualizing Relationships - 1:N



Visualizing Relationships - N:N



Attributes on Relationship

An attribute may be associated with a relationship type

For example, the WorksOn relationship type has two attributes: Project and Hours.

Note that these two attributes belong to the relationship and cannot belong to either of the two entities individually (as they would not exist without the relationship).

Relationship attributes are represented as a separate box connected to the relationship using a dotted line.

Defining Relationship Types for Company Database

EMPLOYEE and DEPARTMENT

MANAGES one to one relationship

attribute **Start_date** is assigned to this relationship type

WORKS_FOR one to many relationship

Employee and Employee?

Defining Relationship Types for Company Database

Employee and Employee?

Employee and Dependent?