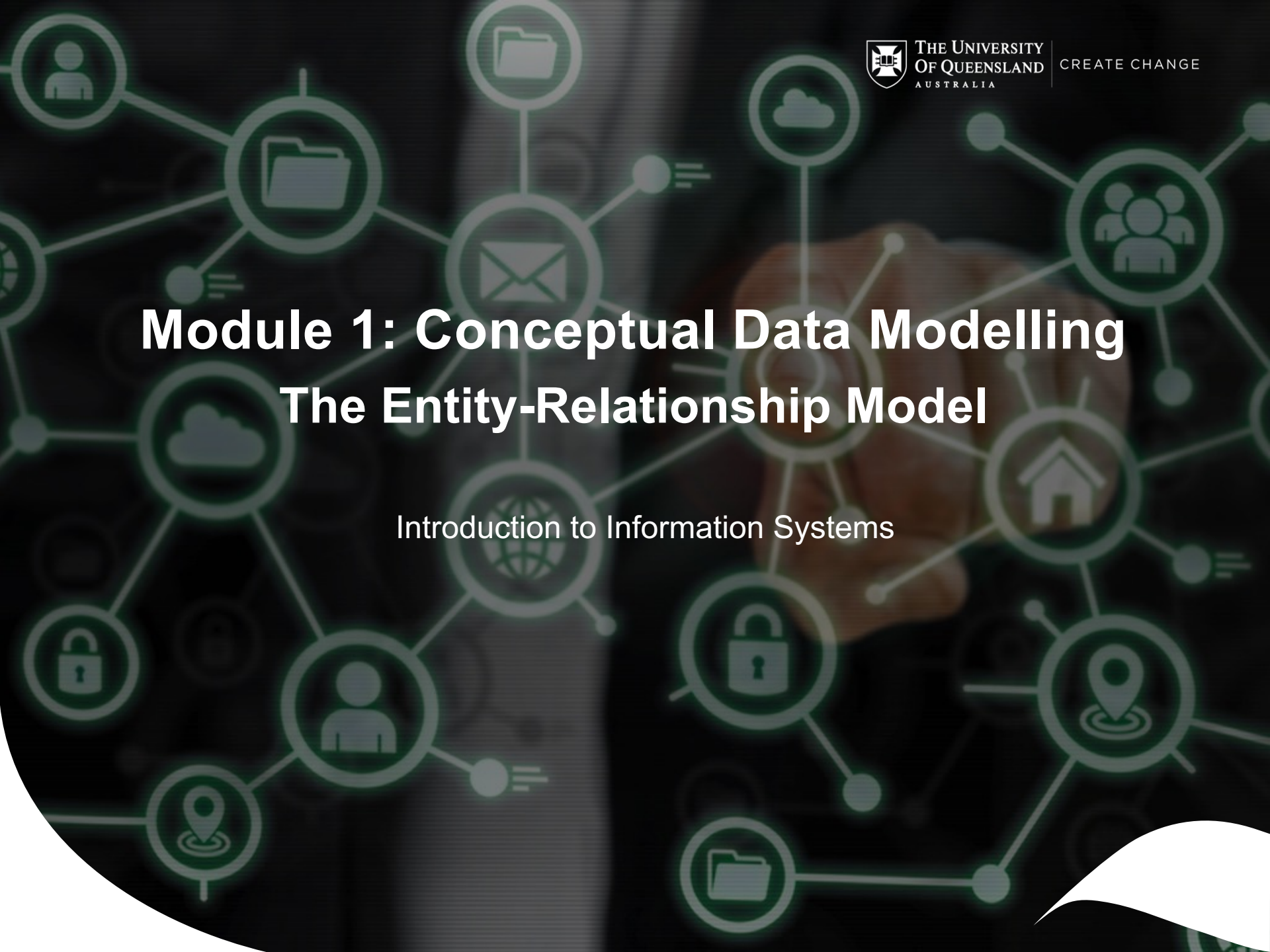


The University Of Queensland AUSTRALIA | CREATE CHANGE

Module 1: Conceptual Data Modelling

The Entity-Relationship Model

Introduction to Information Systems

In This Module

- Why is conceptual modelling important for designing a successful database application?
- How can we use the Entity Relationship (ER) Model for conceptual modelling?
- How can we evaluate the trade-offs between design choices by comparing alternative ER diagrams for the same system?

Learning Outcomes

Analyse, extract and structure information system requirements to create basic conceptual models.

- Define the concepts of entities, relationships, cardinality constraints, participation constraints, weak entities, superclasses and subclasses in the Entity Relationship (ER) model.
- Given the description of a small operating environment, create an ER model of the environment.
- Identify alternative representations for modelling an operating environment and evaluate the choices.

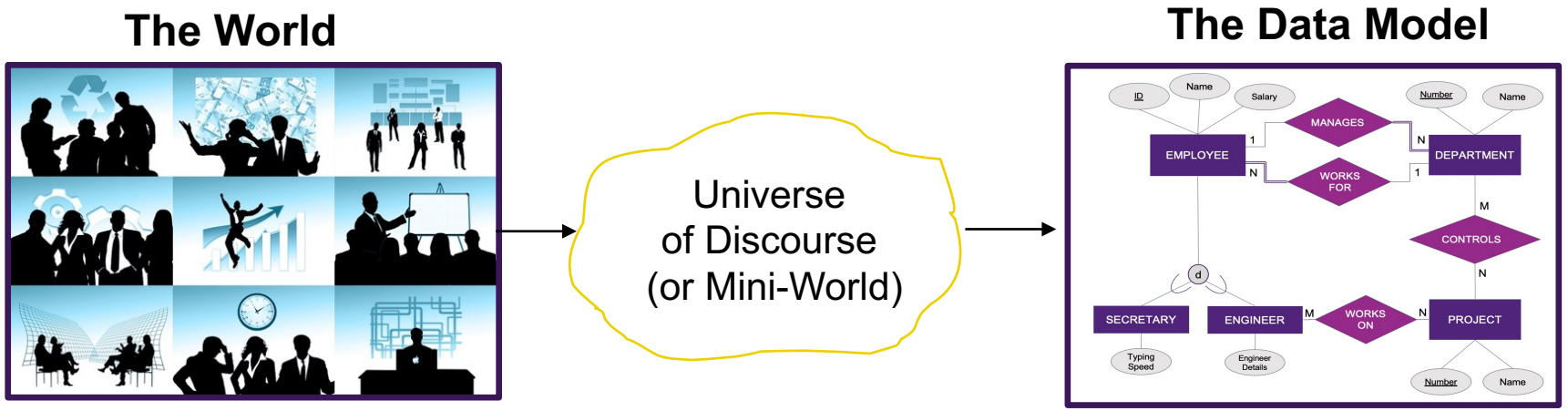
Conceptual Database Design

- Entities and Relationships
- Relationship Constraints
- The EER Diagram
- Design Choices

Conceptual Database Design

Conceptual database design is a very important phase in designing a successful database application.

- Step 1: Identify the “Universe of Discourse” (UoD). The database to be built will not model everything in the world, but rather some “mini-world” or “Universe of Discourse”.
- Step 2: Convert the UoD to a data model, which can be captured by a database.

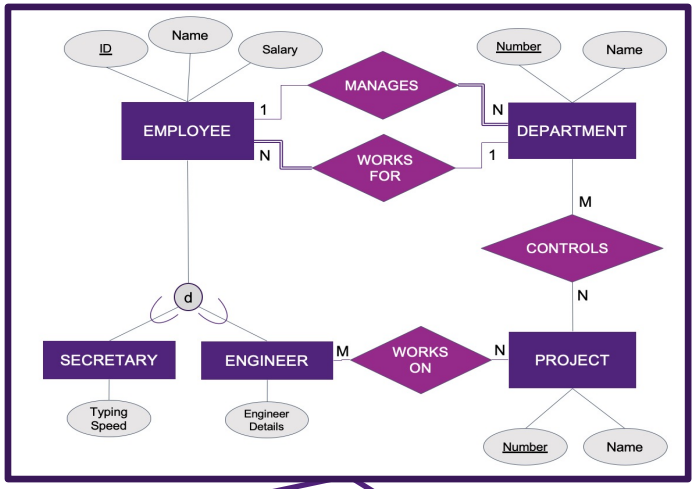


Conceptual Database Design

A Model is never perfect

The World

The Conceptual Schema



Some employees go out for dinner on Fridays

Phenomena not Captured in the Model

Every employee works for a department

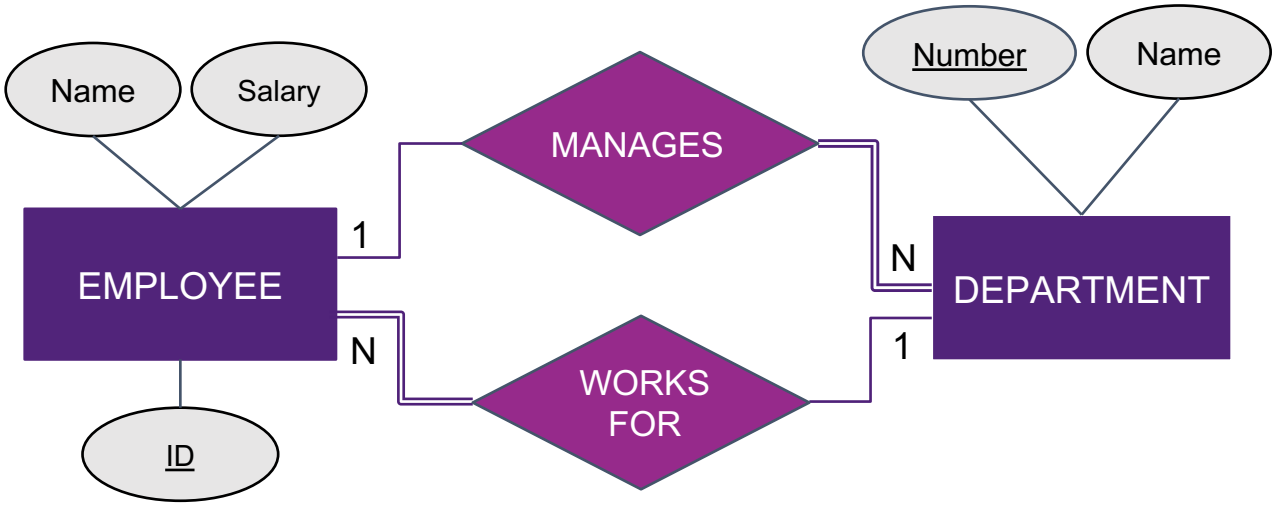
Common Phenomena

Every secretary can type

Phenomena not True in the World

The Entity-Relationship (ER) Model

The Entity-Relationship **(ER) model** at its core provides a **graphical representation** of **data entities.**

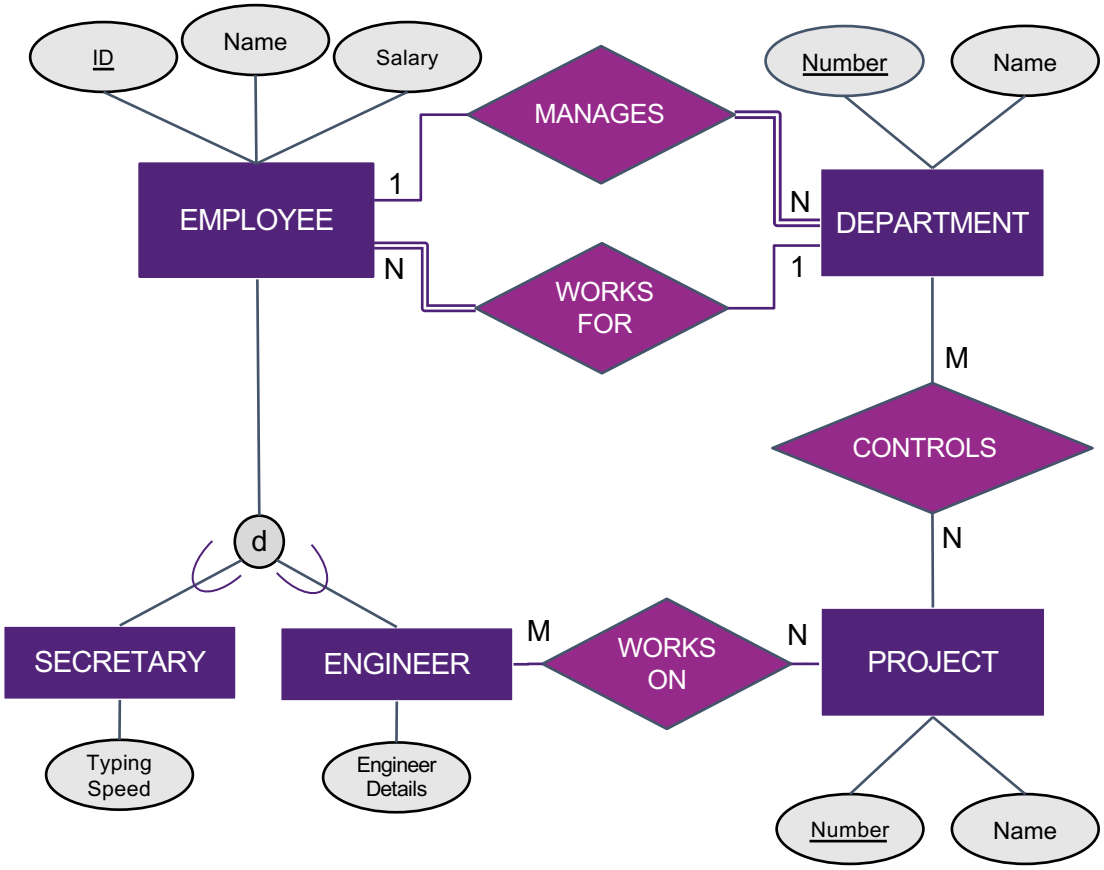


ER Models assist database designers to define a project's scope and requirements for clients and businesses.

An ER Model Example

Example Company ER Diagram

- Constraints in the ER model
- Gauge how the system works, how various data within the system is connected
- The ER diagram can also be extended to allow for generalisations and specialisations of data to be stored



Database Design Overview

Entities and Relationships

Relationship Constraints

The EER Diagram

Design Choices

Entities and Relationships

Entities

- Entities, Entity Sets and Entity Types
- Attributes, Keys and Value Sets
- Weak Entities

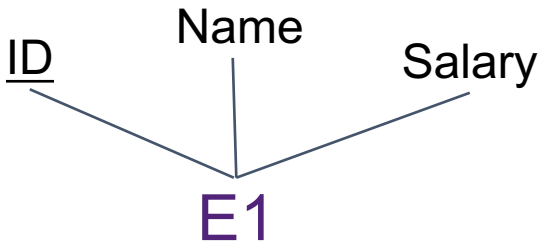
Relationships

- Relationship Types and Sets
- Relationship Degree
- Roles and Recursive Relationships
- Relationship Constraints
- Attributes of Relationship Types

Entity

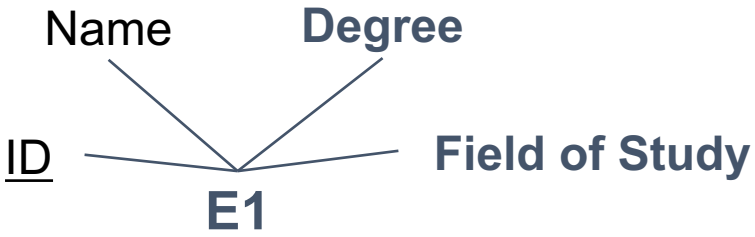
An entity is essentially a **physical or conceptual object which has data associated with it.**

An entity may have **various attributes associated with it.**

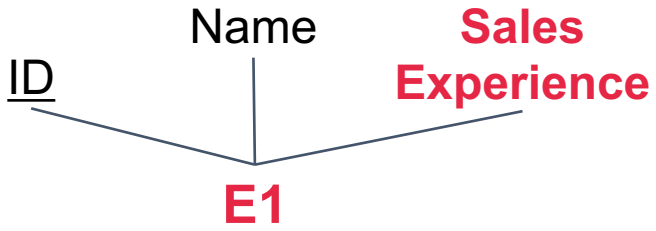


The **same entity can have different attributes** depending on the specific requirements of that system.

R&D Organisation



Retail Organisation

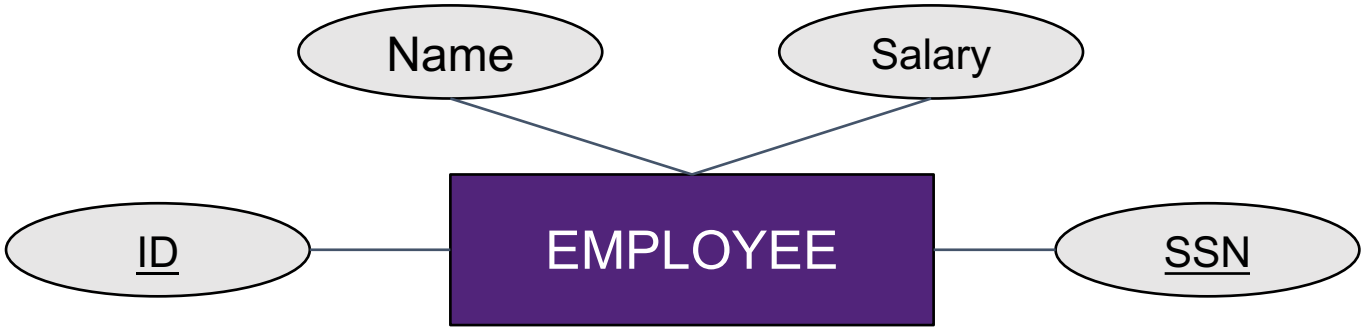


Entity Type

An entity type **provides a format for the data** which needs to be recorded to represent a particular entity. The **entity type** is described by its name and attributes.

In the ER model:

- An entity type is represented as a rectangular box.
- Attribute names are represented by ovals attached to their entity type.



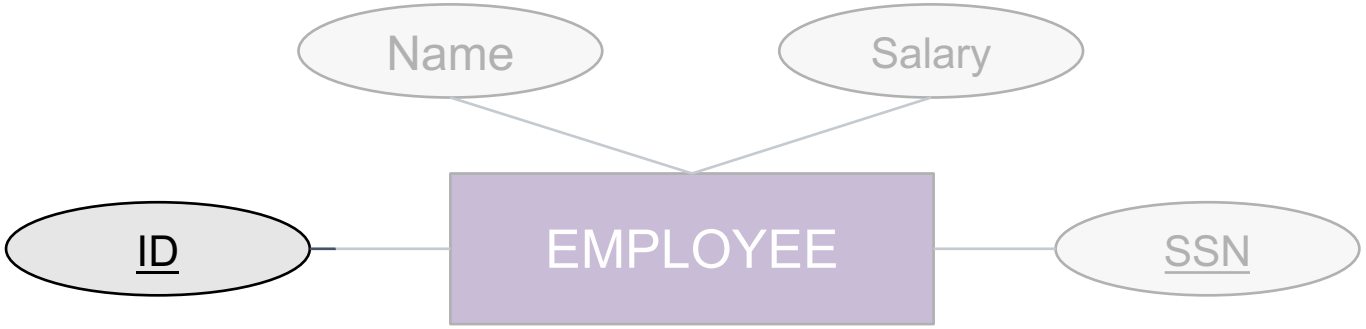
Entity Type – Key Attributes

All **entity types** have at least one **key** (or uniqueness) constraint.

- The value of a key attribute are **distinct (unique) for each individual entity** in the entity set.

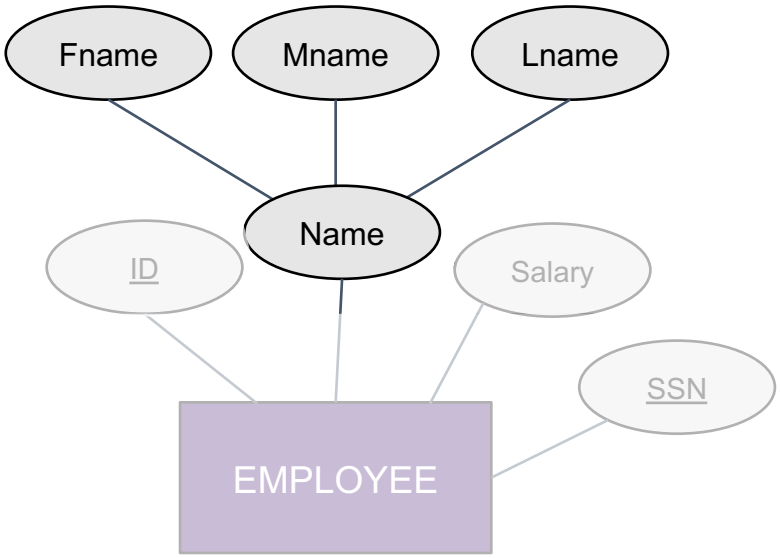
In the ER model:

- A key attribute has its name underlined inside the oval
- Key must hold for **every** possible extension of the entity type (see super/subclasses in later slides)
- Multiple keys are possible



Composite vs. Simple Attributes

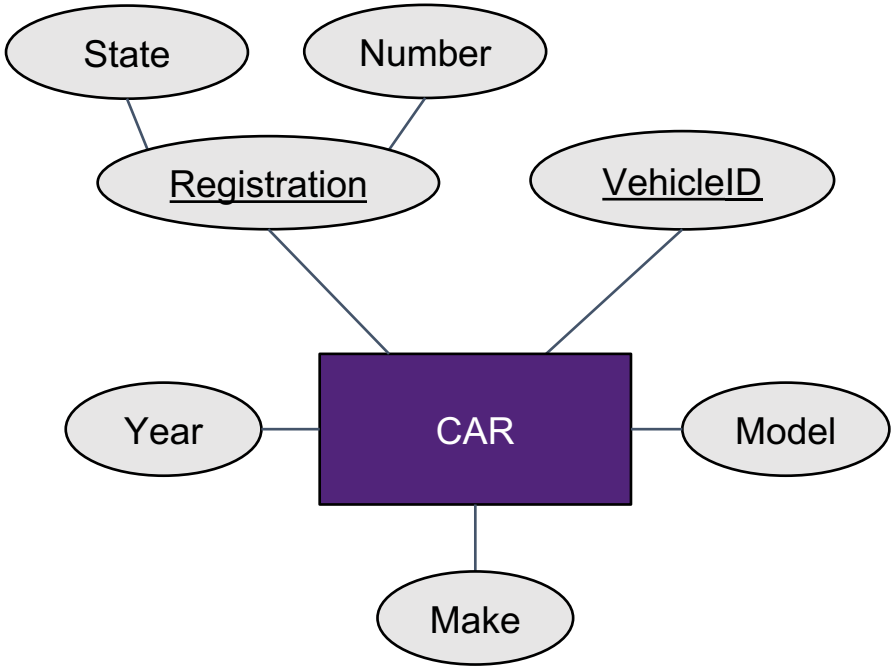
Composite attributes can be divided into smaller parts which represent simple attributes with independent meaning.



Several Attribute Keys

A composite key attribute is also a unique identifier for each entity instance.

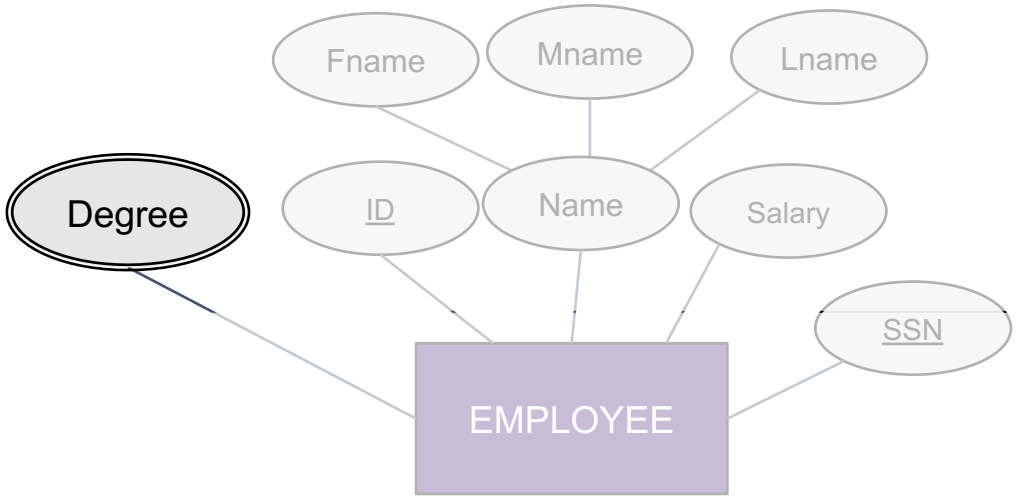
A composite key is **combination** of simple attributes which make up the composite key that must be unique.



Single vs. Multivalued Attributes

Simple attributes can either be single-valued or multi-valued:

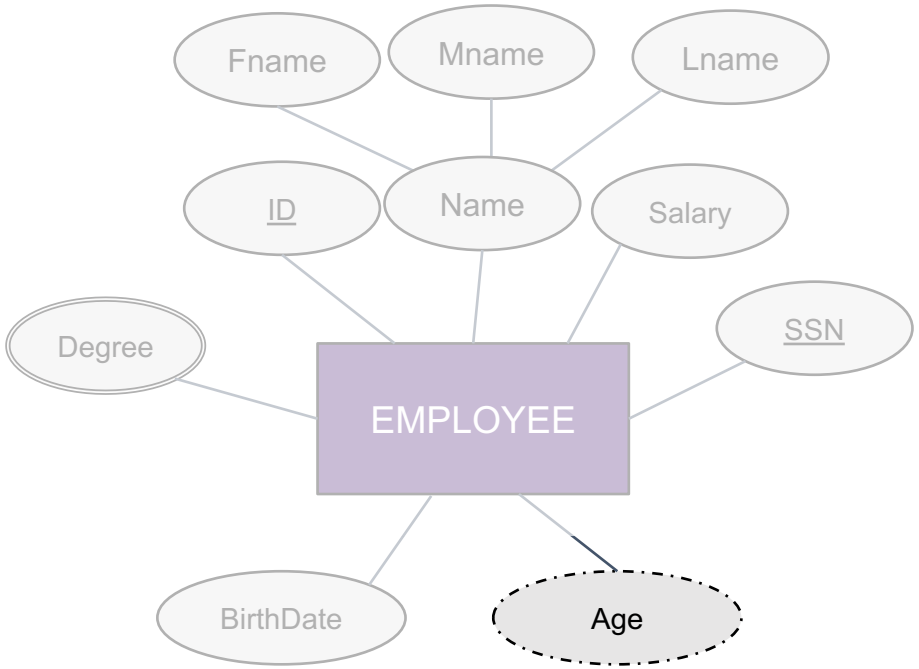
- Single-valued attributes can only have one value
- Multivalued attributes are shown with double-lined ovals and can have multiple values (e.g., one person can hold multiple degrees)



Stored Vs. Derived Attributes

In some cases, attribute values can be derived from related attribute values (e.g., age can be derived from BirthDate).

In the ER model, derived attributes are shown with an oval with a dotted line.



Value Sets of Attributes

Value sets specify the set of values that may be assigned to a particular attribute of an entity

- **employeeAge**: integers between 21 & 65
- **vehicleRegistrationNum**: string of 3 alphabets followed by 3 integers

Value sets are not displayed on the ER diagram

A particular entity may not have an applicable value for an attribute.

A **null** value may be used for representing this.

- **tertiaryDegree**: not applicable for a person with no university education
- **homePhone**: not known if it exists
- **height**: missing

In-class Exercise

Every student has a unique id, name (composed of title, first name, middle initial and last name), email, address (composed of number, street name, suburb and postcode), and one or more phone numbers. Draw an ER diagram for the above description.

Entity Set

The collection of all entities of a particular entity type in the database at any point in time is referred to as an **entity set**.

An entity set can be mapped to a table.

Employee

ID	Name	Salary	Department
175	Paris Lane	60,000	2
467	Anna Lee	70,000	1
1023	Ben Cho	70,000	4
2034	Jack Smith	40,000	1
2670	Grace Mills	50,000	2

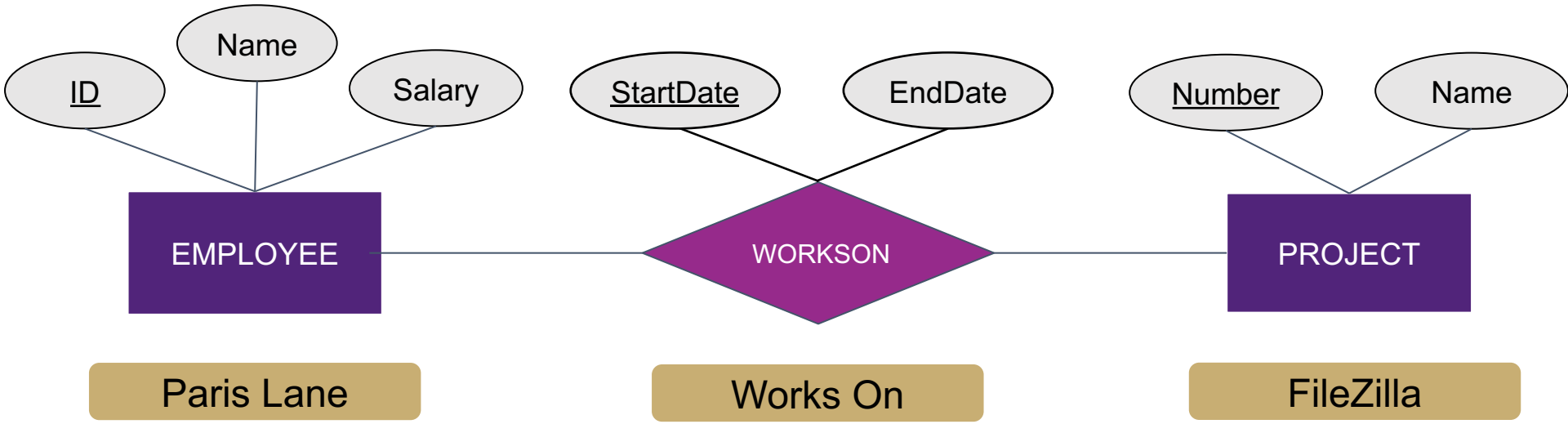
Relationship Types

A relationship is an association among two or more entities

- e.g., Paris Lane works on the project FileZilla

A relationship type defines the relationship.

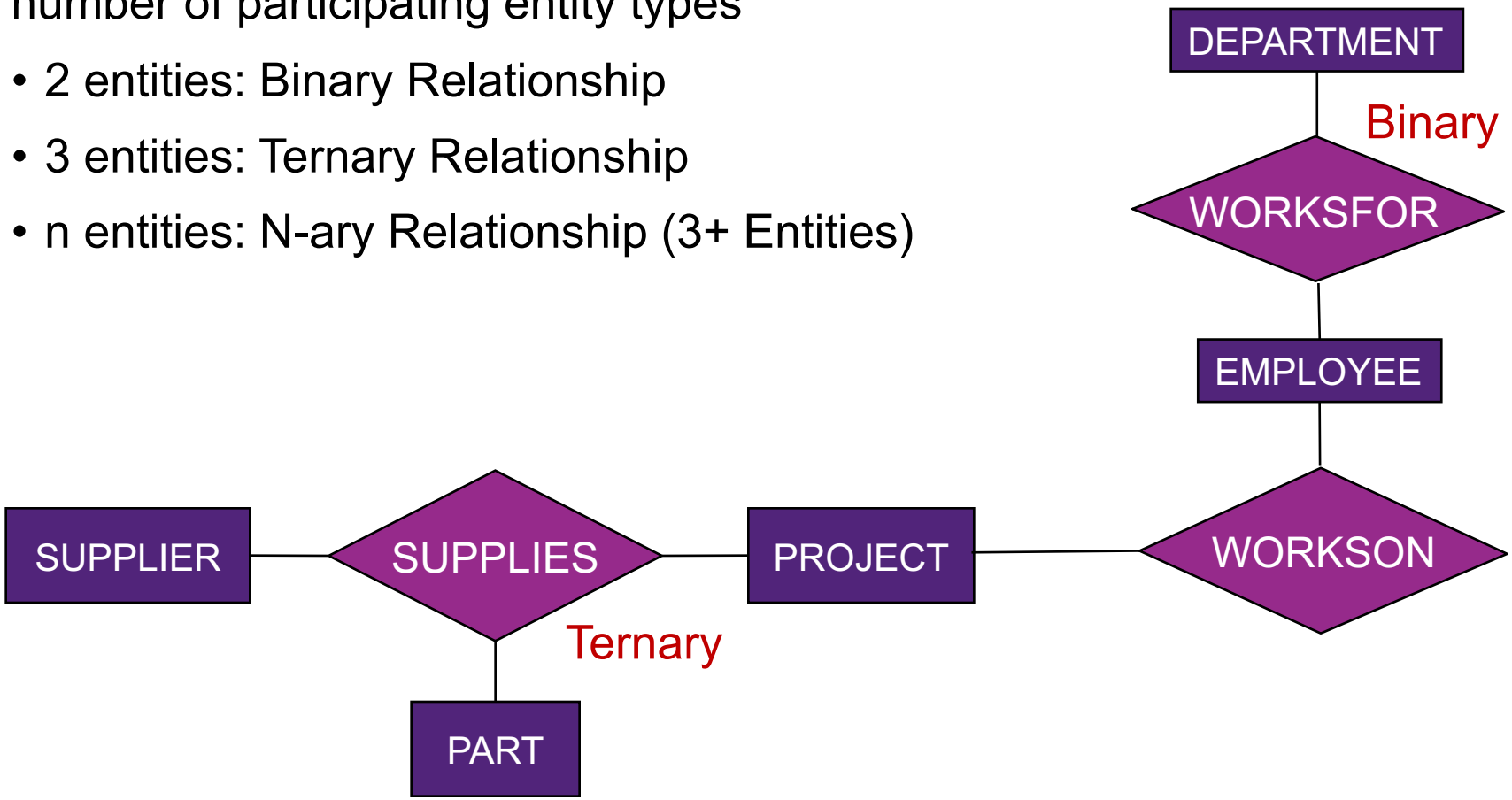
- In the ER model, relationships are represented using a diamond that is connected to the associated entity types.
- A relationship type may have ***descriptive attributes.***
- A relationship type may have key attributes



Relationship Degree

The degree of a relationship type is the number of participating entity types

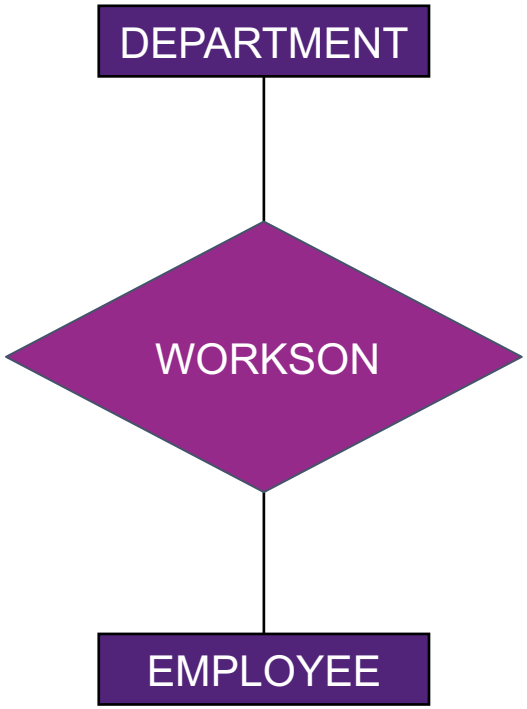
- 2 entities: Binary Relationship
- 3 entities: Ternary Relationship
- n entities: N-ary Relationship (3+ Entities)



Entity Roles

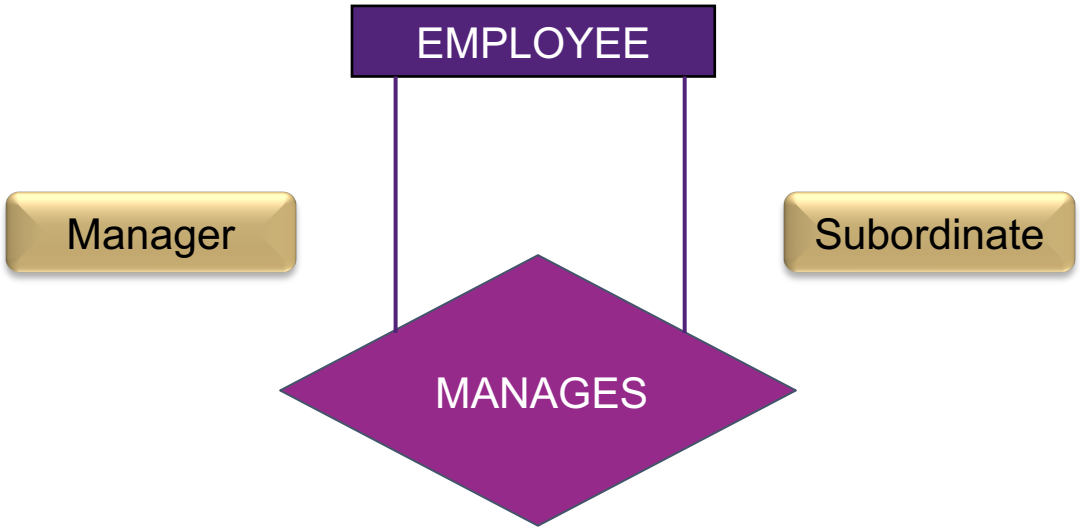
Each entity type that participates in a relationship type plays a particular **role** in the relationship type.

The **role name** signifies the role that a participating entity from the entity type plays in each relationship instance. That is to say, *it explains what the relationship means.*



Recursive Relationships

Same entity types can participate more than once in the same relationship type under different “roles”.



Question: Entities and Relationships

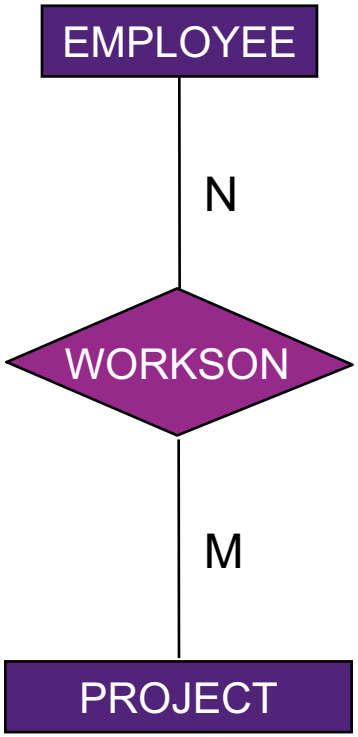
Create an ER diagram to store information about students, courses, the courses students have taken, and the grades students have gotten in these courses. Some relevant details are:

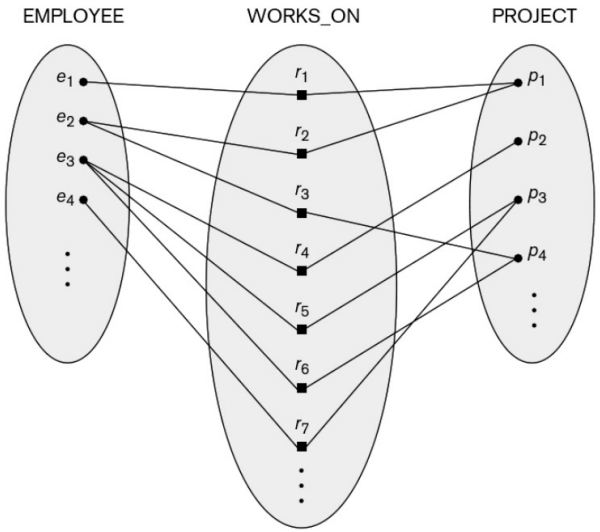
- Courses have a number, a department, and a title.
- Numbers are assigned by departments, and different departments may use the same number.
- Students are represented by their (unique) student ID and their name.
- Students can enrol in courses and upon completion of the course receive a grade.

Relationship Set

The collection of all relationships of a particular relationship type in the database at any point in time is referred to as a **relationship set**.

There are different methods of representing relationship sets.





WORKSON

WorksOn ID	Employee ID	Project ID
r1	e1	p1
r2	e2	p1
r3	e2	p4
r4	e3	p2
r5	e3	p3
r6	e3	p4
r7	e4	p4
…		

Database Design Overview

Entities and Relationships

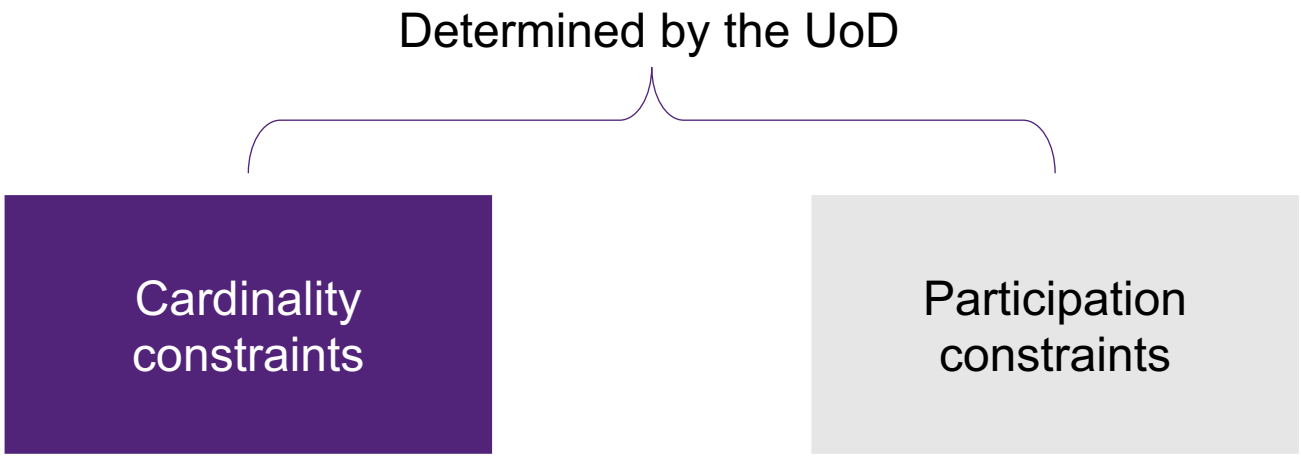
Relationship Constraints

The EER Diagram

Design Choices

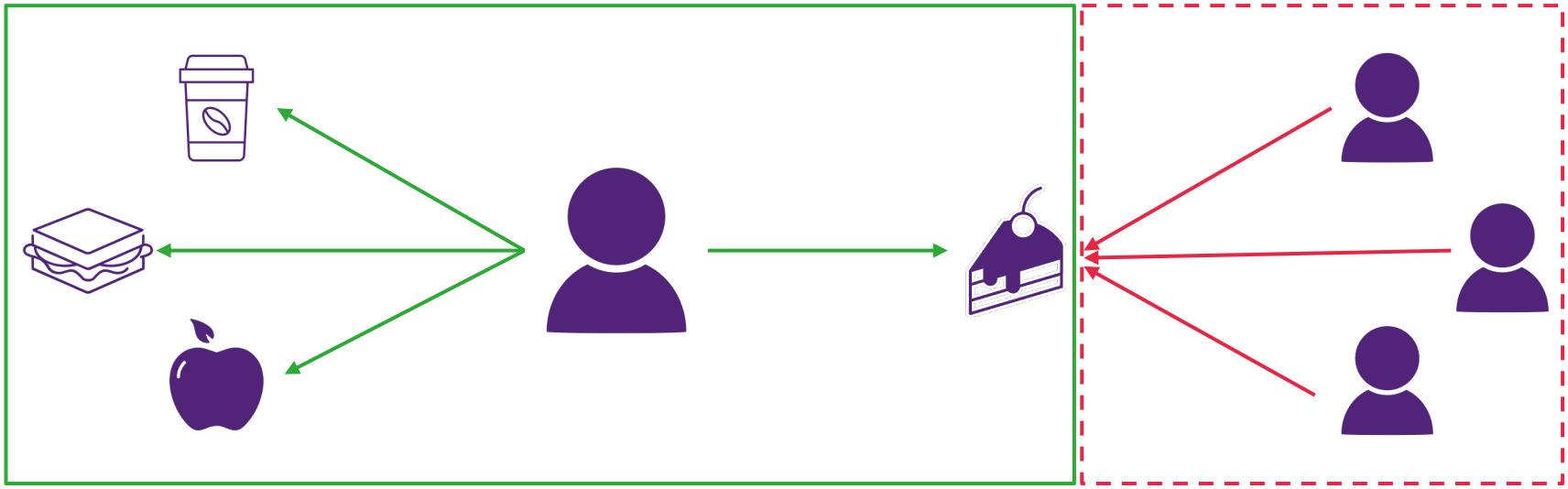
Relationship Constraints

Constraints on the relationship type limit the possible combination of entities that may participate in the corresponding relationship set.



Cardinality Ratio

A **cardinality ratio** for a relationship set specifies the number of relationships in the set that an entity can participate in. Let's consider the example of a **customer** purchasing **item**s.



One particular customer can purchase **many** different items

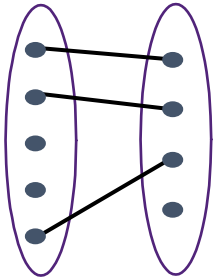
…*however*…

an individual item (E.g. a cake) cannot be purchased by several different customers.

Cardinality Ratio

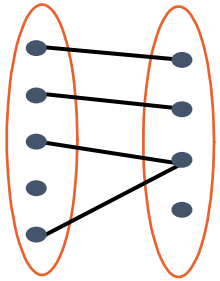
A **cardinality ratio** for a relationship set specifies the number of relationships in the set that an entity can participate in.

1-to-1 (1:1)



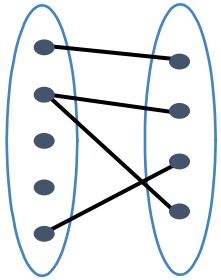
Both entities can participate in only one relationship instance

Many-to-1 (N:1)

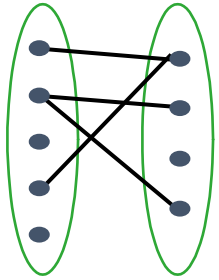


One entity can participate in many relationship instances

1-to-Many (1:N)



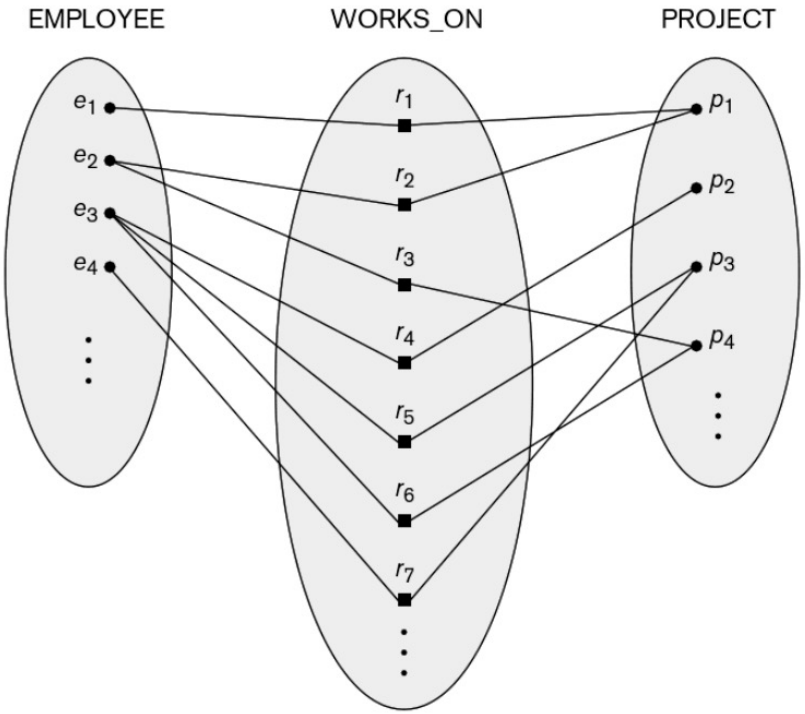
Many-to-Many (M:N)

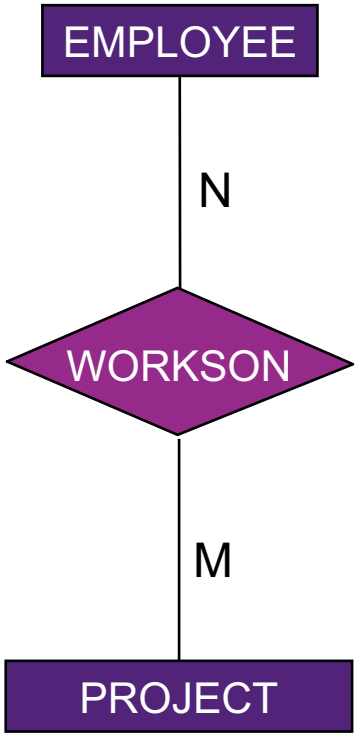


Both entities can participate in many different relationship instances

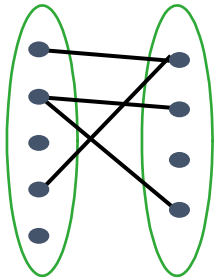
Example Cardinality Constraints

"Each employee can work on any number of projects, and each project can have any number of employees working on it."



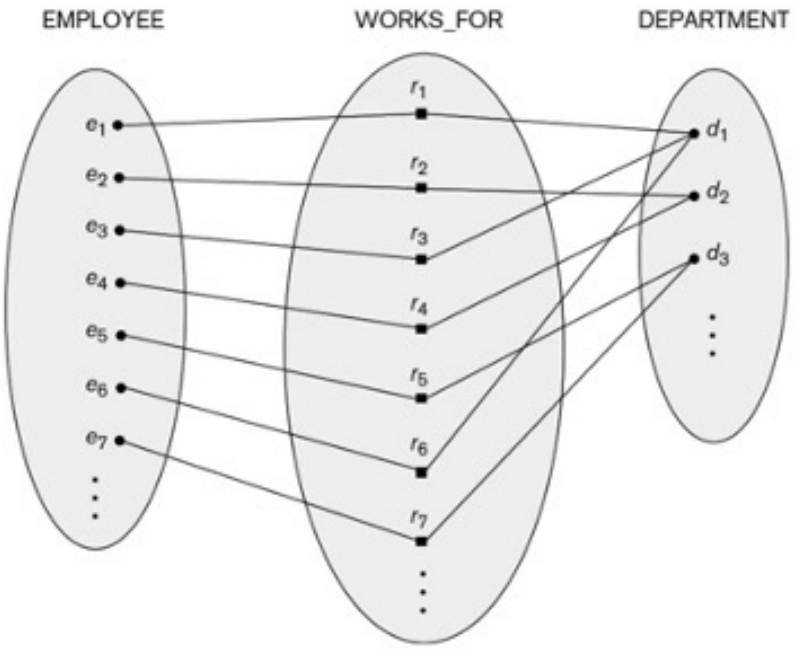


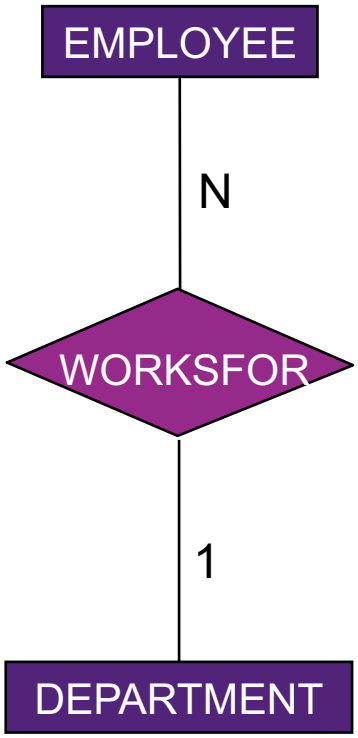
Many-to-Many (M:N)



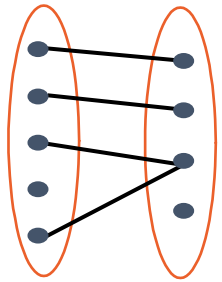
Example Cardinality Constraints

"Each department can have any number of employees, but an employee can work for at most one department."

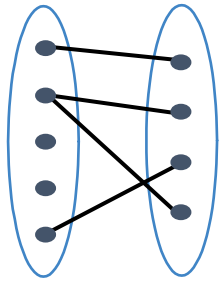




Many-to-1 (N:1)

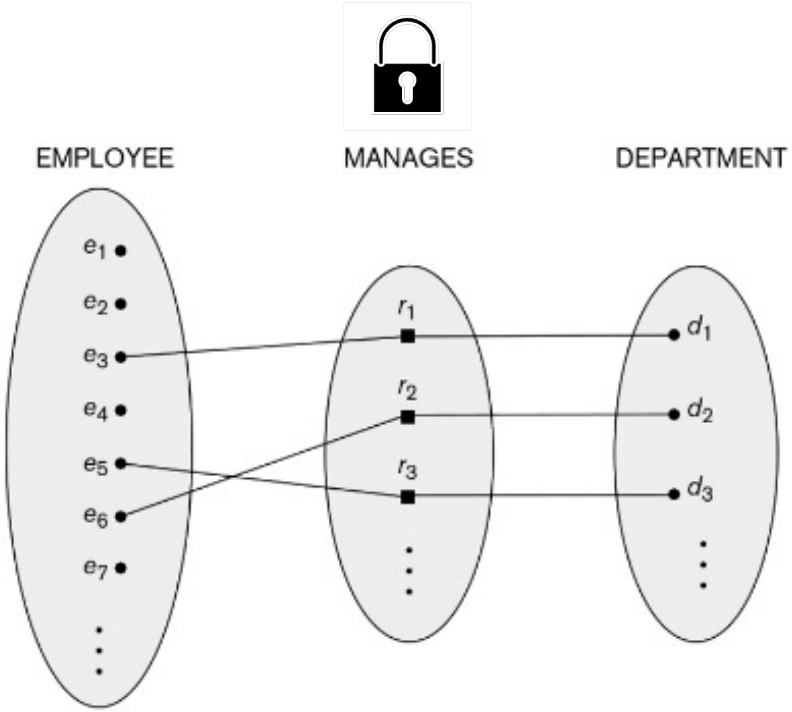


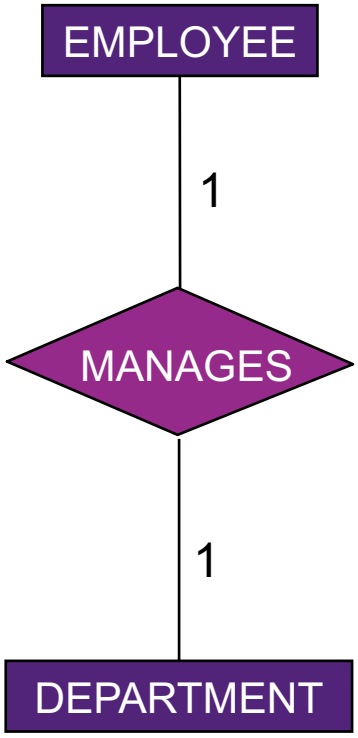
1-to-Many (1:N)



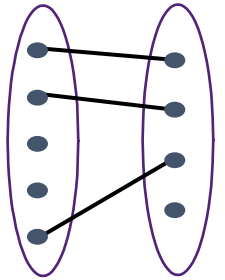
Example Cardinality Constraints

"Each department can have at most one manager and each employee can manage at most one department."

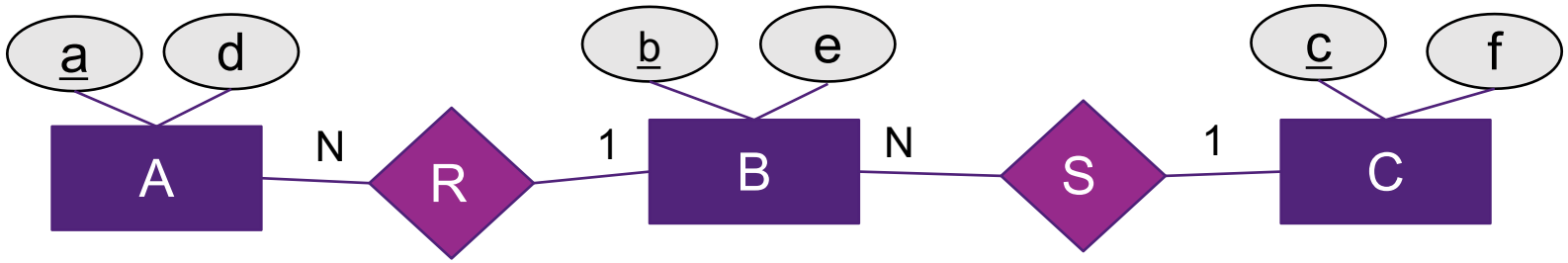




1-to-1 (1:1)



Question: Relationship Constraints



Suppose that A contains entities a1 and a2, B contains entities b1 and b2, and C contains entities c1 and c2. Which of the following presents possible relationship sets for R and S?

A.

R	

S	
b2	c1
b2	c2

B.

R	

S	
b1	c2
b2	c2

C.

R	
a2	b2

S	
b2	c1
b2	c2

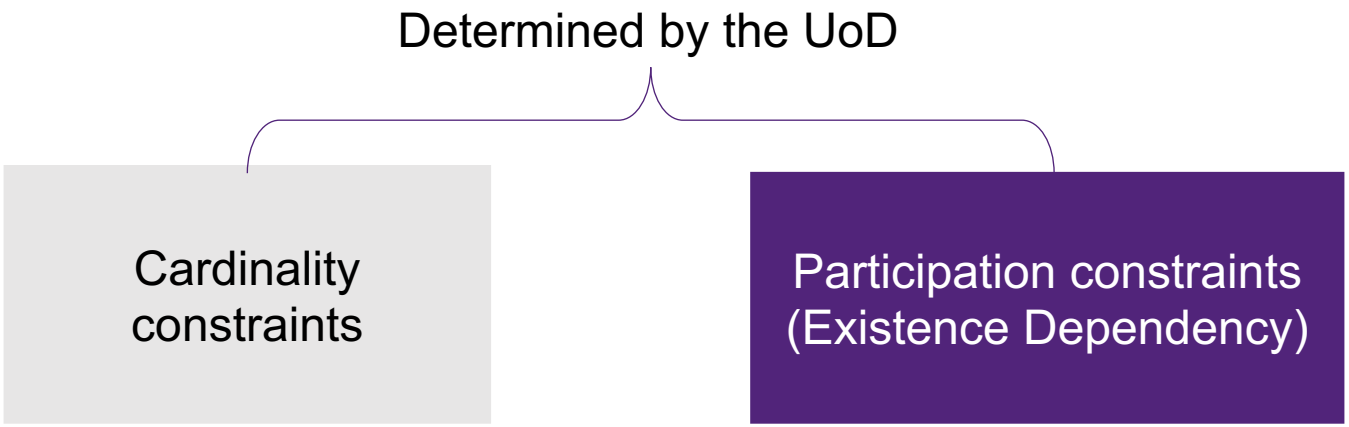
D.

R	
a1	b2
a2	b1
a2	b2

S	

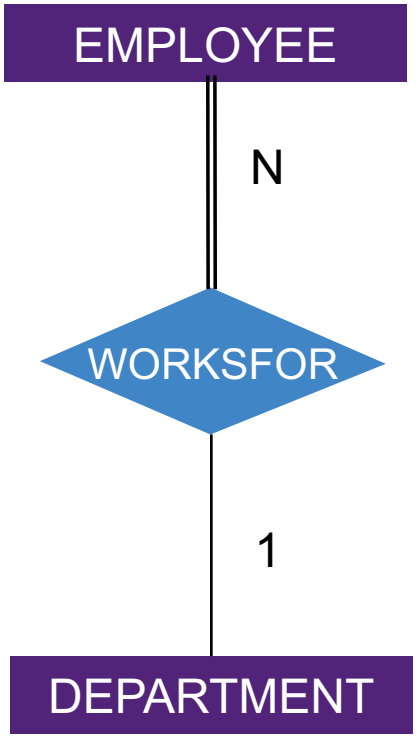
Relationship Constraints

Constraints on the relationship type limit the possible combination of entities that may participate in the corresponding relationship set.



Existence Dependency

Existence dependency indicates whether the existence of an entity depends on its relationship to another entity.

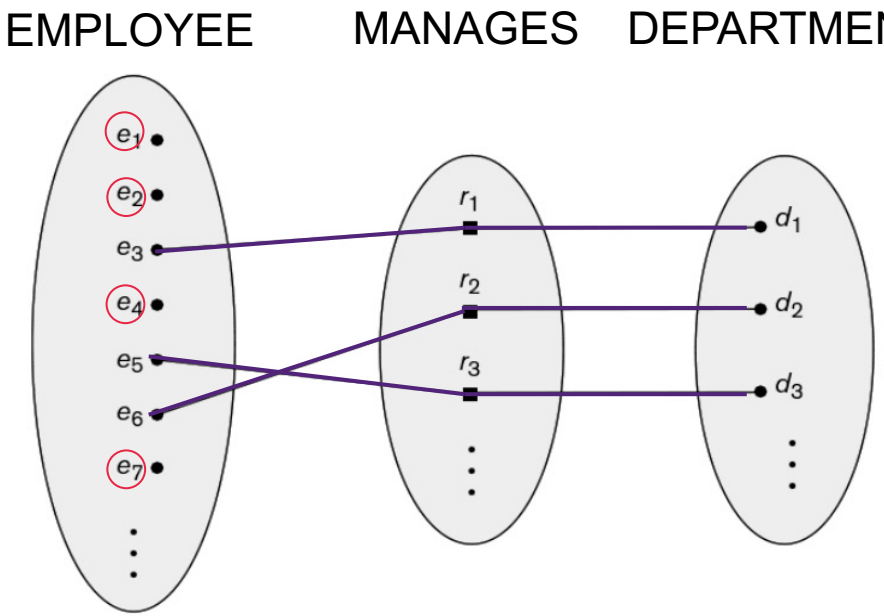


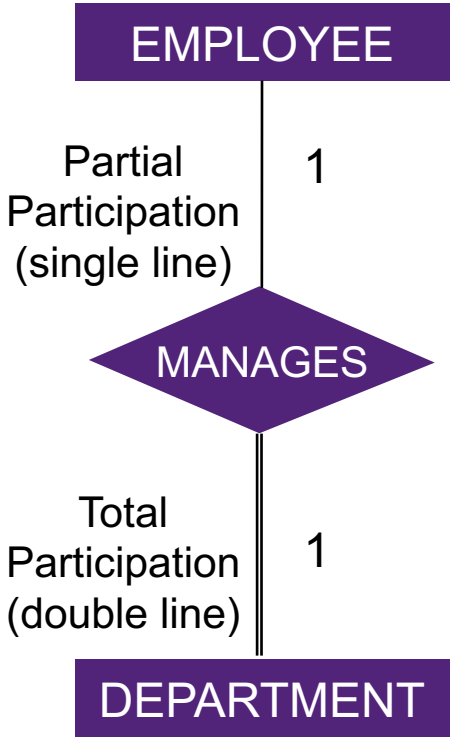
Every employee must work for a department.

Employee is existentially dependent on **Department** via the **WORKSFOR** relationship type.

Existence Dependency

Existence dependency indicates whether the existence of an entity depends on its relationship to another entity.

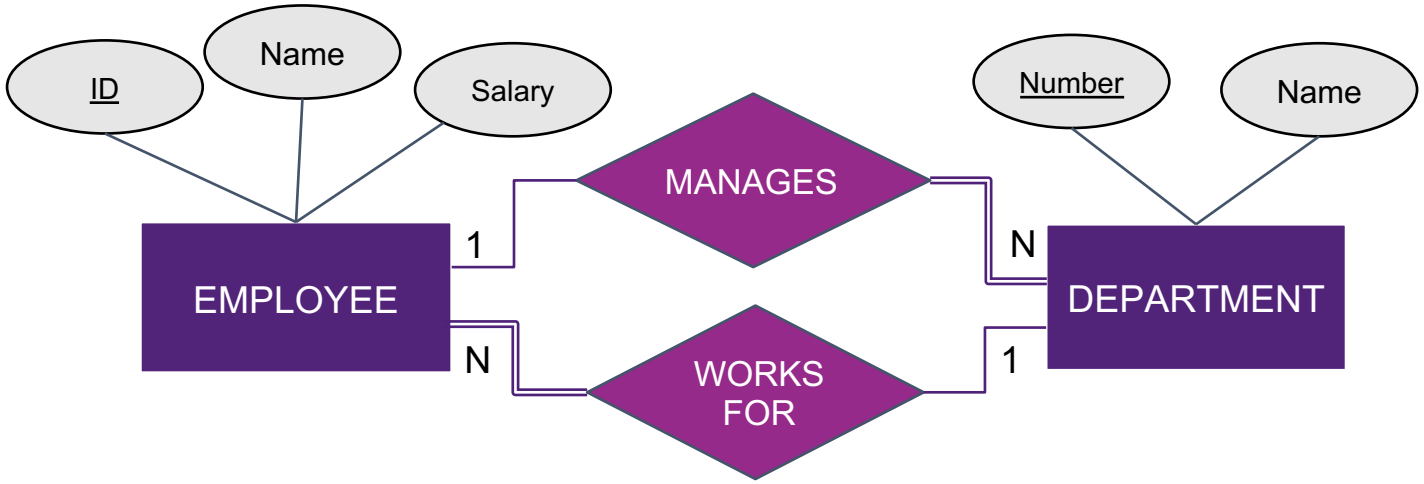




Every department must have a manger.

Department is existentially dependent on **Employees** via the **Manages** relationship type.

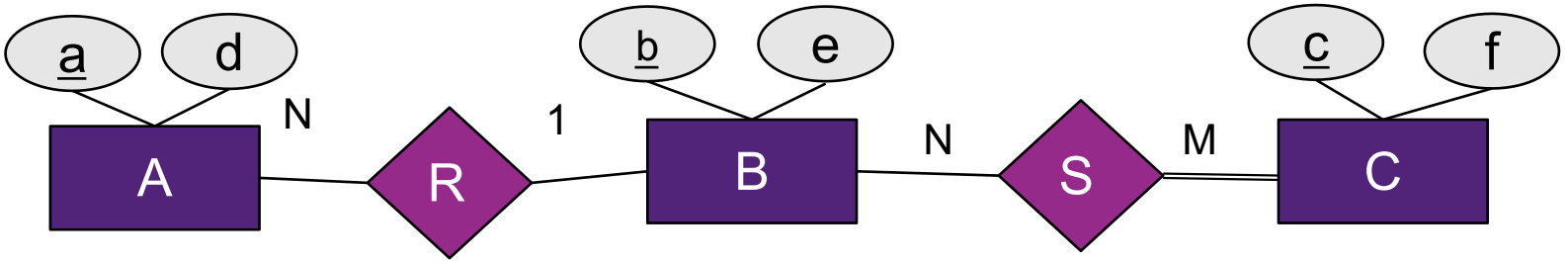
Relationship Constraints Example



1. Every department must have a manager
2. Every employee can manage 0 or more departments

3. Every employee must work for a department
4. Each department can have any number of employees

Question: Relationship Constraints



Suppose that A contains entities a1 and a2, B contains entities b1 and b2, and C contains entities c1 and c2. Which of the following presents possible relationship sets for R and S?

A.

R		S	

B.

R		S	
a1	b1	b2	c2

C.

R		S	
a1	b1	b1	c1
a1	b2	b2	c2

D.

R		S	
a1	b2	b1	c2
		b2	c1
		b1	c1

Exercise: ABC Banks

The ABC bank is organised into branches. Each branch is located in a particular city and is identified by a unique name. Each branch has an overall budget and rating (which is a number from 1 to 10).

A bank customer is identified by their customer name and phone number. The bank also keeps track of each customer's current address.

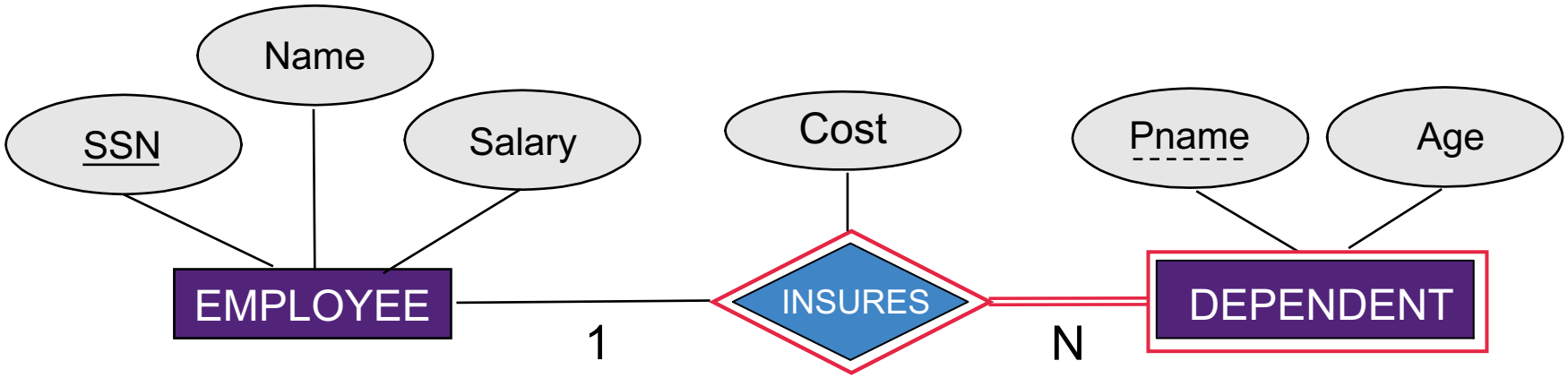
The bank offers accounts and loans to its customers. Each account and loan has a unique number and is created and maintained by a single branch.

Each account is assigned to one or more customers and its balance can never be negative.

A loan is always assigned to a single customer, has a fixed interest rate and its balance cannot be negative either.

Weak Entities

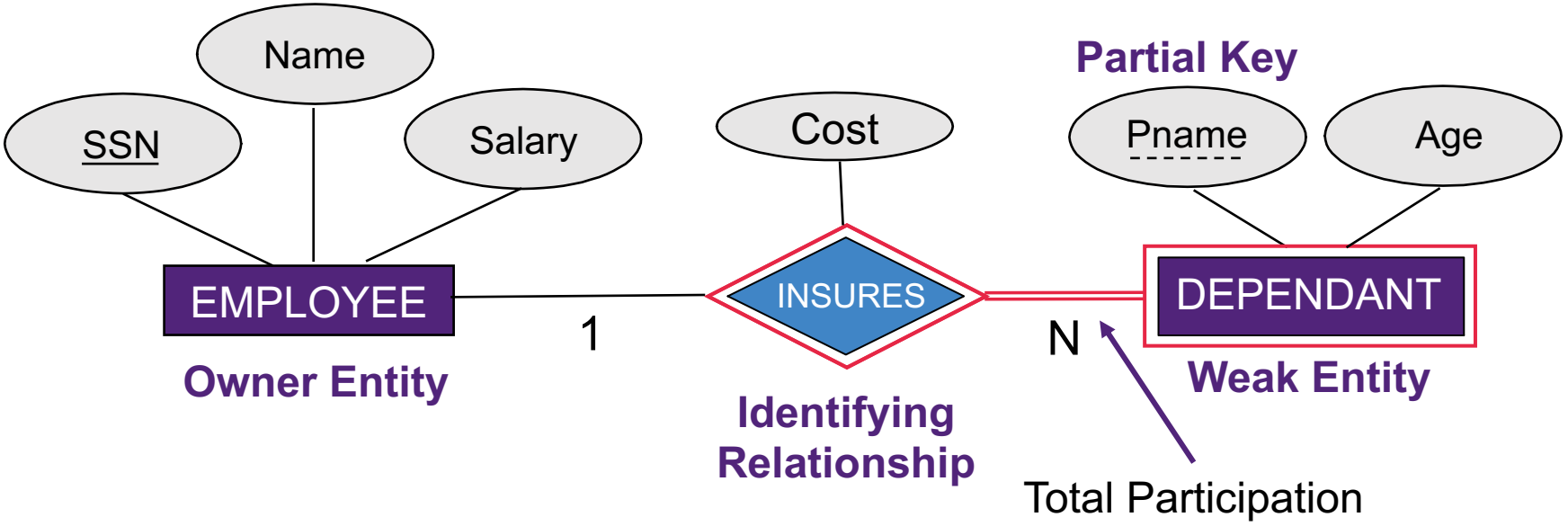
Entity types that do not have key attributes of their own are called weak entities.



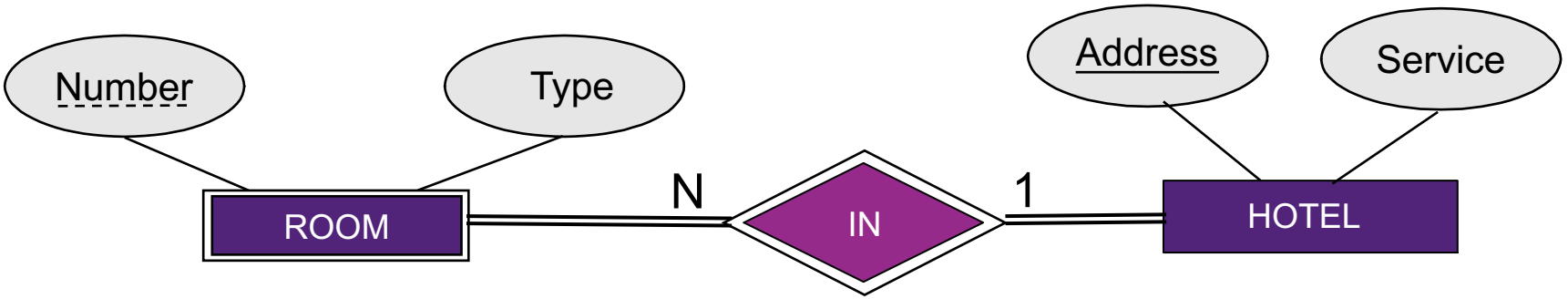
SSN	Pname
12345	John
22336	John

Weak Entities

- A weak entity can be identified uniquely only by considering the primary key of another owner entity and its own partial key, which is underlined with a dotted line.
- The relationship type that relates a weak entity to its owner is called the identifying relationship. A weak entity and its identifying relationship are represented with double lines.



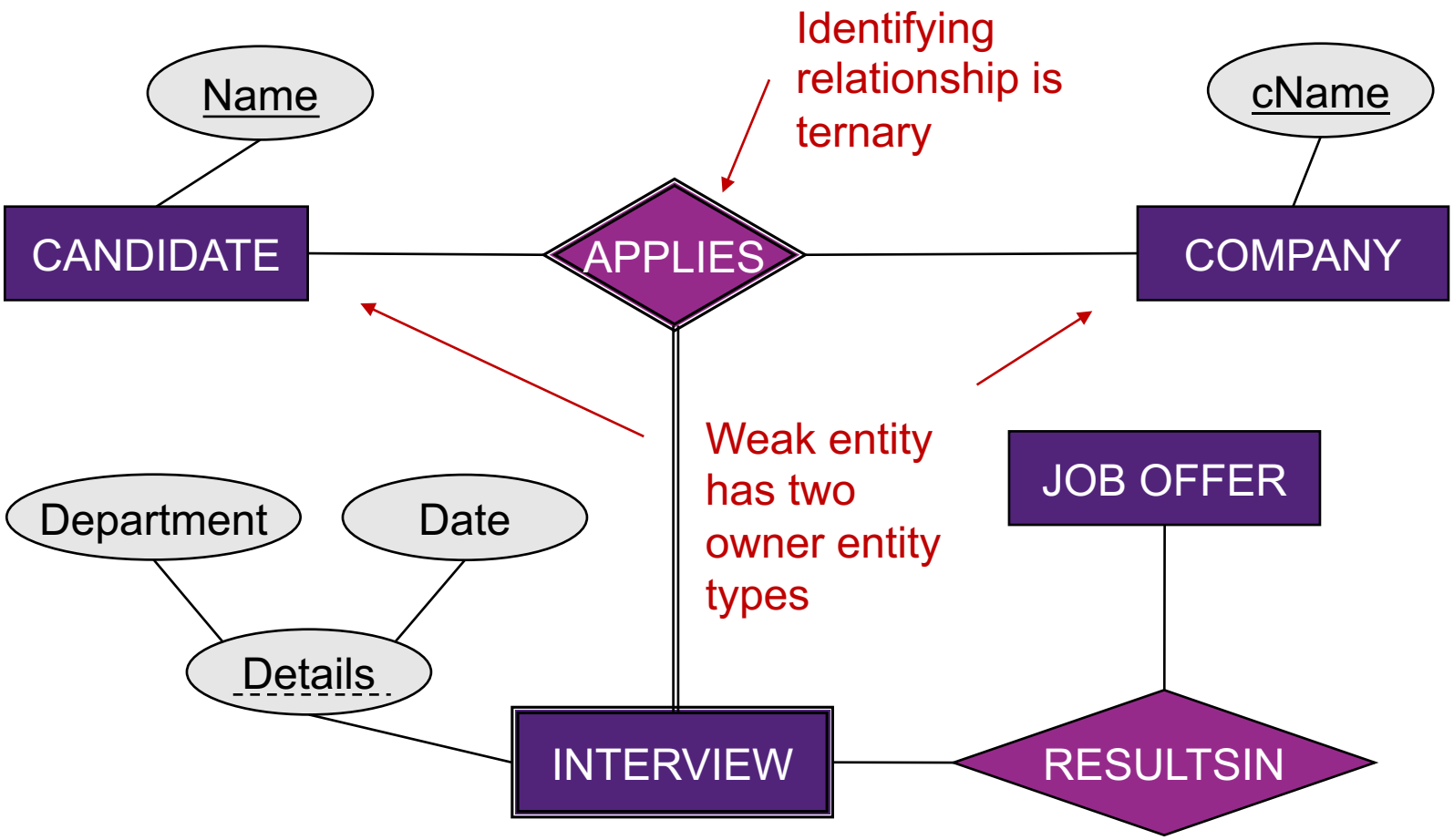
Question: Weak Entity



Which of the following is true?

A. Two hotels can have the same address.
B. No two rooms can have the same number.
C. No two hotels can have rooms with the same number.
D. No two room with the same number can have the same type.
E. None of the above

One More Example



Database Design Overview

Entities and Relationships

Relationship Constraints

The EER Diagram

Design Choices

The Enhanced ER (EER) Model

Entity Type is called **class** in EER

Entities in the same class have the same attributes

Class can be a **superclass** or **subclass**

- Attributes of a superclass are inherited by the subclasses
- Subclass can have its own specific attributes
- Subclass can have its own specific relationships

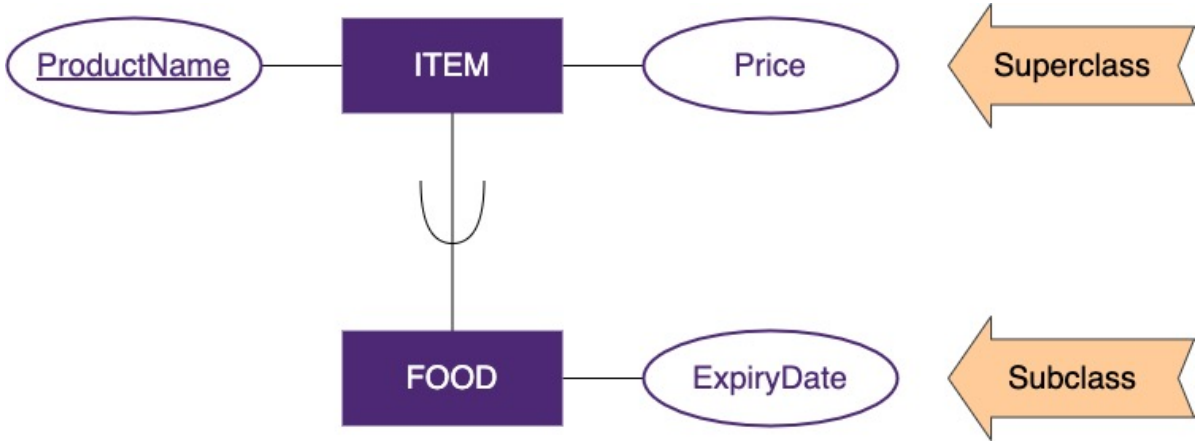
Every entity in a subclass is a member of its superclass(es)

E1 is the subclass of E2, as indicated by the direction of the $\subset$.

Motivating Example

You may purchase a variety of different items from a supermarket but what do all these items have in common? It's more than likely a **product name** and a **price**.

For some type of items though, say for example, food items there may be extra associated data such as an **expiry date**.



In this scenario, **"item"** is what we could call a **super class**. It captures the data common for a variety of different objects. A **"food"** item, which is just an extension of a regular item, is then what we would call a **subclass**.

Specialisation/Generalisation

There are two ways super/subclasses can be identified:

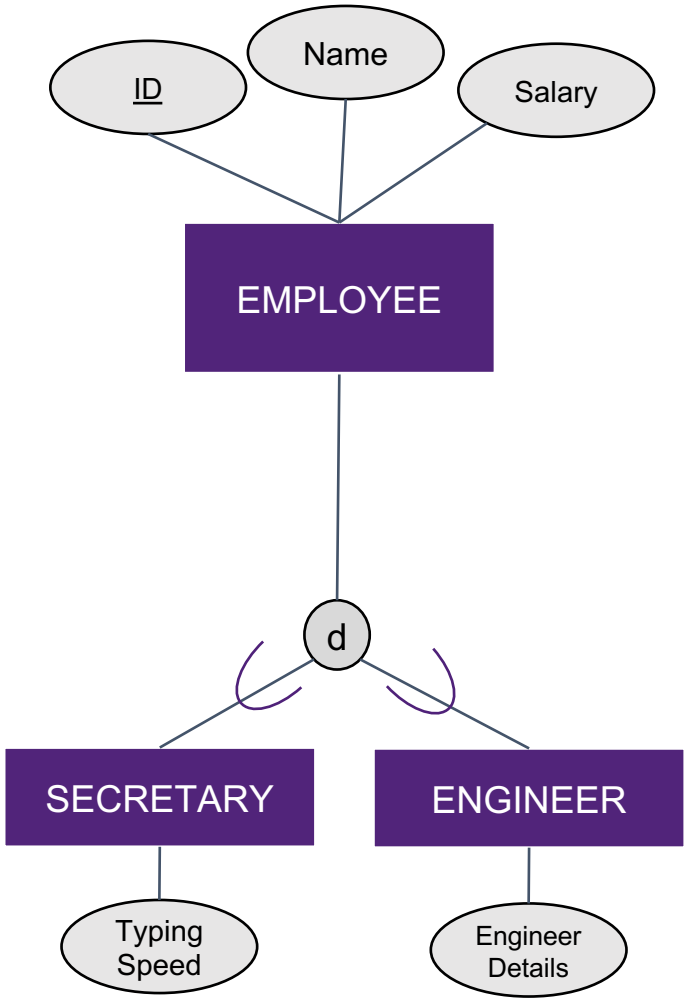
Specialisation

- Define a number of subclasses of an entity type
- Each subclass contains a subset of entities from the superclass
- A subclass is defined based on more specific distinguishing characteristic on entities of the superclass

Generalisation

- Abstraction is the process of ignoring differences amongst some subclasses and generalise them into a superclass

In summary, subclasses are specialisation of superclass, and superclass is generalisation of subclasses



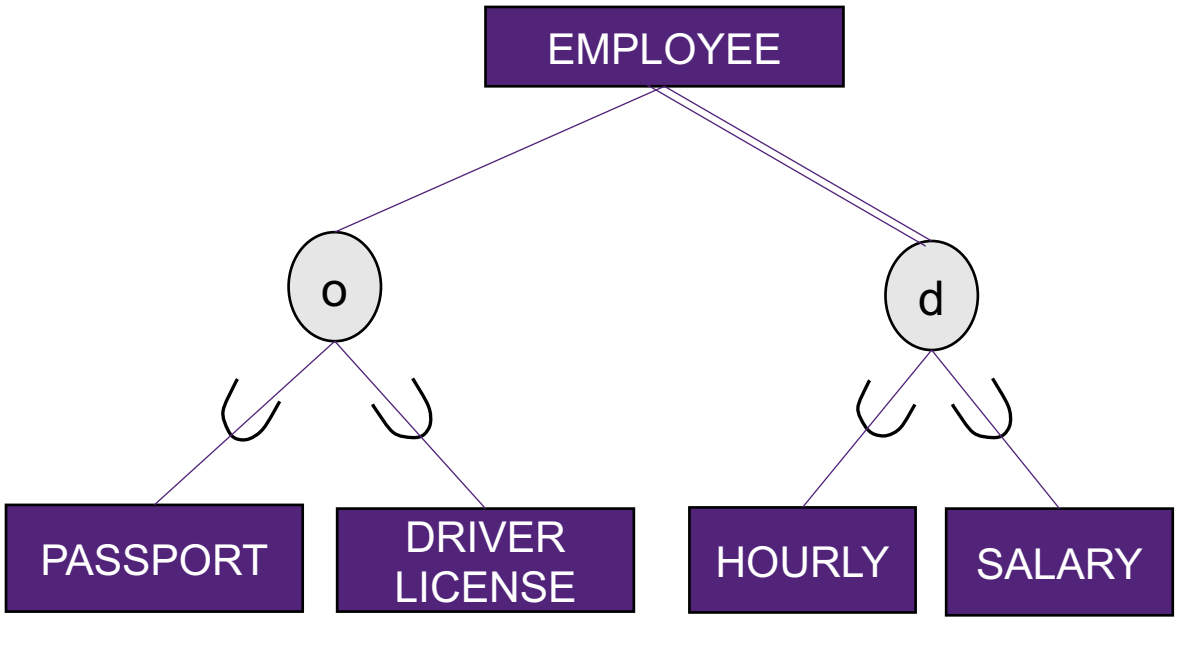
Constraints

Specialisation may be:

- total
- partial

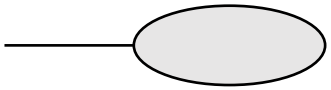
Subclass sets may be:

- overlapping
- disjoint

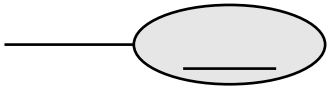


Notation Guide

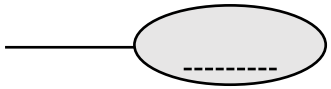
ENTITY TYPE



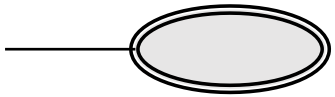
ATTRIBUTE



KEY ATTRIBUTE



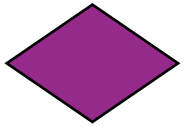
PARTIAL KEY ATTRIBUTE



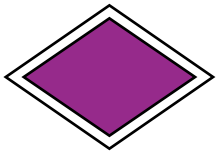
MULTIVALUED ATTRIBUTE

DERIVED ATTRIBUTE

WEAK ENTITY TYPE



RELATIONSHIP TYPE

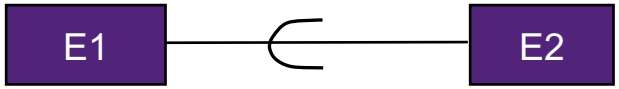


IDENTIFYING RELATIONSHIP TYPE

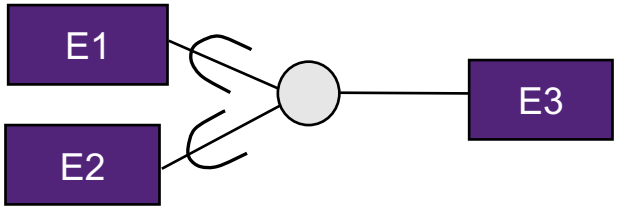
Notation Guide

TOTAL PARTICIPATION OF E2 IN R

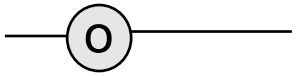
CARDINALITY RATIO 1:N FOR E1:E2 IN R



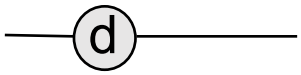
E1 IS A SUBCLASS OF E2



E1 and E2 ARE SUBCLASSES OF E3



Overlapping specialization



Disjoint specialization

Draw an ER diagram for the following:

- The primary function of UofU is to offers courses to students.
- A student is identified by a unique student id, and has a name, address and a phone number. Each student is registered in a major at UofU.
- Visiting students stay at UofU for a year.
- A course offered by UofU is identified by the department that offers the course and a course#, which is unique within the department. The title and number of credits of a course are also recorded.

Draw an ER diagram for the following:

- A course may be offered many times, even within the same semester. Each offering is assigned a section# which is unique for a given course and semester, and is taught by a single instructor.

- Each instructor is responsible for some section; there are no idle instructors. Instructors have unique names and may teach a number of sections of different courses. For each instructor we like to keep info about their higher degree.

- A student enrols in a course section and gets a mark for the course.

- A course may have any number of other courses as prerequisites.

Database Design Overview

Entities and Relationships

Relationship Constraints

The EER Diagram

Design Choices

Design Choices for ER Conceptual Design

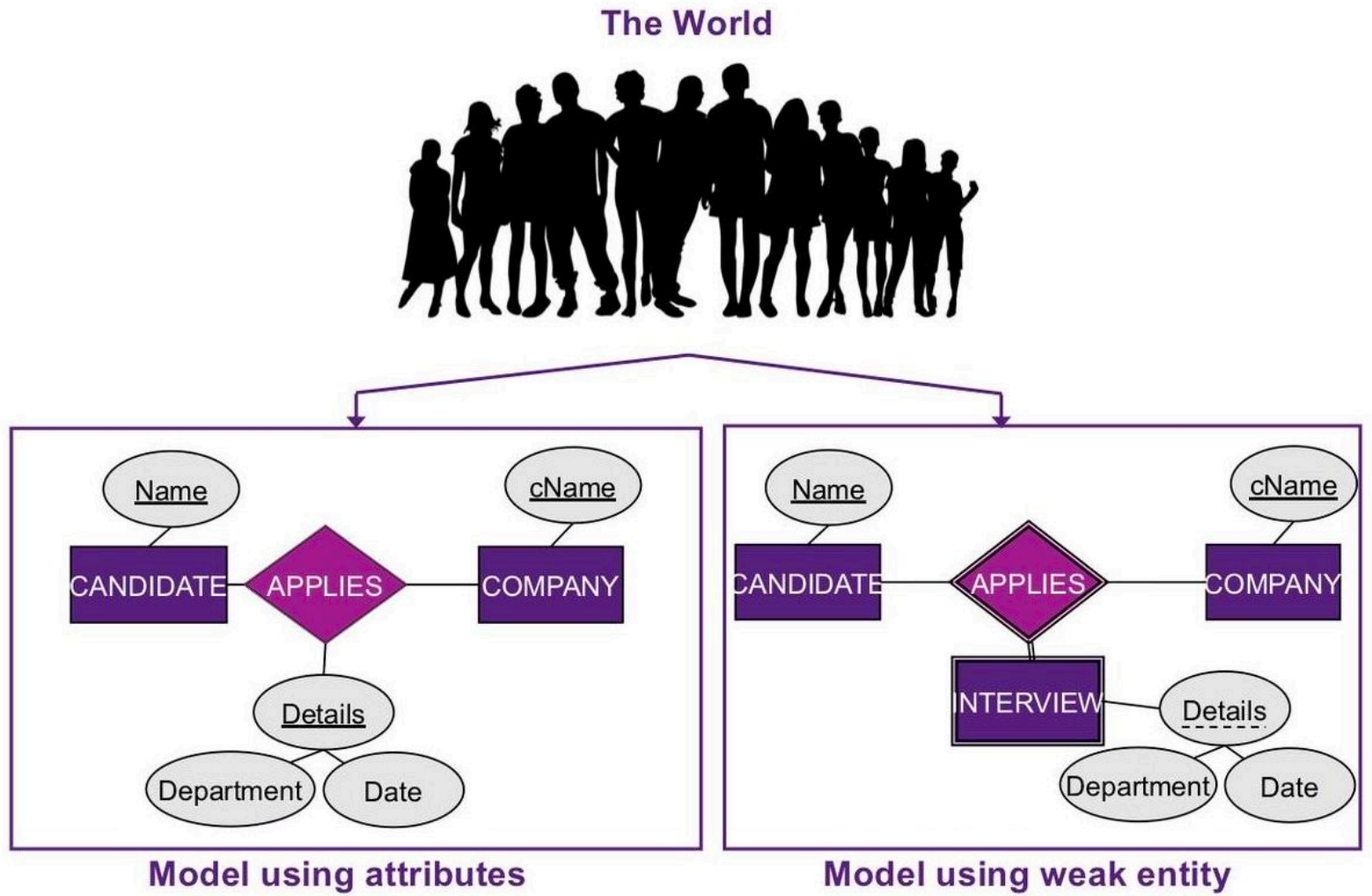
When modelling complex organisations, an database designer may face design choices. These choices generally fall into two categories:

- **Equivalent Choices:** An aspect of the UoD could be represented in an ER diagram using two different methods which are both equivalent (i.e. the resulting database will be the same).
- **Inequivalent Choices:** When an UoD is slightly ambiguous, an aspect of the UoD could be mapped two different ways which both meet the specification of the UoD but may lead to different limitations/constraints being enforced by the database.

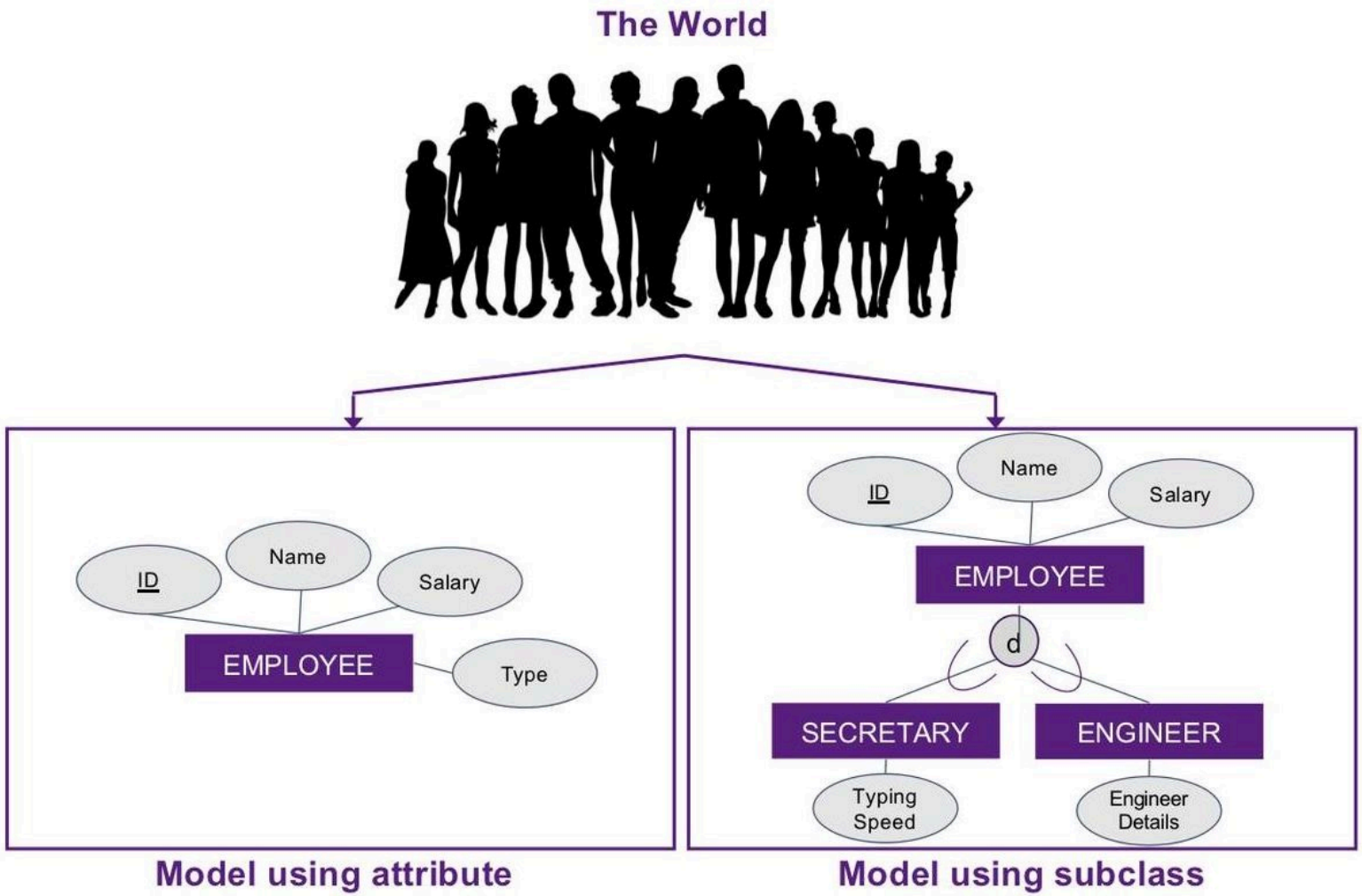
Some common design choice include:

- Attribute vs. (weak) entity type
- Attribute vs. subclass
- Subclass relationship vs. superclass relationship
- Binary relationships vs. N-ary relationships

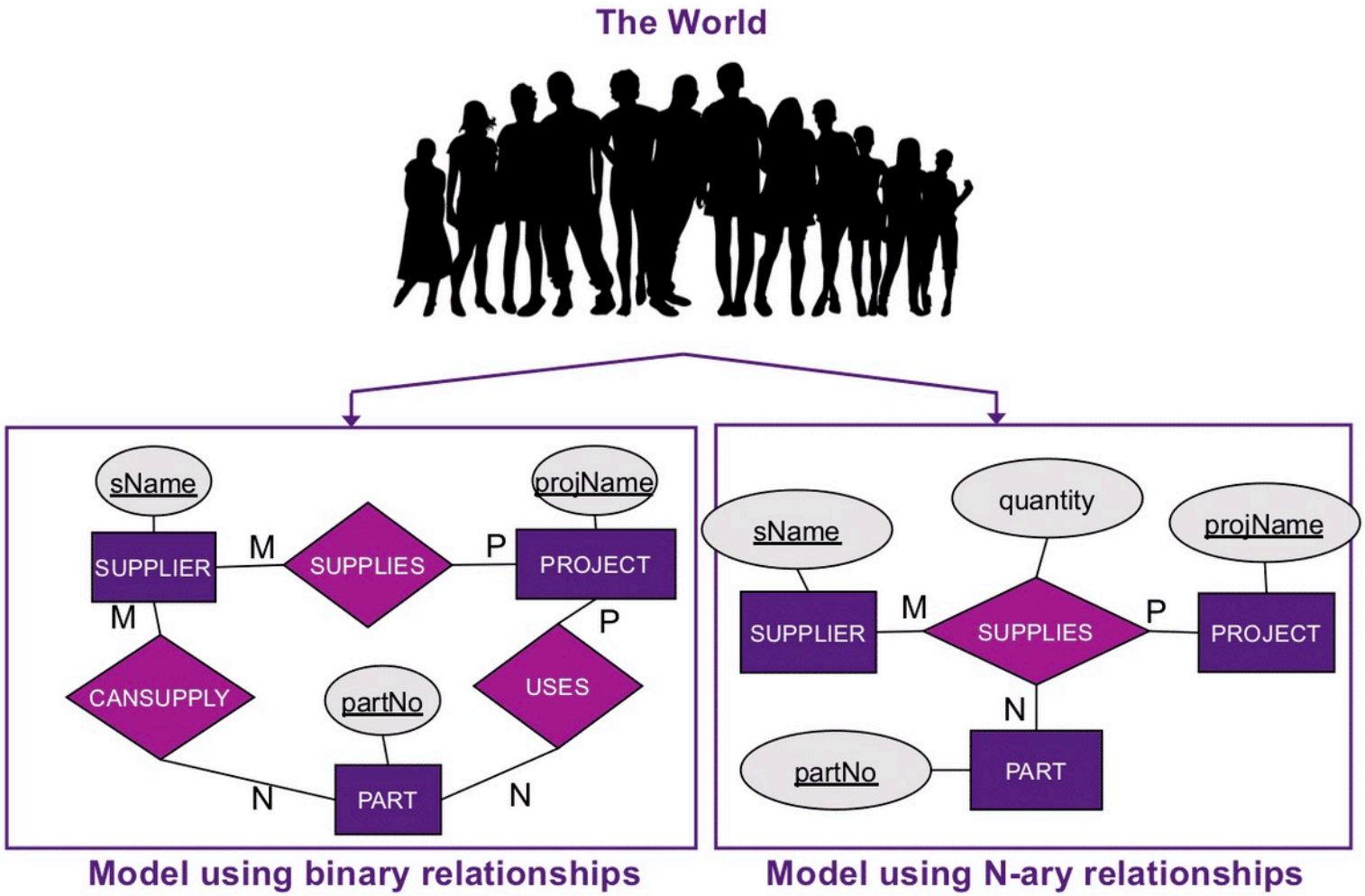
Design Choice Example: **Attribute vs. (Weak) Entity Type**



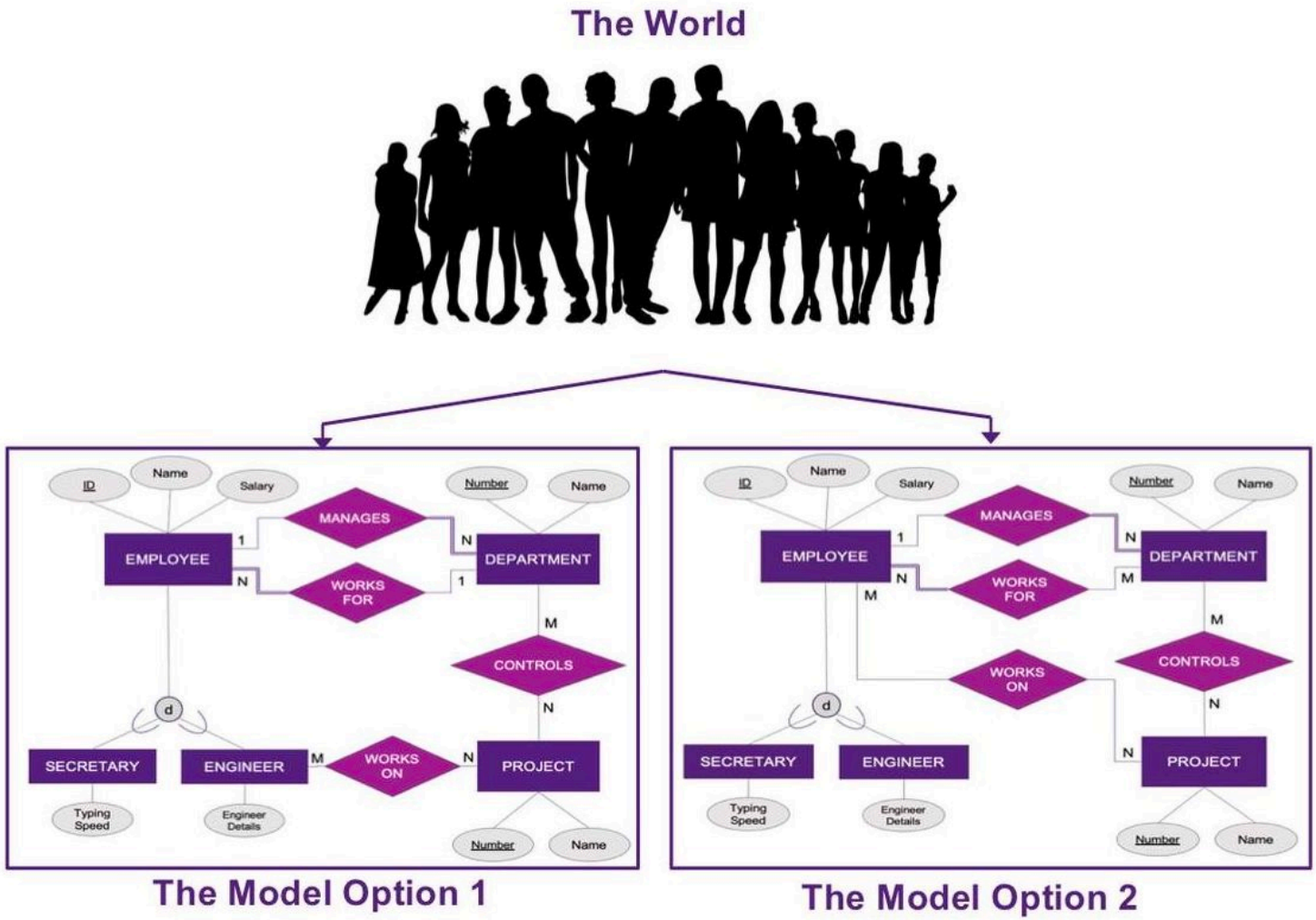
Design Choice Example: **Attribute vs. Subclass**



Design Choice Example: **Binary Relationships vs. N-ary Relationships**



Design Choice Example: **Subclass Relationship vs. Superclass Relationship**



Learning Outcomes Revisited

Analyse, extract and structure information system requirements to create basic conceptual models.

- Define the concepts of entities, relationships, cardinality constraints, participation constraints, weak entities, superclasses and subclasses in Entity Relationship (ER) model.
- Given the description of a small operating environment, create an ER model of the environment.
- Identify alternative representations for modelling an operating environment and evaluate the choices.