Fall 2019

CS 220
Database Systems

Lecture 8

Types of Entities

Strong Entity

A strong entity type is an entity type whose existence is not dependent on another entity type.

A strong entity type always has a primary key of its own attributes that uniquely identifies its instances

e.g. Employee

Weak Entity

A weak entity type is an entity type whose existence is dependent on another entity type.

A weak entity type does not have a set of its own attributes that uniquely identifies its instances.

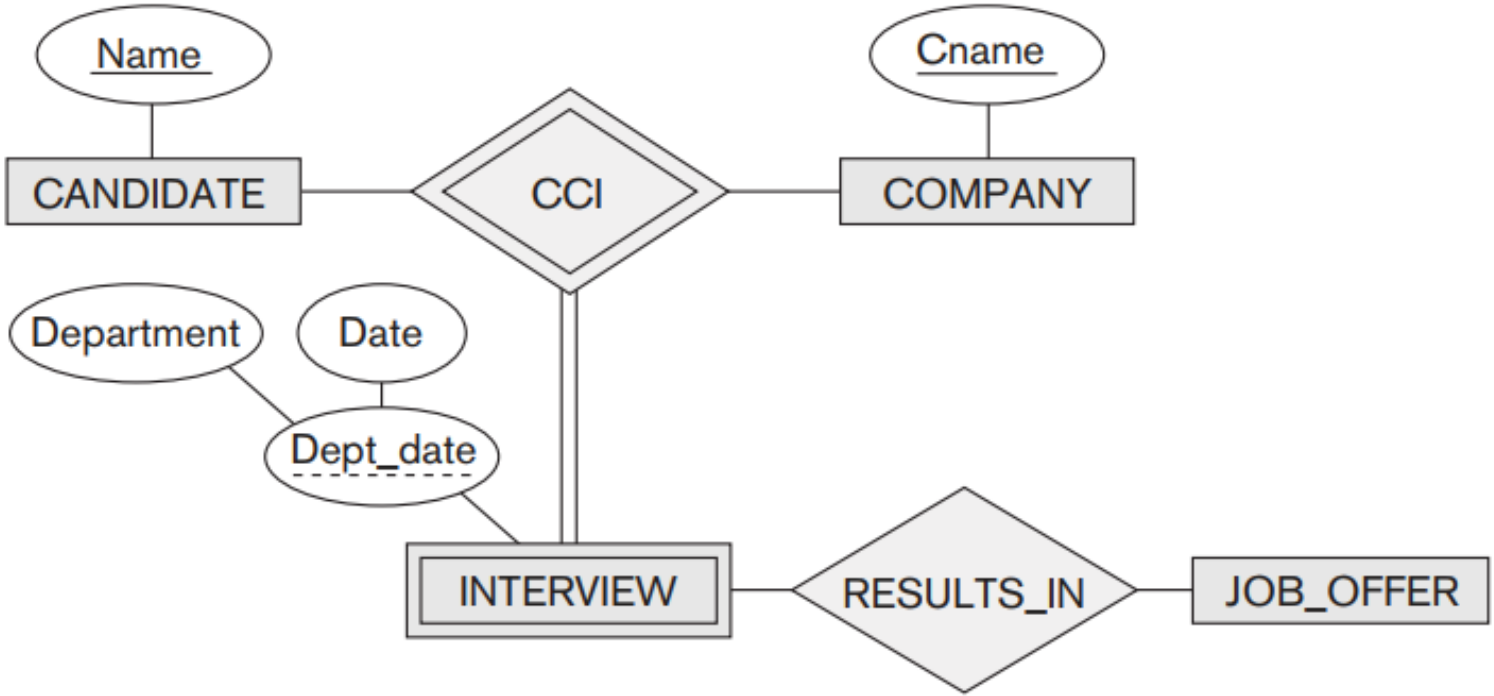
e.g. Dependent

Weak Entities

An employee is a strong entity because it has an employee number to identify its instances.
A dependent (child) is a weak entity because the database does not store a key for each child, but rather they are identified by the parent's employee number and their name

- A weak entity type normally has a **partial key**, which is the attribute that can uniquely identify weak entities that are related to the same owner entity. The partial key attribute is underlined with a dashed or dotted line
- In ER diagrams, both a weak entity type and its identifying relationship are distinguished by surrounding their boxes and diamonds with double lines

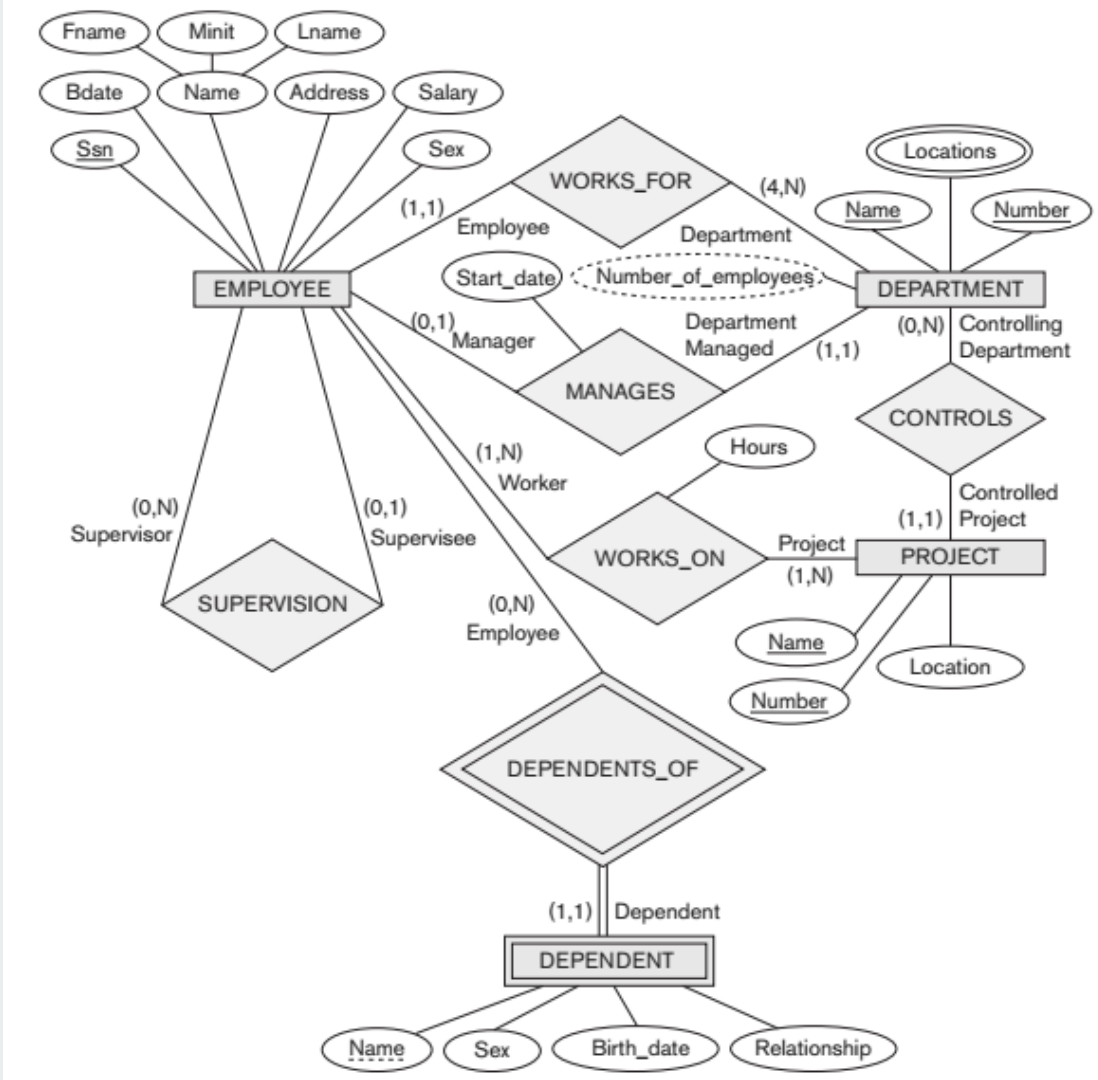
Weak Entities



Weak Entities

- Weak entity types can sometimes be represented as complex (composite, multivalued) attributes.
- In the preceding example, we could specify a multivalued attribute **Dependents** for **EMPLOYEE**, which is a multivalued composite attribute with the component attributes *Name, Birth_date, Sex, and Relationship*.
- The choice of which representation to use is made by the database designer.
- One criterion that may be used is to choose the weak entity type representation if the weak entity type participates independently in relationship types other than its identifying relationship type.

Entity Relationship Diagram



Alternative Representations to ER Diagrams

UML Class Diagrams

An entity in ER corresponds to an object in UML

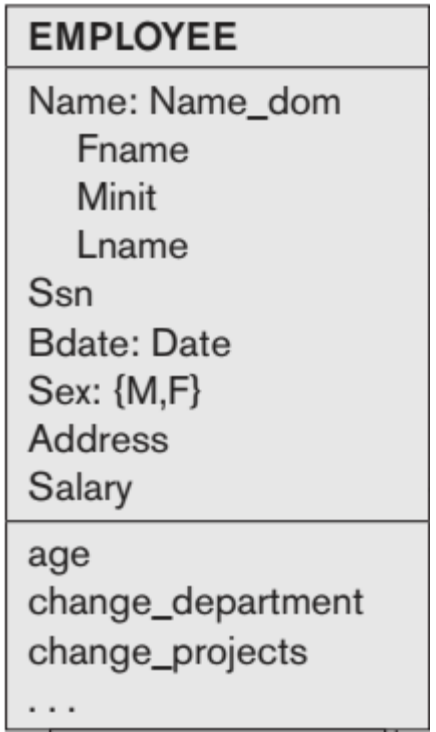
Entity is represented as a box

The top section gives the entity type name

The middle section includes the attributes

The last section includes operations* that can be applied to individual objects

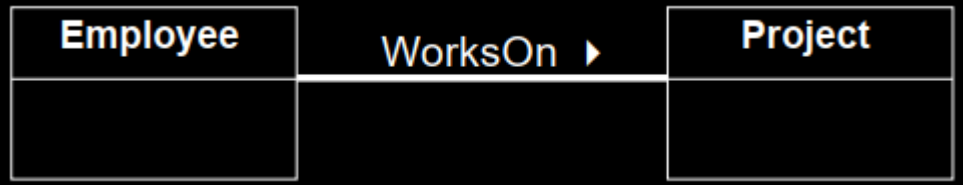
Operations are not specified in ER diagrams



UML Class Diagrams

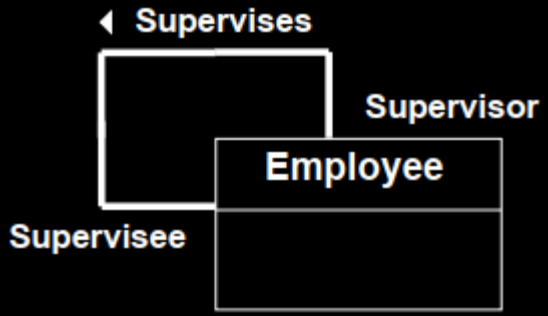
A multivalued attribute will generally be modeled as a separate class

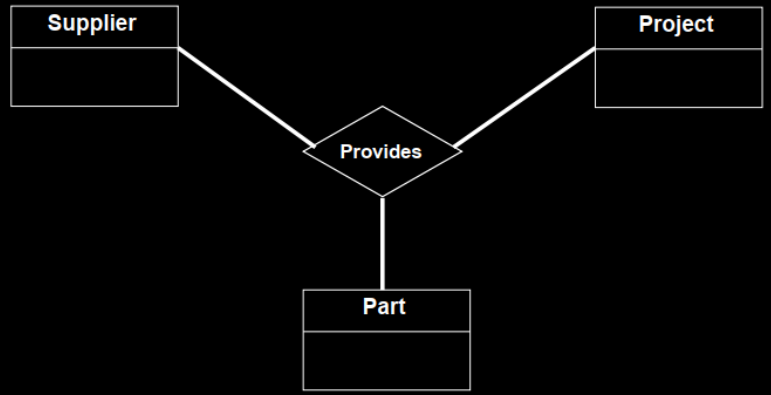
Relationship types are called associations represented as a line connecting the participating entities



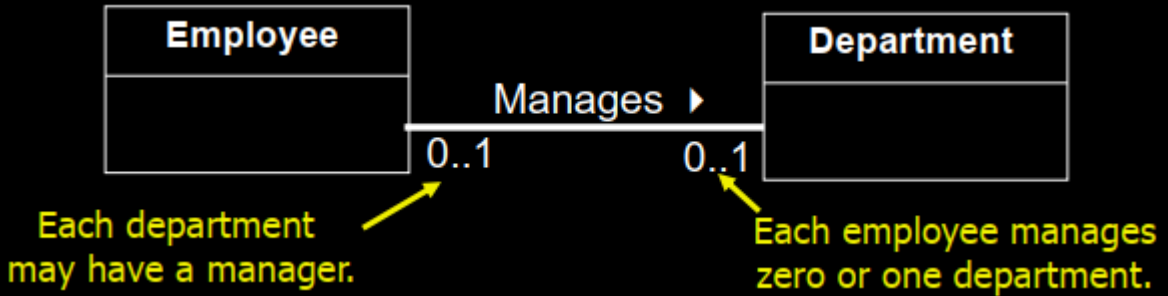
UML Class Diagrams

Tertiary & Recursive Relationship



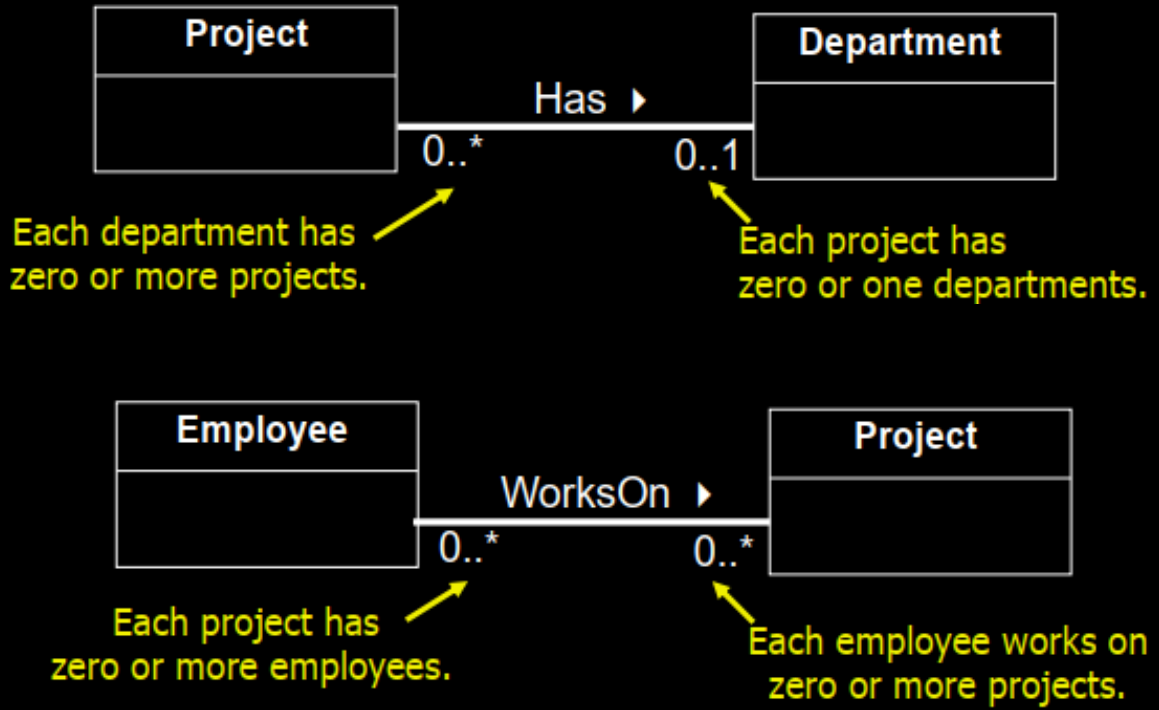


Relationship Cardinalities



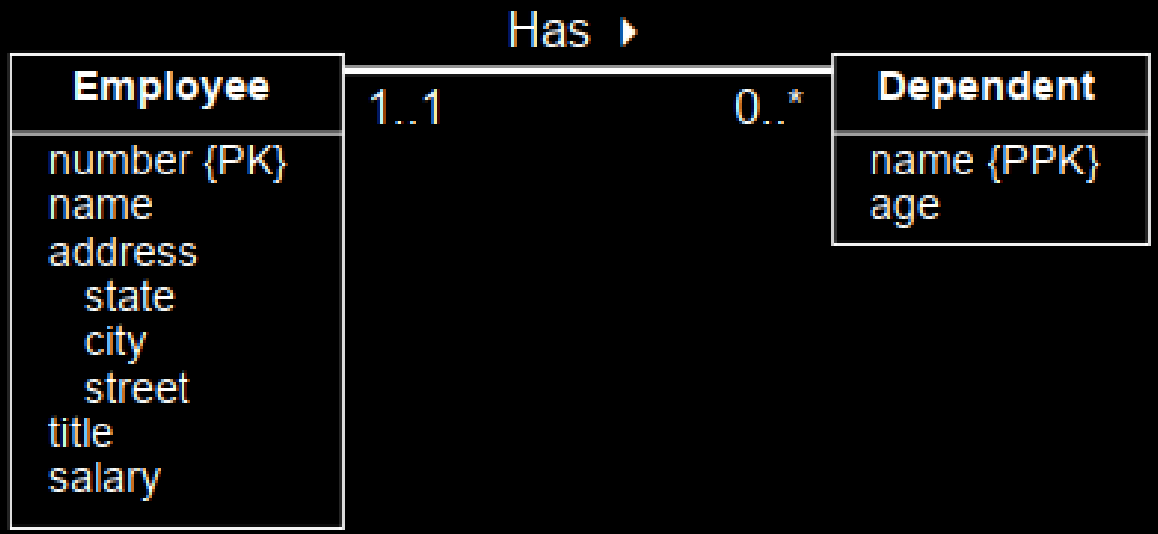
UML Class Diagrams

Relationship Cardinalities

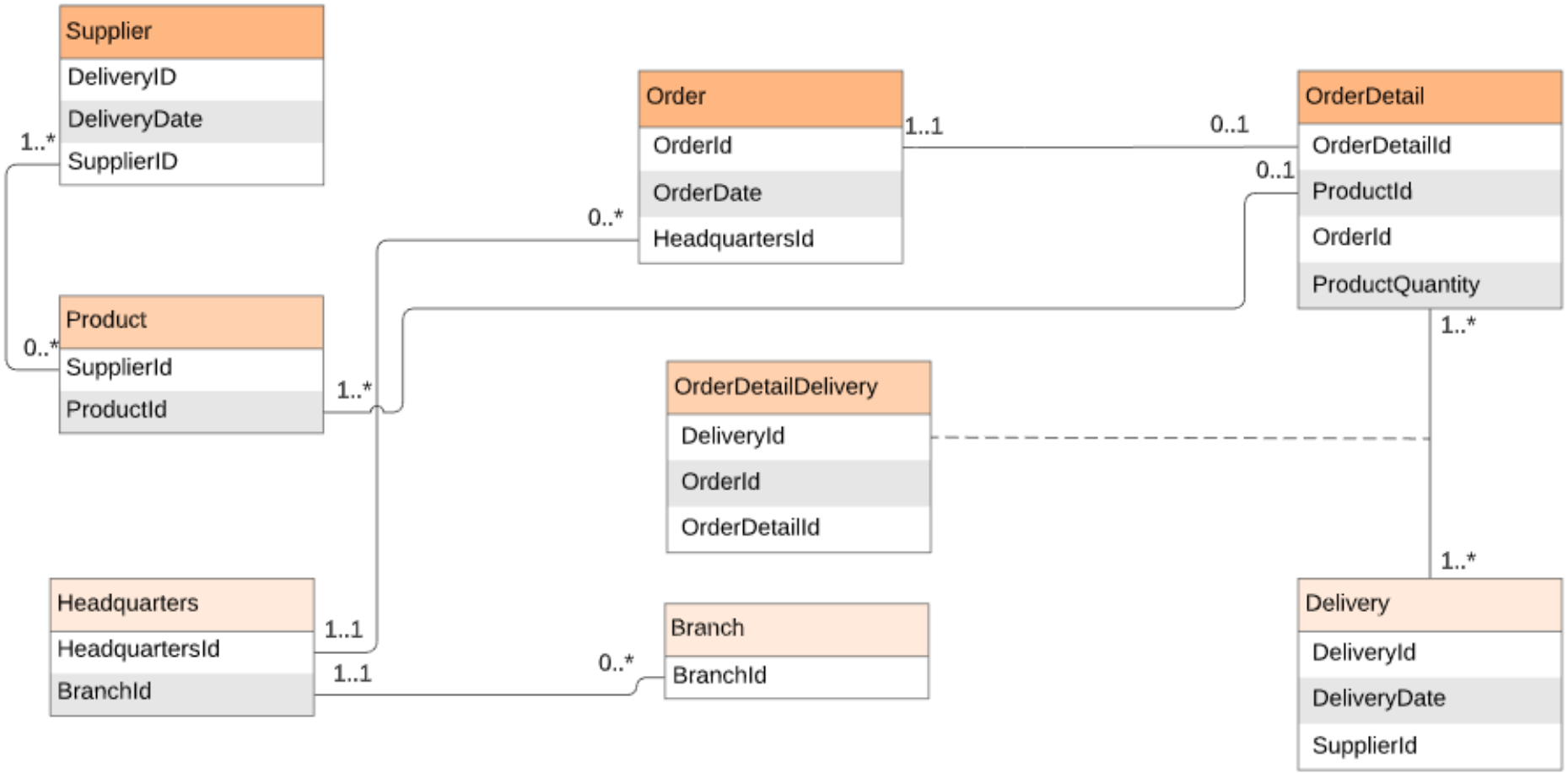


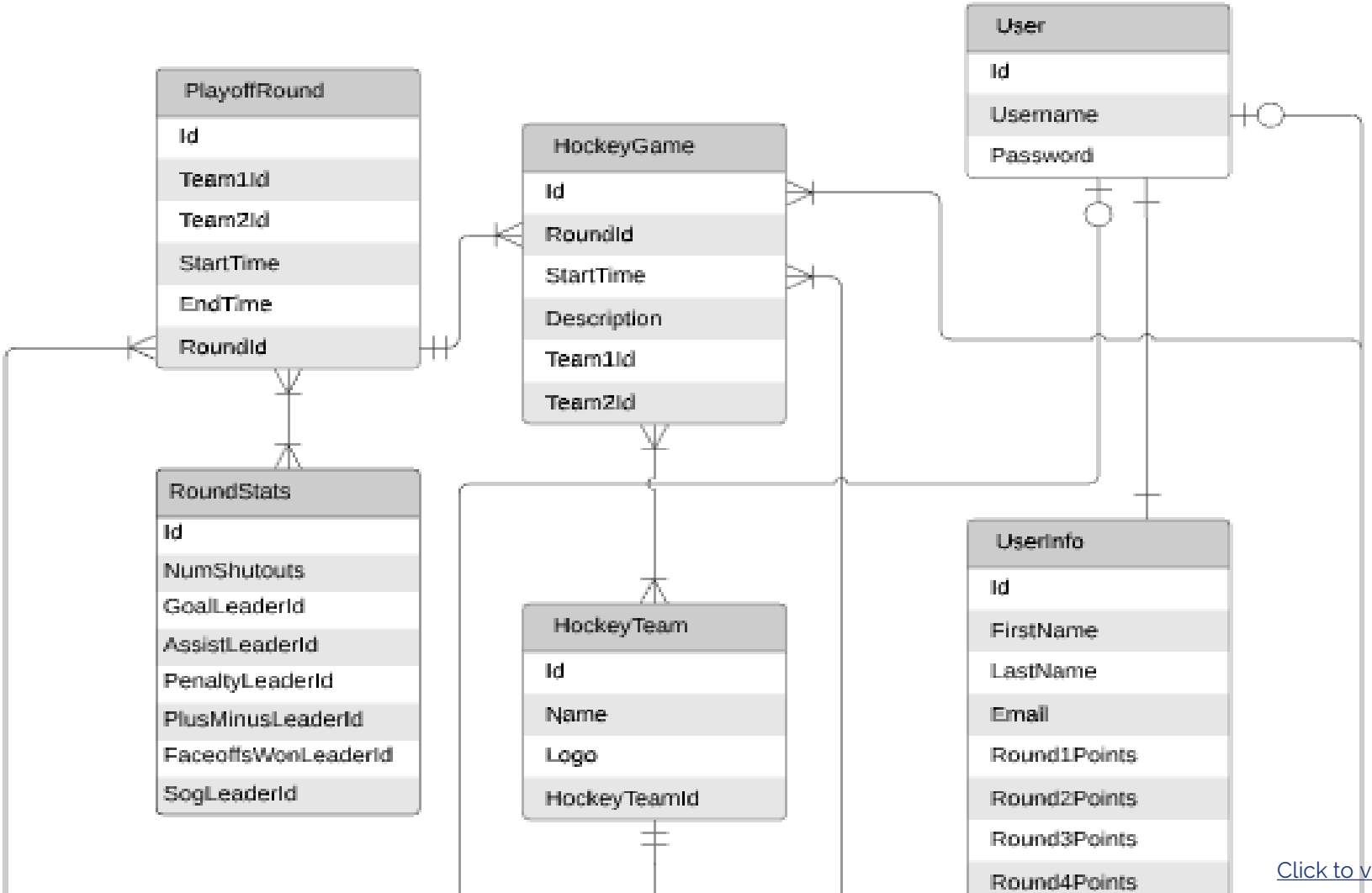
UML Class Diagrams

Weak Entities in UML



UML Class Diagram





Click to view a short video

More on E-R Modeling

Weak Entity Sets

Avoid the use of weak entity sets in modeling. Although at first glance there are very few natural keys based on the properties of objects (name, age, etc.), there are many good human-made keys that can be used (SSN, student#, etc.)

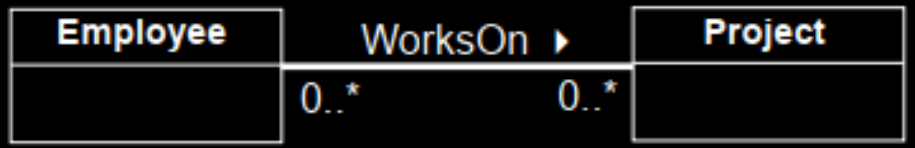
Whenever possible use a global, human-made key instead of a weak entity. Note that sometimes a weak entity is required if no such global key exists.

- For example, a database storing the students of multiple universities could not use a single student# as a key as it is unlikely that universities could agree on a global numbering system. Rather, student becomes a weak entity with partial key student# and identifying entity the university the student attends.

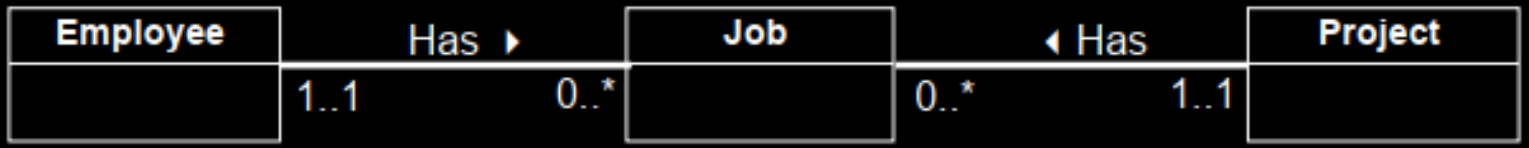
Simplifying Many to Many Relationship

A many-to-many relationship can be converted into one entity with two 1:N relationships between the new entity and the original entities participating in the relationship.

Original:



Simplified:

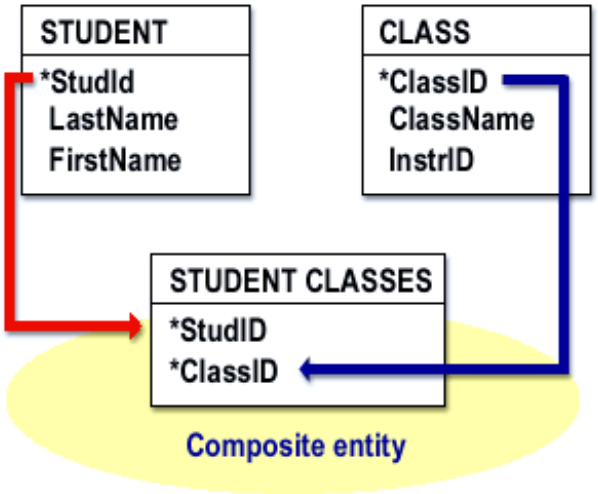


Composite Entity

The M:N relationship between STUDENT and CLASS has been dissolved into two one-to-many relations:

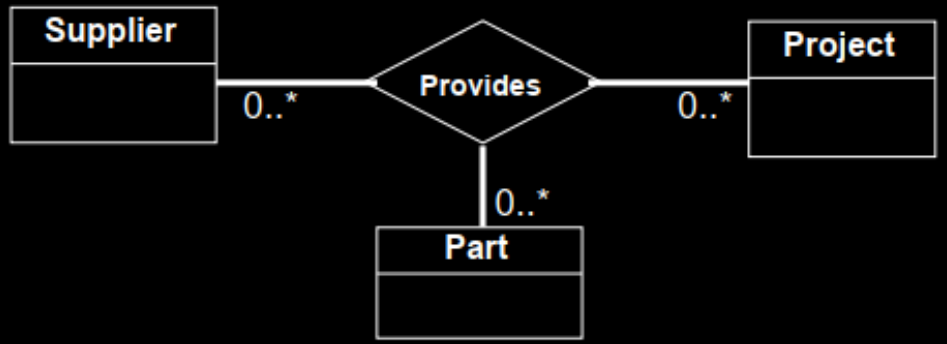
The 1:N relationship between STUDENT and STUDENT CLASSES reads this way: for one instance of STUDENT, there exists zero, one, or many instances of STUDENT CLASSES; but for one instance of STUDENT CLASSES, there exists zero or one instance of STUDENT.

The 1:N relationship between CLASS and STUDENT CLASSES reads this way: For one instance of CLASS, there exists zero, one, or many instances of STUDENT CLASSES; but for one instance of STUDENT CLASSES, there exists zero or one instance of CLASS.

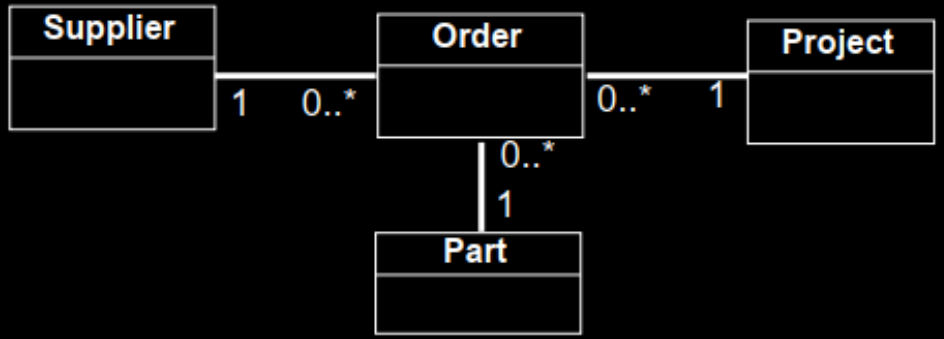


Higher Degree Relationship Handling Using Weak Entities

Original:



With weak entity:



Enhanced Entity Relationship Modelling

E-R Modeling Capabilities

Entity Relationship Models are normally adequate for building data models of traditional, administrative based database systems such as

Stock control

Product ordering

Customer invoicing.

Evolution of Databases

Designers of database applications have tried to design more accurate database schemas that reflect the data properties and constraints more precisely.

This was particularly important for newer applications of database technology, such as databases for

- Engineering design and manufacturing
- Telecommunications
- Complex software systems
- Geographic information systems (GISs)

Enhanced Entity-Relationship Modeling

Enhanced Entity-Relationship (EER) modeling is an extension of ER modeling to include object-oriented concepts such as:

- ➔ superclasses and subclasses
- ➔ specialization and generalization
- ➔ aggregation and composition

Superclasses & Subclasses

A **superclass** is a general class that is extended by one or more subclasses.

A **subclass** is a more specific class that extends a superclass by inheriting its methods and attributes and then adding its own methods and attributes.

Inheritance is the process of a subclass inheriting all the methods and attributes of a superclass.

Subclasses

Entity type has numerous subgroupings or subtypes of its entities that are meaningful and need to be represented explicitly because of their significance to the database application

EMPLOYEE entity type may be distinguished further into

- Secretary, Engineer, Technician
- Manage, Director
- Salaried employee, Hourly employee

Why Complicate Everything?

Hans Rosling