CSC 343 - Introduction to Databases

Tutorial #3:

Mapping E/R Diagram to Relational Schema

by Solmaz Kolahi

Today, we are focusing on the second step of database designing process:

Ideas → E/R Design → Relational Schema → Relational DBMS

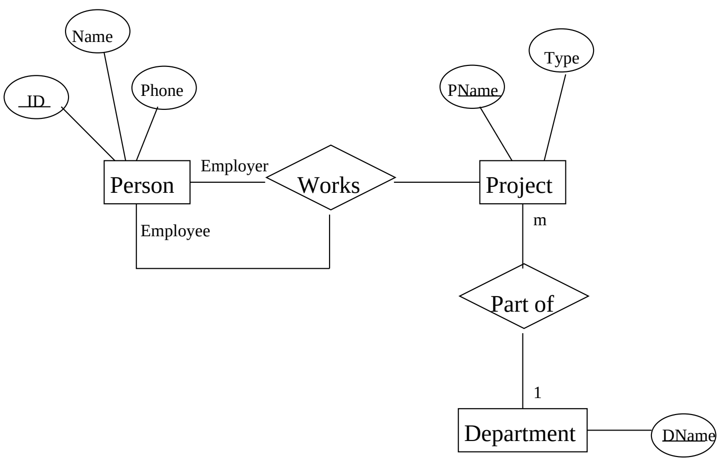
General Rules

- One relation for each entity set with the same attributes
- One relation for each relationship, attributes are the keys for the connected entity sets as well as the relationship attributes.

Special cases:

1. Sometimes we combine two relations.
2. Weak entity sets.
3. "Isa" relationship

Example:



Relations:
$Person(\underline{ID}, Name, Phone)$
$Project(\underline{PName}, Type, DName)$
$Department(\underline{DName})$
$Works(EmployerID, EmployeeID, PName)$

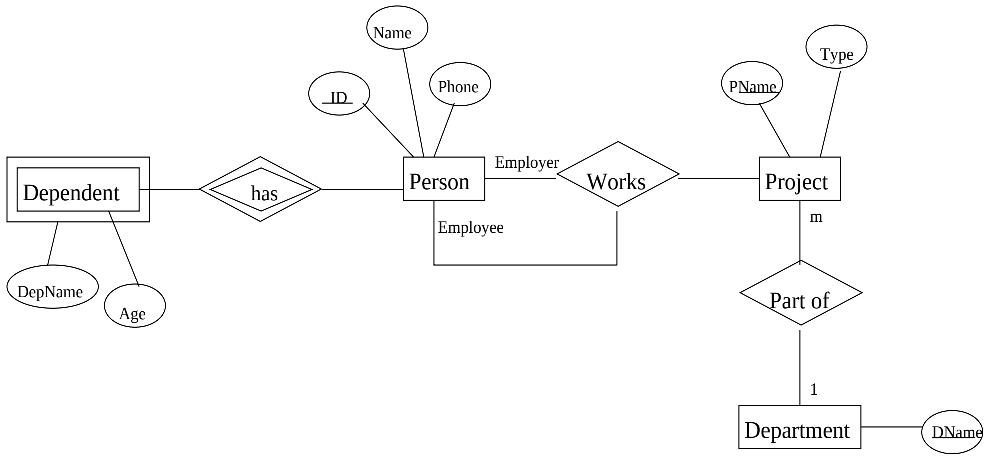
Notes: There is a many to one relationship between *Project* and Department entity sets. In this case, we can just add *Department* key to *Project* relation and do not make a separate relation for *Partof*. This is an example of first special case.

Weak Entities

Remember three things:

1. The relation for the weak entity, should include the keys of other entities which form the key for weak entity, plus other attributes of the weak entity.
2. The relation corresponding the weak entity's relationships should also the keys of other entities which form the key for weak entity.
3. There is no need to make a relation for the relationships between weak entity and other entities which form the key for weak entity.

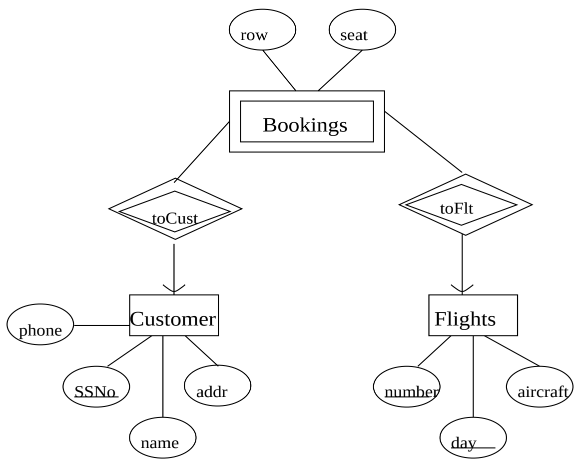
Example:



Relations:
$Dependent(\underline{Depname, ID}, Age)$

The weak relationship *has* does not have a corresponding relation explicitly.

Exercise 3.2.1



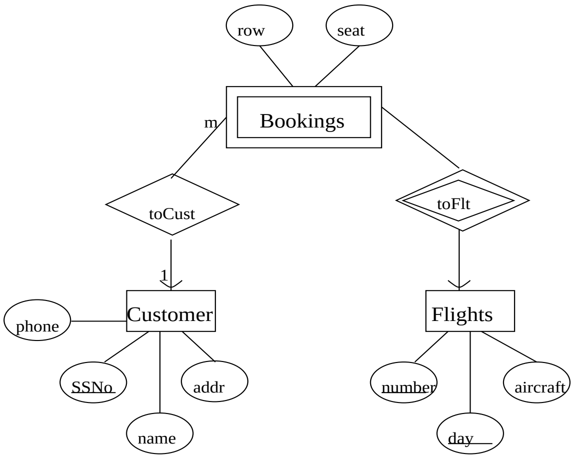
$Customers(ssNo, name, address, phone)$
$Flights(number, day, aircraft)$
$Bookings(ssNo, number, day, row, seat)$

Being a weak entity set, Bookings' relation has the keys for Customers and Flights and Bookings' own attributes. Notice that the relations obtained from the toCust and toFlt relationships are unnecessary. They are:
$toCust(ssNo, ssNo1, number, day)$
$toFlt(ssNo, number, day, number1, day1)$

That is, for toCust, the key of Customers is paired with the key for Bookings. Since both include ssNo, this attribute is repeated with two different names, ssNo and ssNo1. A similar situation exists for toFlt.

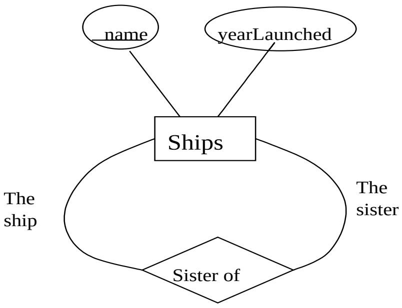
Exercise 3.2.2

There is another E/R diagram that could describe the weak entity set *Bookings* in previous exercise. Notice that a booking can be identified uniquely by the flight number, day of the flight, the row, and the seat; the customer is not then necessary to help identify the booking. Revise the diagram, and convert it into relations.



We first remove ssNo from Bookings. But since toCust is a many to one relationship, we add ssNo to Bookings instead of making a new relation for toCust. So the database would be the same as previous exercise.
$Customers(ssNo, name, address, phone)$
$Flights(number, day, aircraft)$
$Bookings(ssNo, number, day, row, seat)$

Exercise 3.2.3



Relations:
$Ships(name, yearLaunched)$
$SisterOf(name, sisterName)$

Isa Relationship

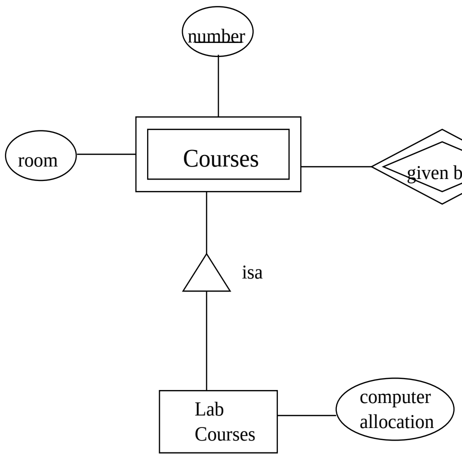
Three approaches:

- E/R-Style conversion: Create a relationship for each entity. If entity set E is not at the root of hierarchy, then the relation for E will include the key attributes at the root, plus all the attributes of E. We use these attributes to identify entities of E in the relation corresponding to E's relationship with others.

- Object-Oriented approach: Enumerate all the possible subtrees of the hierarchy. For each, create one relation that represent entities that have components in exactly those subtrees; the schema for this relation has all the attributes of any entity set in the subtree.

- Using Null Values: We can handle a hierarchy of entity sets with a single relation. This relation has all the attributes belonging to any entity set of the hierarchy. An entity is then represented by a single tuple. This tuple has NULL in each attribute that is not defined for that entity.

Exercise 3.3.1



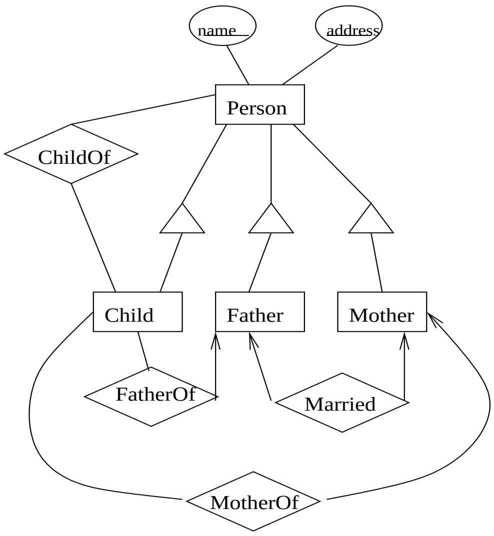
(a) E/R-Style conversion: Since Courses is weak, its key is number and the name of its department. We do not have a relation for GivenBy. In part (a), there is a relation for Courses and a relation for LabCourses that has only the key and the computer-allocation attribute. It looks like:
$Depts(name, chair)$
$Courses(number, deptName, room)$
$LabCourses(number, deptName, allocation)$

(b) Object-Oriented approach: LabCourses gets all the attributes of Courses, as:
$Depts(name, chair)$

$Courses(number, deptName, room)$
$LabCourses(number, deptName, room, allocation)$

(c) Using Null Values: Courses and LabCourses are combined, as:
$Depts(name, chair)$
$Courses(number, deptName, room, allocation)$

Exercise 3.3.2



E/R-Style conversion:
$Person(name, address)$
$Child(name, address, father\_name, father\_address, mother\_name, mother\_address)$
$Mother(name, address)$
$Father(name, address, mother\_name, mother\_address)$
$ChildOf(child\_name, child\_address, parent\_name, parent\_address)$

When defining the relations in SQL, referential integrity constraints should be enforced by defining foreign keys.