# CPSC 333: More About Relationships



This material was not covered in lectures in Winter, 1997, due to lack of time. However, it is ``required reading'' for CPSC 333 students.



## Types of Binary Relationships

Almost all relationships that appear on an entity-relationship diagram are *binary relationships* - they connect exactly two entities together (or connect a single entity to itself). It is useful to determine the ``cardinality'' and ``conditionality'' of binary relationships, in order to define keys and to map the entity-relationship diagram to a set of data tables (or to implement the data model in some other way).

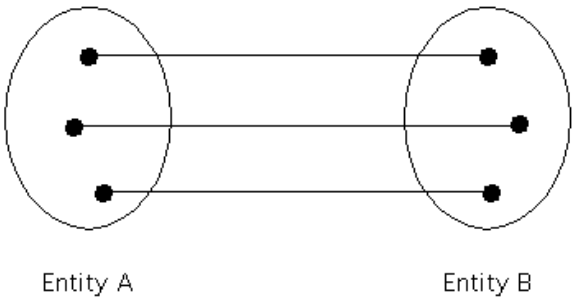
For the rest of this section, suppose a binary relationship connects two entities, called ``Entity *A*'' and ``Entity *B*.''

## Cardinality

Three types of ``cardinalities'' of binary relationships are common.

### One-to-One Relationships

A *one-to-one relationship* (``1:1'') (between entities *A* and *B*) has the property that every instance of *A* is mapped by the relationship to *exactly one instance* of *B*, and vice-versa:

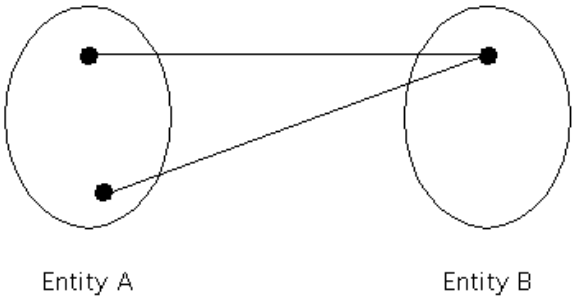



**1 : 1**

A description of this picture is also available.

**One-to-Many Relationships**

A *one-to-many relationship* (``1:M,''- also called a ``many-to-one relationship'') has the property that every instance of *A* is mapped to *exactly one* instance of *B*, while every instance of *B* is mapped to *one or more* (``many'') instances of *A* - or vice-versa (that is, we could exchange the roles of *A* and *B*, without changing the type of the relationship that connects them):

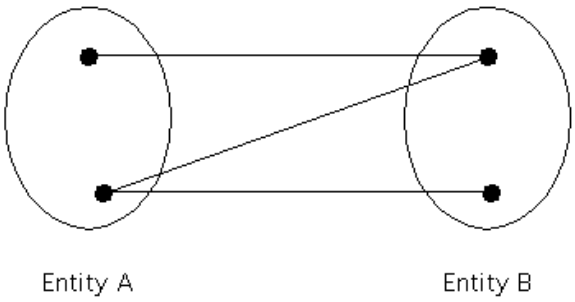



**1 : M**

A description of this picture is also available.

**Many-to-Many Relationships**

A *many-to-many relationship* (``M:M'') has the property that every instance of *A* is mapped to *one or more* instances of *B* and vice-versa:




**M : M**

A description of this picture is also available.

## Conditionality

### Unconditional Relationships

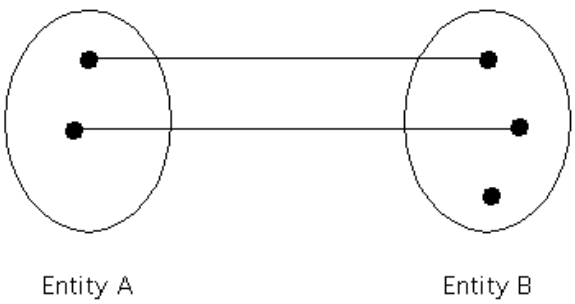
All three of the types of relationships that have been described above (that is, one-to-one relationships, one-to-many relationships, and many-to-many relationships) are types of *unconditional* relationships: each instance of *A* is connected to *at least one* instance of *B*, and vice-versa.

### Conditional Relationships

A relationship is *conditional* if each instance of *A* must be connected to at least one instance of *B*, but instances of *B* don't need to be connected to any instances of *A* at all - or vice-versa.

It turns out that (when we consider cardinality as well as conditionality) exactly *four* different kinds of conditional binary relationships can be defined.

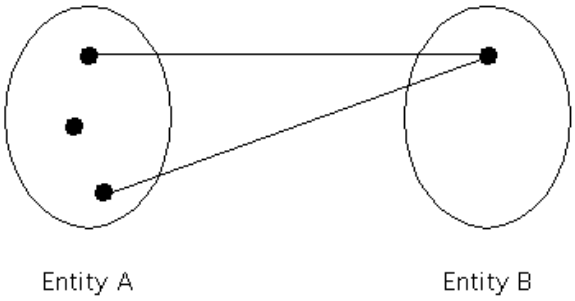
A relationship has type ``1:1c'' if each instance of *A* must be connected to *exactly one* instance of *B* and each instance of *B* is connected to *either zero or one* instance(s) of *A* (or vice-versa):




**1 : 1c**

A description of this picture is also available.

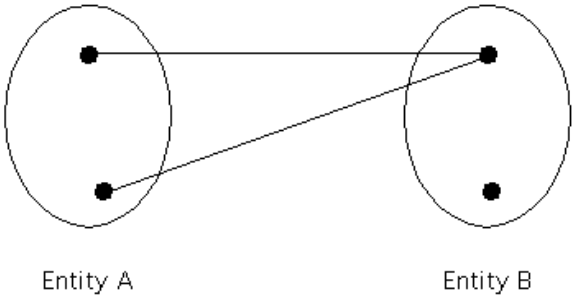
A relationship has type ``1c:M'' if each instance of *A* is connected to *either zero or one* instance(s) of *B*, while each instance of *B* must be connected to *one or more* instances of *A* (or, vice-versa):




**1c : M**

A description of this picture is also available.

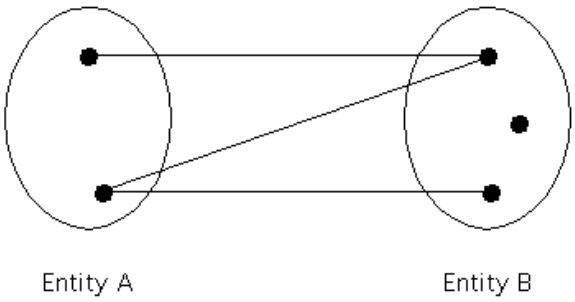
A relationship has type ``1:Mc'' if each instance of *A* is connected to *exactly one* instance of *B* while each instance of *B* is connected to *zero or more* instances of *A* (or, vice-versa):




**1 : Mc**

A description of this picture is also available. Note that you generally *can't* exchange the roles of *A* and *B* in a relationship of type ``1:Mc'' to get a relationship of type ``1c:M'' (or vice-versa).

Finally, a relationship has type ``M:Mc'' if each instance of *A* is connected to *one or more* instance(s) of *B* and each instance of *B* is connected to *zero or more* instance of *A* (or vice-versa):




**M : Mc**

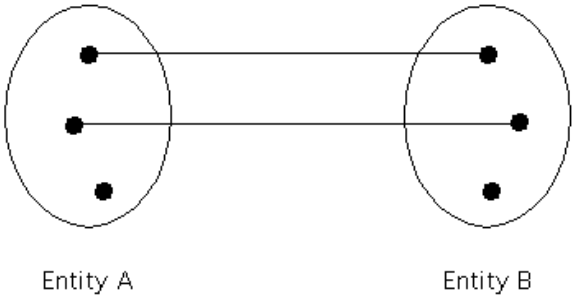
A description of this picture is also available.

### Biconditional Relationships

A relationship is *biconditional* if it isn't necessary to connect an instance of *A* to any instances of *B* at all, *and* it isn't necessary to connect an instance of *B* to any instances of *A* at all, either.

When cardinality is also considered, three types of biconditional relationships can be defined.

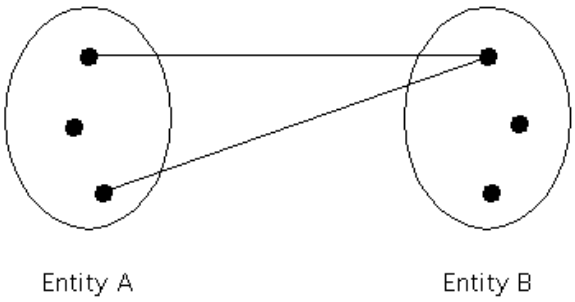
A relationship has type ``1c:1c'' if each instance of *A* is mapped to *either zero or one* instance of *B* and vice-versa:




**1c : 1c**

A description of this picture is also available.

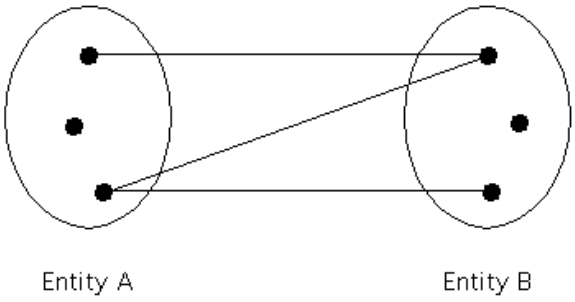
A relationship has type ``1c:Mc'' if each instance of *A* is connected to *either zero or one* instance of *B* while each instance of *B* is connected to *zero or more* instances of *A* (or vice-versa):




**1c : Mc**

A description of this picture is also available.

Finally, a relationship has type ``Mc:Mc'' if each instance of *A* is connected to *zero or more* of instances of *B*, and vice-versa:




**Mc : Mc**

A description of this picture is available as well.

## Choosing Primary Keys for Relationships

Suppose now that we wish to choose ``primary keys'' for relationships as well as entities.

If the type of a binary relationship is ``1:1,'' ``1:1c,'' ``1c:1c,'' ``1:M,'' ``1c:M,'' ``1:Mc,'' or ``1c:Mc,'' then at least one of the two entities connected by the relationship (without loss of generality, ``Entity A'') has the property that each instance of that entity is connected to *at most one* instance of the other. In this case, the primary key for this relationship (assumed above to be ``Entity A'') can also be used as the ``primary key'' for the relationship.

Otherwise, the type of the binary relationship is one of ``M:M,'' ``M:Mc,'' or ``Mc:Mc,'' and it is necessary to include all the attributes in the primary keys of *both* of ``Entity A'' and ``Entity B'' in order to form a primary key for the relationship.

## Generalization to *k*-Way Relationships

While this classification (and set of rules for choosing primary keys) can be generalized to three-way relationships, and so on, it all gets very complicated, very quickly. In general, when describing these relationships, try to make it clear how may instances of the other entities each instance of every given entity can be related to, and try to choose attributes from *as few* of the related entities' primary keys, when choosing a primary key for a relationship, as is possible.

Location: [CPSC 333] [Listing by Topic] [Listing by Date] [Previous Topic] [Next Topic] More About Relationships

*Department of Computer Science*
*University of Calgary*

*Office: (403) 220-5073*
*Fax: (403) 284-4707*

*eberly@cpsc.ucalgary.ca*