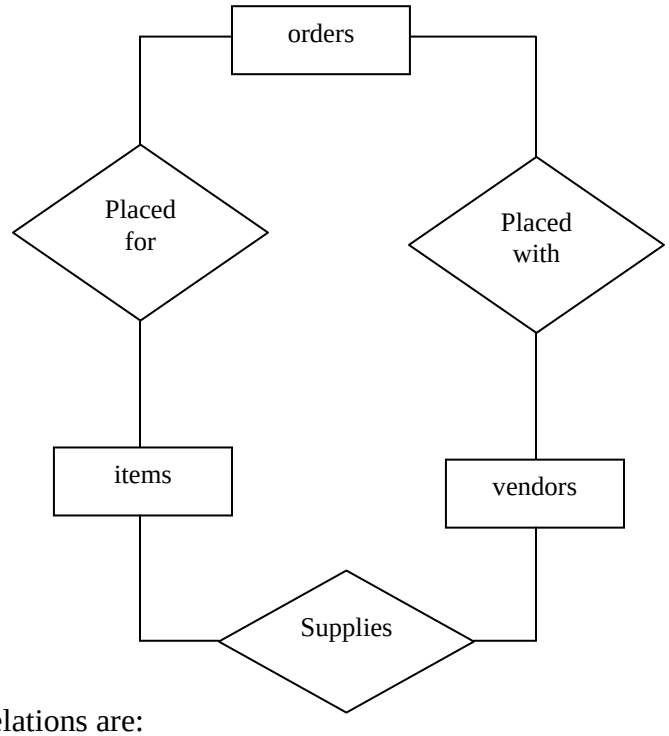
Mini Case Example 1

We will illustrate the method used to obtain an E-R diagram with an example. Usually an imprecise statement of the needs of an organization is given by the management. We now give a sample of such a statement. This is to be used to obtain an E-R diagram.

User's narrative of requirements: "Our company receives a number of items from many vendors and they are received at the receiving office. As we receive over 1000 items a day it is now virtually impossible for the receiving office to check whether the vendor has supplied items against an order, or sent a wrong item and inform the purchase office. We are also not able to find out if there are excesses or deficiencies in delivery and whether the vendor adhered to the delivery schedule as per the order. The items received at the receiving office are sent for physical inspection. The physical inspection consists of checking whether the quantities stated in the delivery note agree with the physical count, whether the item is the correct one ordered, and a check on the quality of item. We would like to keep a record of rejections due to bad quality, incorrect items, excess/deficient supply etc., determined during inspection. This will enable us to assess vendors' reliability to guide us in placing orders in the future, besides keeping track of supplies. Items cleared by the inspection office are taken into the inventory by the stores office which keeps a ledger of items stocked and quantity available of each item. Customers send requisitions to the stores. The stores fulfill the requests based on availability and update the ledger. Currently we are not able to meet some of our customers' requests. We would like to incorporate automatic reordering by our purchase office if the inventory level of an item.

We would also like to keep track of unfulfilled requests and meet them when items reach the store. Currently we are not able to pay our vendors promptly due to delays in payment order reaching our accounts office. We would like to rectify this. We would also like to bill our customers promptly and keep track of customers' payments".

The E-R diagram for the problem is:



The relations are:

ORDER (**order no.**., order date)

ORDER PLACED FOR (**order no**., **item code**, qty. ordered, delivery time allowed)

ORDER PLACEDWITH (**order no.**. **vendor code**, **item code**)

VENDOR (**vendor code**, vendor name, vendor address)

ITEM (**item code**, item name, price/unit)

SUPPLIES (**vendor code, item code, order no**., qty.supplied, date of supply).

The key attribute(s) are in bold letter(s) in each relation.

Let us examine whether the relations are in normal form. ORDERS and ORDER PLACED FOR are simple relations. In the relation ORDER PLACED WITH, the key is the composite attributes *order no.*, *vendor code* and *item code.* However, the entity *item code* is not needed in the ORDER PLACED WITH relation. Given an **order no**., all the items supplied against this order can be found from ORDER PLACED FOR relation. The vendor with whom the order has been placed can be found from the ORDER PLACED WITH relation given below as each order no., has only one vendor.

ORDER PLACED WITH (**order no**., **vendor code**)

The two relations ORDER PLACED WITH and ORDER PLACED FOR have composite keys. The non-key fields are not related to one another. In a key more than one attribute, the individual attributes are not functionally dependent. Thus these two relations are in normalized form and do not need any further change. The relations VENDOR and ITEM are simple and are in normalized form. The relation SUPPLIES is, however, not normalized. **vendor code** and **order no**., are functionally dependent. There is a multivalued dependency between vendor code and item code as a vendor can supply many items. We thus split the relations into two relations.

ACTUAL VENDOR SUPPLY (**order no**., **item code**, qty.supplied, date of supply)

VENDOR SUPPLY CAPABILITY (**vendor code, item code**)

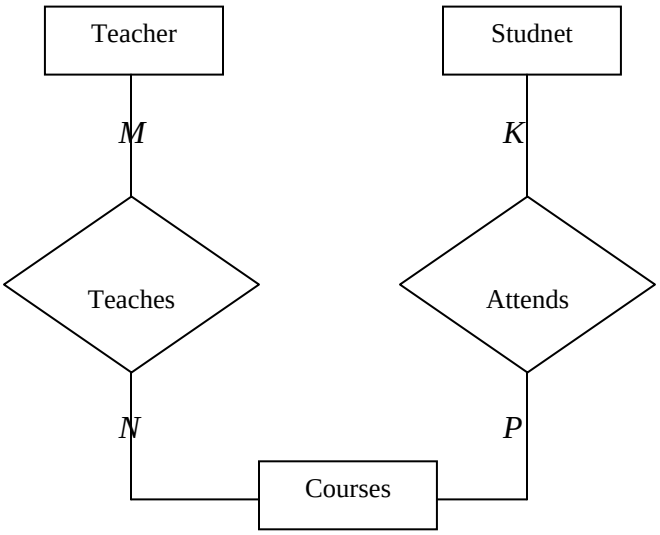
Observe that the relation VENDOR SUPPLY CAPABILITY will have a (**vendor code**, **item code**) table without a vendor having supplied any item. The relation ORDER PLACED WITH will have a tuple only when a vendor actually supplies an item.

Minicase Example 2

Let a database contain the following: Teacher code, Teacher's name, Teacher's address, rank, department, courses taught by the teacher, course name, credits for course, no.of students in the class, course taught in semester no., student no., name, dept., year, and courses taken in semesters no. The following information is given on dependencies.

- A teacher may teach more than one course in a semester
- A teacher is affiliated to only one department
- A student may take many courses in a semester
- The same course may have more than one section and different sections will be taught by different teachers.
- A course may be taught in more than one semester

An entity relationship diagram for this problem is given below. The relations corresponding to the E-R diagram are:



An E-R diagram for teacher database

TEACHER (**Teacher code**, teacher's name, teacher's address, rank, dept.)

TEACHES COURSES (**Teacher code, course no**., **semester no.,** no.of students, section no)

COURSE (**course no., semester no.,** course name, credits)

STUDENT (**student no**., student's name, dept., year)

STUDENT –COURSES (**student's no**., **course no**., **semester no**.)

TEACHER relation has only one key. All non-key attributes are functionally dependent only on the key. There is no functional dependency among non-key attributes. Thus the relation is normalized in 3NF (No higher NFs are applicable).

STUDENT relation is also, similarly, in 3NF. In the COURSE relation, course name could also be a key. The relation is in 3NF and no further normalization is required. The relations TEACHES COURSES and STUDENT-COURSES have multiattribute keys, but

the relations themselves are in normal form. The only point which is not clear, from these relations, is the relation between teacher and student. This has been missed in the E-R diagram. The relationship is between the teacher, courses taught and students. In other words, we should be able to answer the question “which teacher is teaching course no.X to student no. Y in Semester 2?” Let us add a relation.

TEACHER-STUDENT (**Teacher code, student no**., **course no**.)

In this relation **Teacher code** and **course no.** have a multivalued dependency. Similarly, **Teacher code** and **student no.** as well as **student no.** and **course no.** have multivalued dependency. However, TEACHES COURSES (**Teacher code, course no**., **semester no.**, no.of students, section no.) and STUDENT COURSES (**student no**., **course no**., **semester no**) relations are already in the database. Thus the relation TEACHER-STUDENT as it is specified above is sufficient to give the idea that student Y takes course X from Teacher Z.