DATABASE

E-R MODELLING

DATABASE
E-R MODELLING

DATABASE → an organised collection of structured information, or data, stored in a computer system

→ one of the main aspects of database is the **organisation of data** in it

→ to understand the organisation of data, you need to know a **modelling approach**

→ the modelling approach you are going to study is **entity relationship (E-R) modelling**

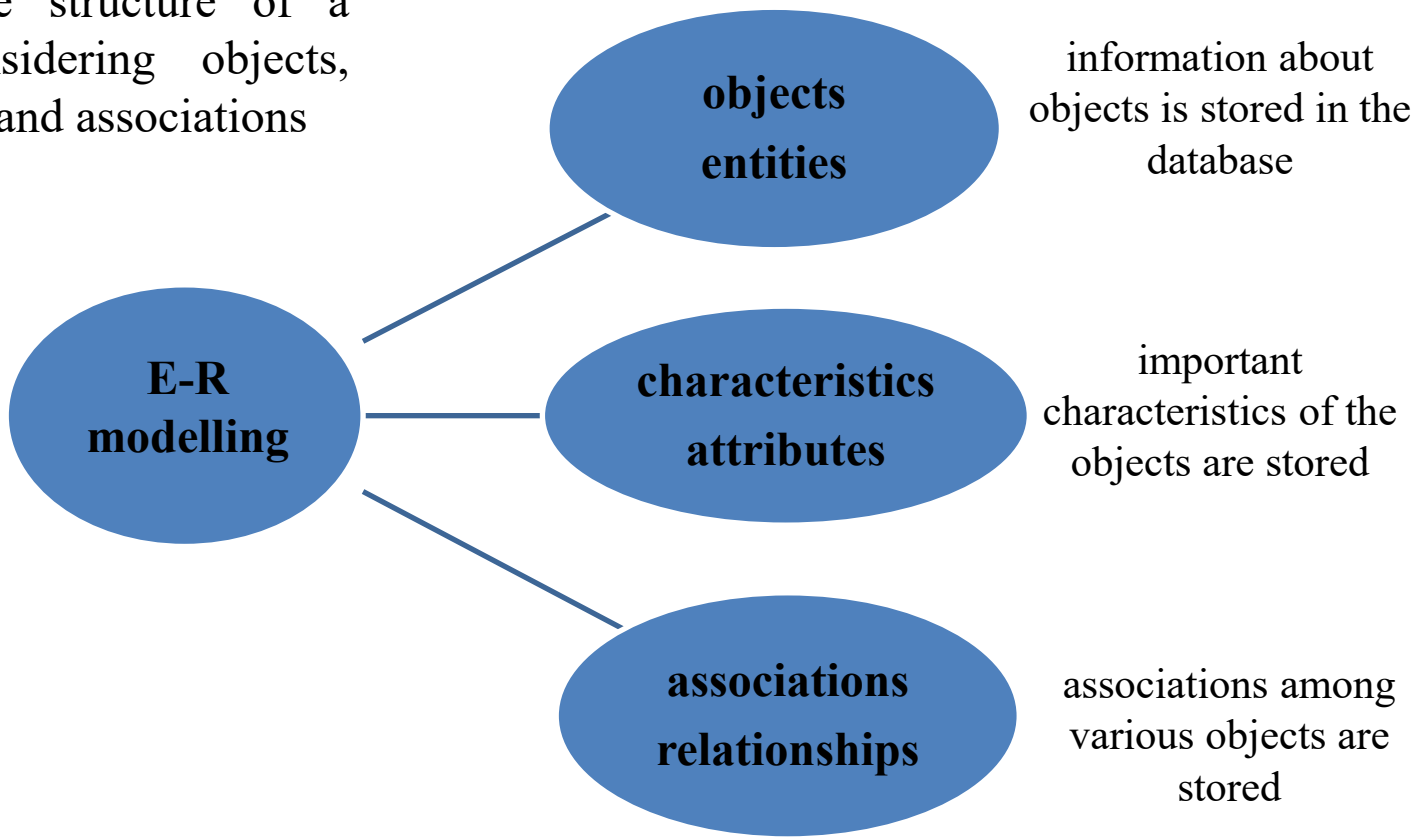
→ E-R modelling is an approach to **semantic modelling**

→ semantic modelling tries to **understand and represent meaning**

→ E-R modelling was introduced by Peter **Chen** in **1976**

DATABASE
E-R MODELLING

E-R modelling consists of a number of activities to help understand the structure of a database, considering objects, characteristics and associations



DATABASE
E-R MODELLING

The result of E-R modelling is the E-R Diagram (ERD)

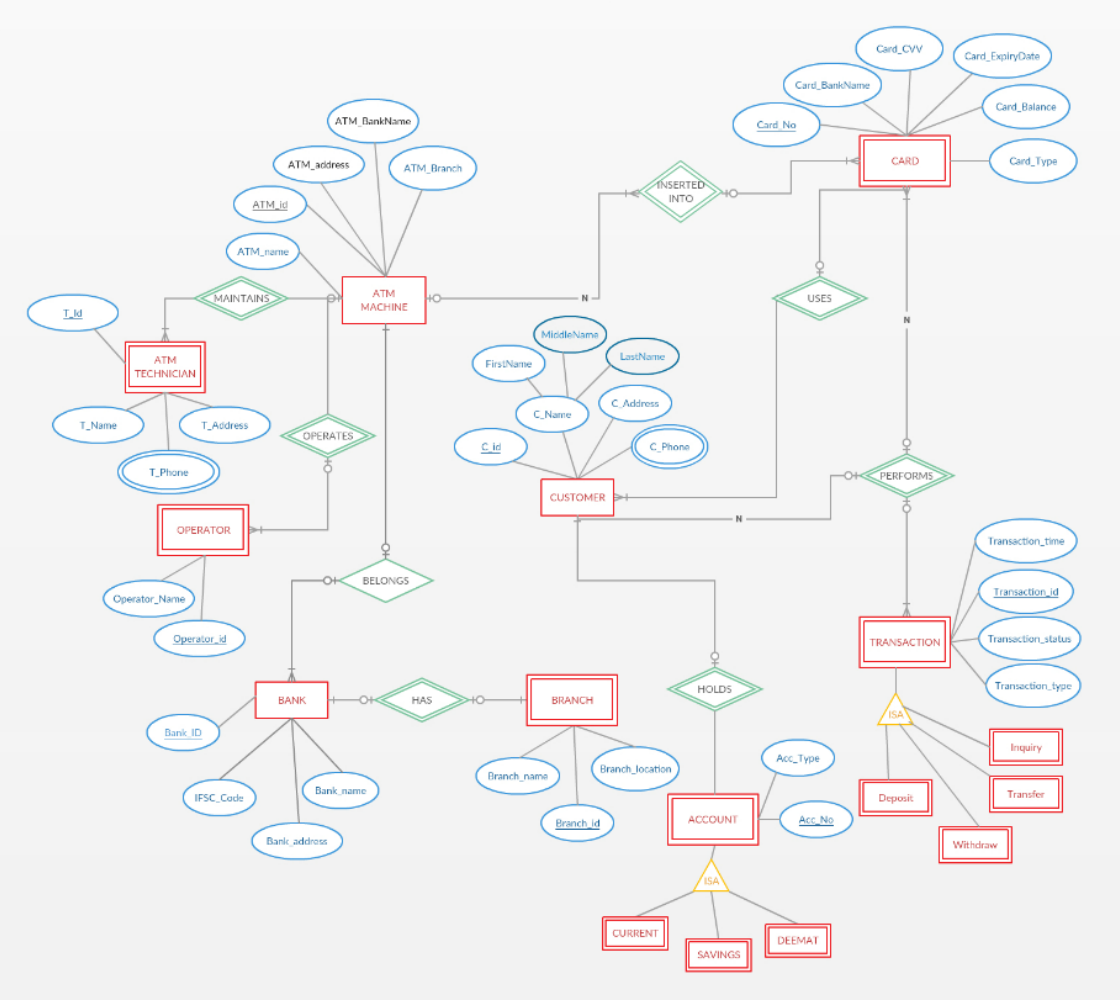
ERD is a graphical representation

ERD shows the logical structure of the database

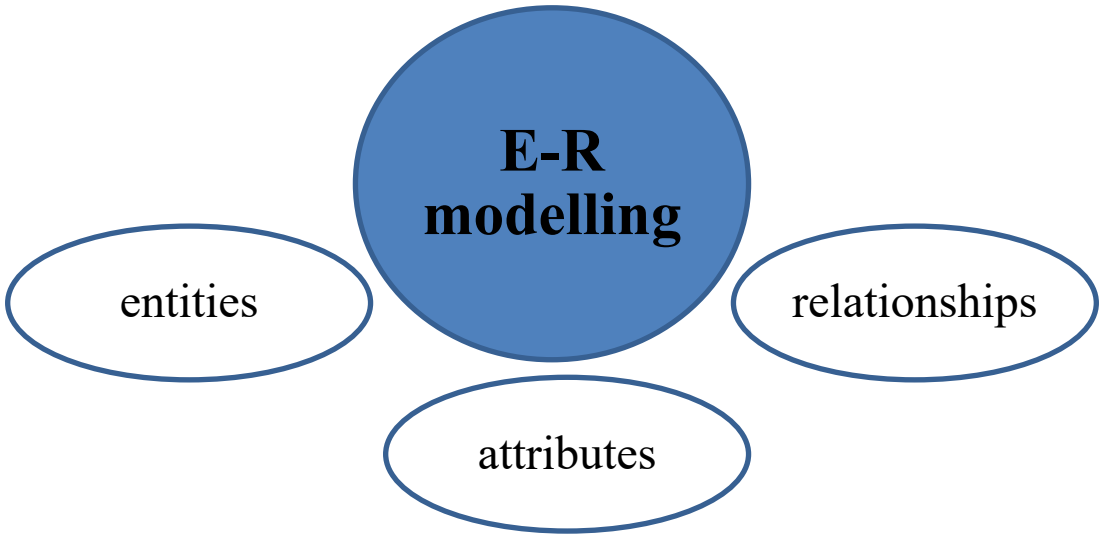
The ERD is an instrument and it serves several purposes:

- it allows to understand the information contained in the database
- it is a documentation tool
- it communicates the logical structures of the database to users

DATABASE
E-R MODELLING



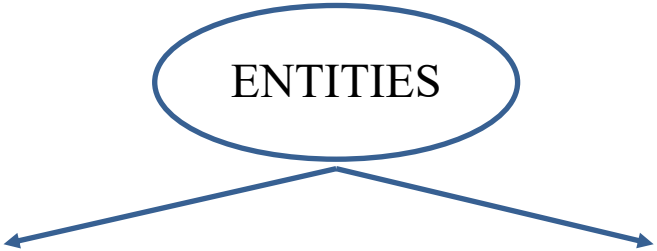
DATABASE
E-R MODELLING



The E-R modelling process identifies three basic elements:

- entities
- attributes
- relationships

DATABASE
E-R MODELLING



"a thing that can be easily identified"

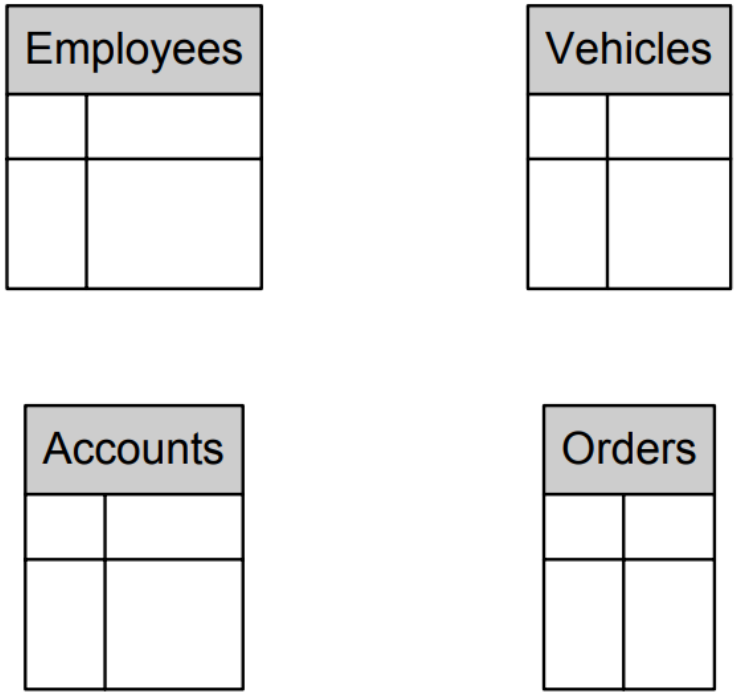
the things about which the database stores information:
- tangible items (equipment)
- concepts (accounts)
- people (employees)
- events (sales)
- places (business locations)

→ **ENTITY TYPE** refers to the general definition of that object (*employee*)

→ **ENTITY INSTANCE** refers to a single occurrence of an entity type (*employee 123-45-6789*)

DATABASE
E-R MODELLING

In the E-R Diagram, entities are represented by rectangles



DATABASE
E-R MODELLING

ATTRIBUTES →

a single data value that describes a characteristic of an entity

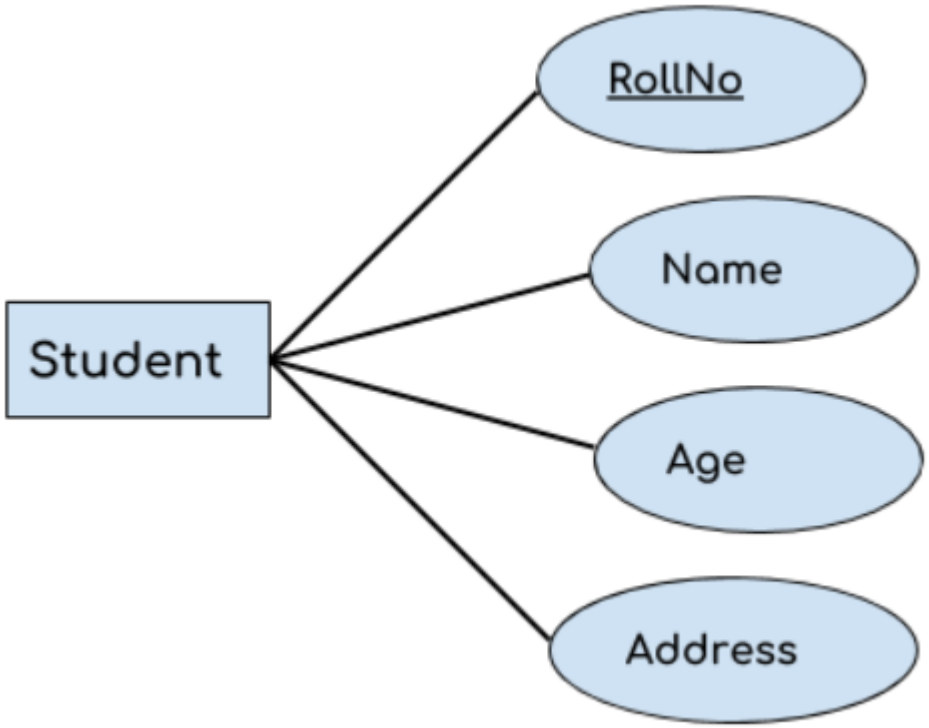
a set of characteristics associated to the entity that define the entity itself

each entity has a corresponding set of attributes that represent the information about the entity →

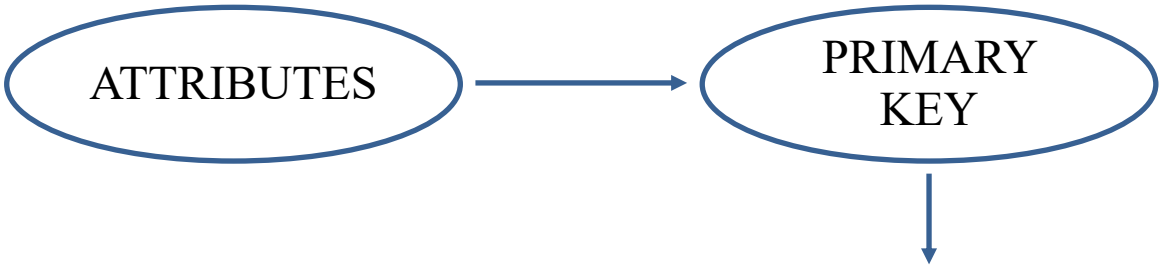
a university may wish to know the name, address, phone number of each student. If STUDENT is the entity put in the database, NAME, ADDRESS, PHONE NUMBER are the attributes of that entity

DATABASE
E-R MODELLING

In the E-R Diagram, attributes are represented by ellipses



DATABASE
E-R MODELLING



an attribute or combination of attributes that uniquely identifies an instance of the entity

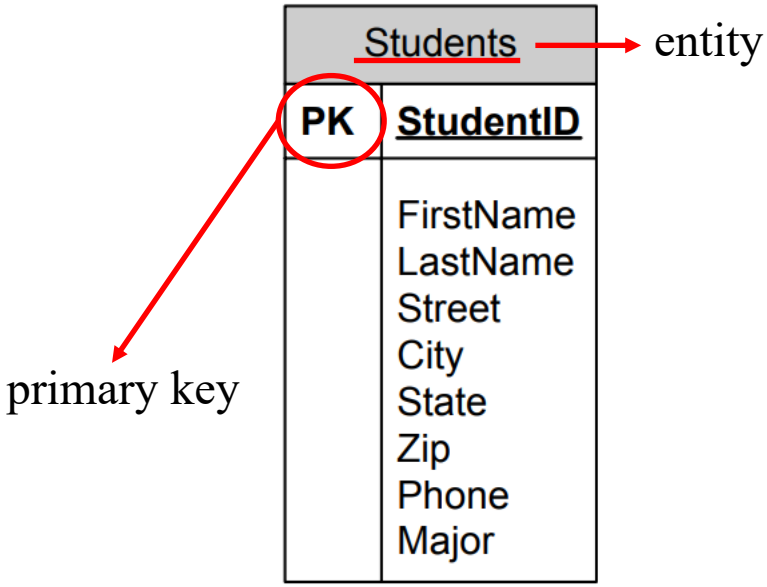
no two instances of an entity may have the same value for primary key

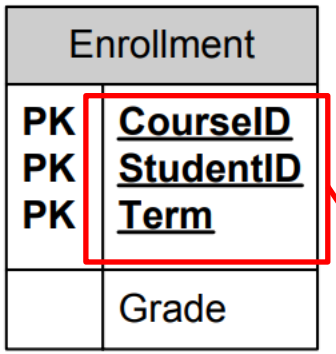
sometimes more than one attribute is used to form a primary key: this is a **composite key** or **compound key**

in this case, it is the **combination** of values for all attributes that must be unique, not the single attribute

DATABASE
E-R MODELLING

For example, the entity ENROLLMENT has a composite primary key comprised of the attributes STUDENT_ID and COURSE_ID. Each instance of ENROLLMENT must contain a unique combination of values for StudentID and CourseID. However, there can be duplications of StudentID or CourseID. So, it is possible for many instances of ENROLLMENT to have the value MIS100 for CourseID, but each of those instances must contain different values for StudentID.



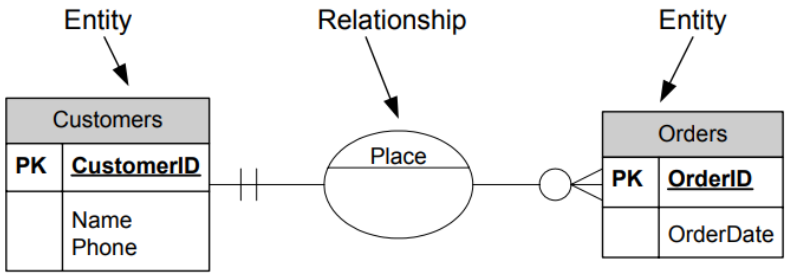


In the E-R Diagram, primary keys are underlined

DATABASE
E-R MODELLING

RELATIONSHIPS → the associations between two or more entities

in the E-R Diagram, they are represented as an ellipse connected by lines to the related entities



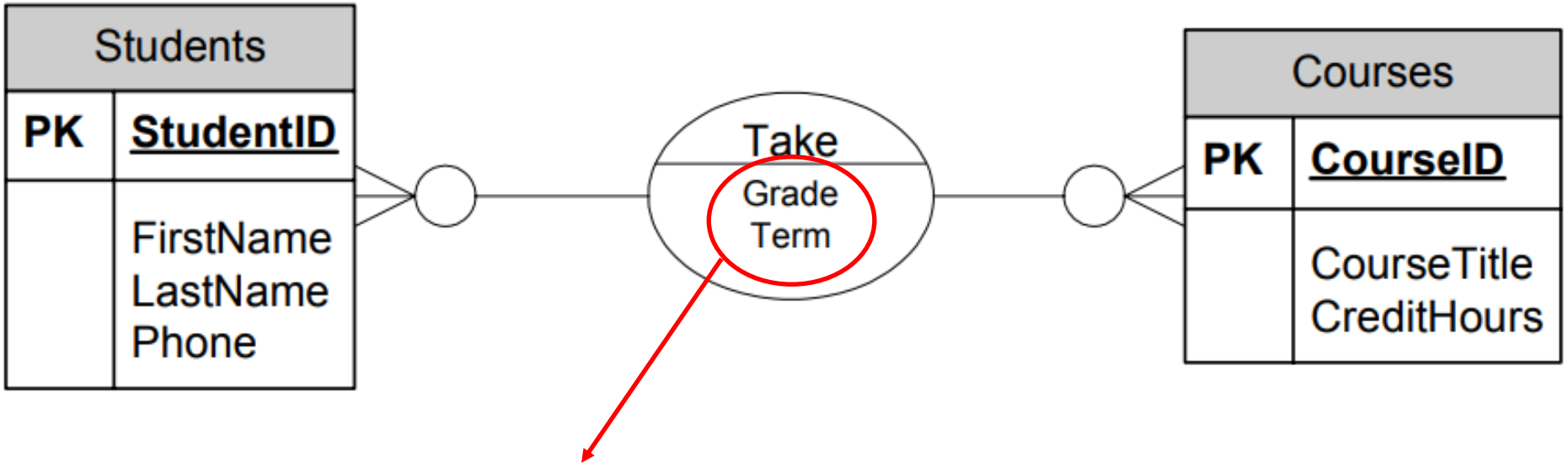
it can be defined according to two criteria:
- **degree**
- **cardinality**

DATABASE
E-R MODELLING

It may happen that ATTRIBUTES are attached to RELATIONSHIPS and not to ENTITIES

In some cases, attributes may be attached to a relationship, rather than an entity. For example, a GRADE attribute is a function of the combination of STUDENTS and COURSES, but is not strictly a function of either entity by itself. Attaching GRADE to STUDENT would not indicate that a STUDENT has a GRADE for a particular COURSE, while attaching GRADE to COURSES doesn't show that the GRADE is for a particular STUDENT. Attaching the GRADE attribute to the relationship between COURSES and STUDENTS shows that a value of GRADE is dependent on an intersection of COURSES and STUDENTS. A similar argument can be made for TERM.

DATABASE
E-R MODELLING



GRADE and TERM are attributes that can be associated to the relationship between the entities STUDENTS and COURSE, and not to the entities themselves.

In the E-R Diagram, these attributes are written in the ellipse containing the relationship they refer to.

DATABASE
E-R MODELLING

DEGREE: the number of entities involved in a relationship

Unary relationship (recursive): one entity

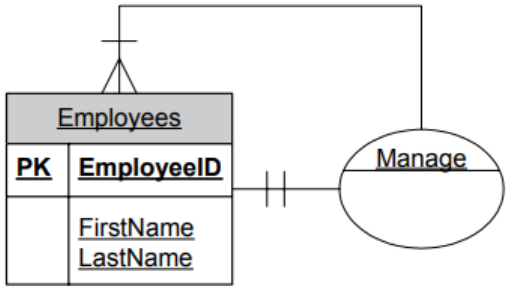
Binary relationship: two entities

Ternary relationship: three entities

N-ary relationship: n is the number of the entities

This is an example of the **unary** or **recursive relationship**.
It happens when one instance of an entity is related to another instance of the same entity.

In the E-R Diagram, unary or recursive relationship is represented by an ellipse connected to the entity twice

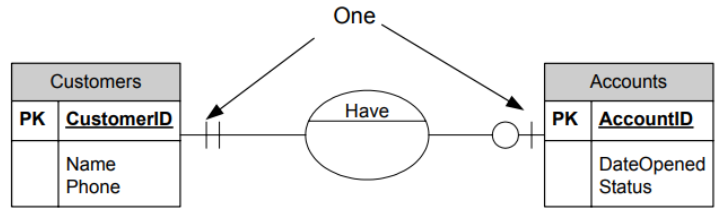


In this example, an EMPLOYEE may manage other EMPLOYEES, or may not manage any.

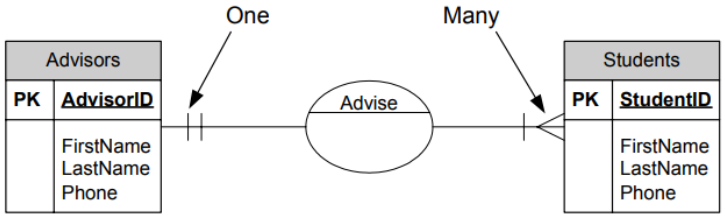
DATABASE
E-R MODELLING

Maximum cardinality: the maximum number of instances of one entity that can be associated with a single instance of a related entity

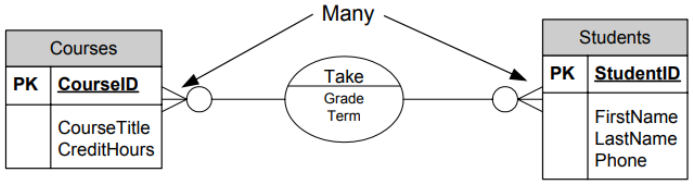
Minimum cardinality: the minimum number of instances of one entity that must be associated with a single instance of a related entity



one-to-one (1:1): one CUSTOMER can be related to only one ACCOUNT and one ACCOUNT can be related to only one CUSTOMER



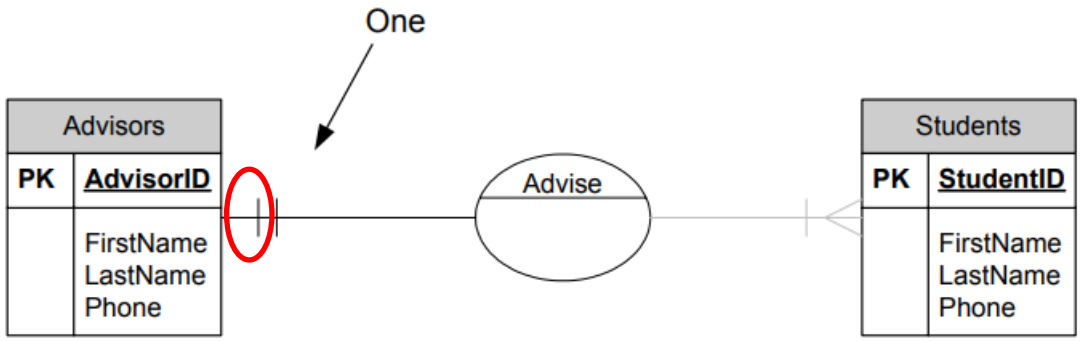
one-to-many (1:N): one ADVISOR can be related to one or more STUDENTS, but a STUDENT can be related to only a single advisor



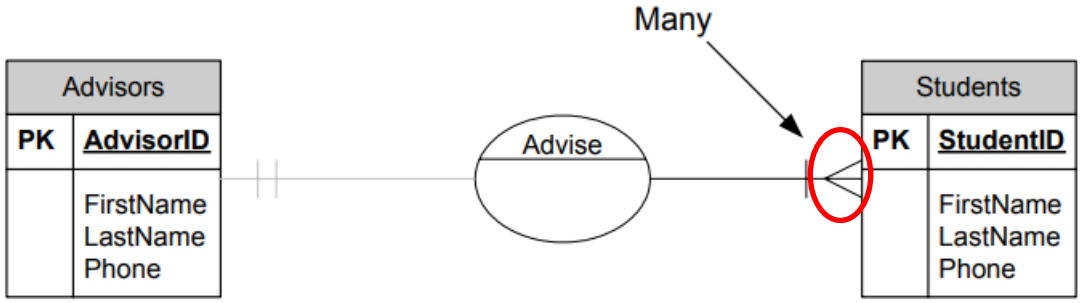
many-to-many (M:N): a single STUDENT can be related to zero or more COURSES and a single COURSE can be related to zero or more STUDENTS

DATABASE
E-R MODELLING

In the E-R Diagram, cardinality is represented by symbols attached to the relationship line



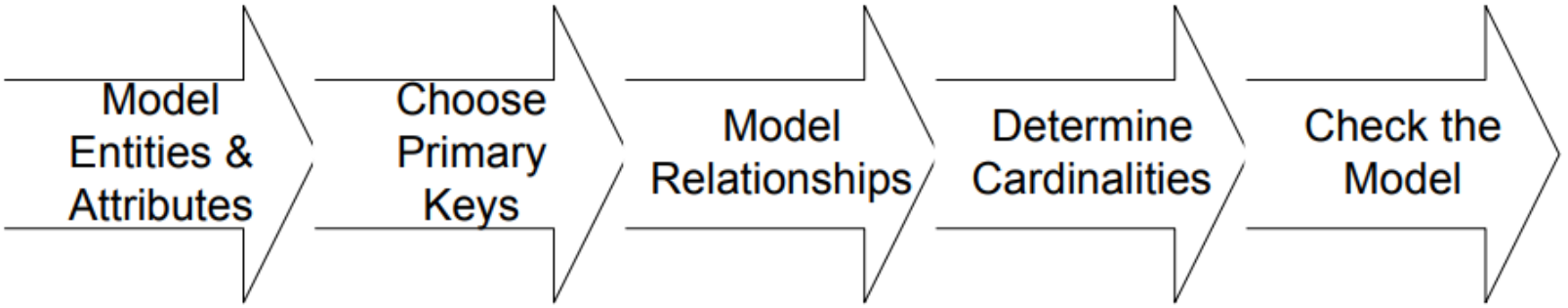
"One" cardinality is represented by a single vertical line



"Many" cardinality is represented by a crowfoot

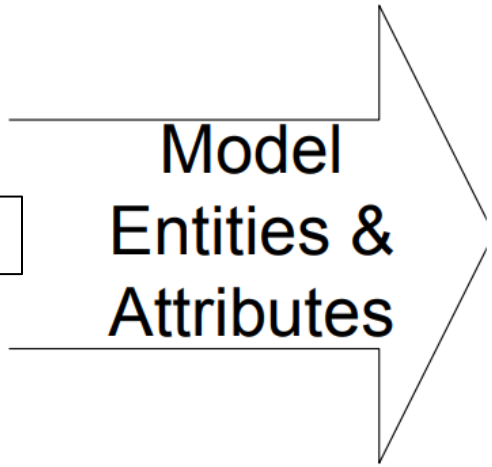
DATABASE
E-R MODELLING

To create an E-R diagram, some steps are required:



DATABASE
E-R MODELLING

STEP 1



The first thing to do when creating an E-R diagram is to identify the ENTITIES, which are something about which data need to be stored.

As well as ENTITIES, it is also important to define and identifies ATTRIBUTES and for PRIMARY KEYS too.

→ it is mostly important to distinguish ENTITIES from ATTRIBUTES

At this stage, you should include as many ENTITIES as you think are important.
The more the better, because in this way you can be more precise and accurate.
Moreover, it is easier to remove unnecessary ENTITIES later on than to add more ENTITIES at a further stage.

The best way to understand whether an ENTITY is really an ENTITY is to look at its ATTRIBUTES: if an ENTITY has no ATTRIBUTES, it means that it isn't an ENTITY, but possibly an ATTRIBUTE of another ENTITY.

DATABASE
E-R MODELLING

Document

-NO:	44-44-4444	-ID:	1002	
DATE:	10/31/98	**CUST-NAME:**	ABC Inc.	
ID	**DESCRIPTION**	**PRICE**	**QTY**	**EXT**
A123	STEREO SYSTEM	375.00	2	750.00
C235	8" SPEAKER	150.00	8	1,200.00
X002	SPEAKER WIRE	10.00	5	50.00
		TOTAL		2,000.00

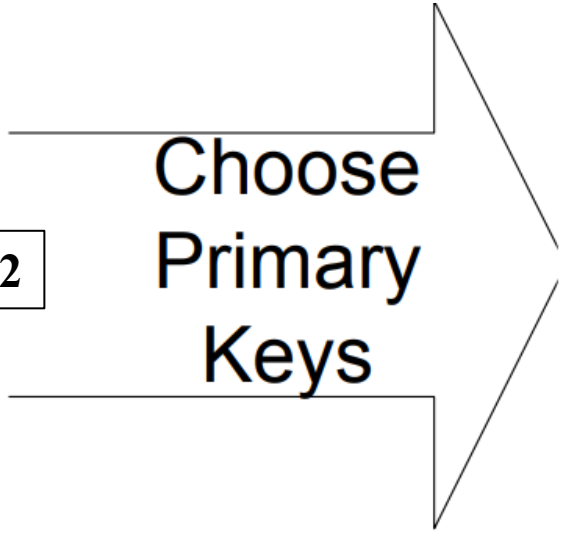
E-R diagram draft

Entity	Attributes
ORDER	ORDER-NO, DATE
PRODUCT	PROD-ID, DESCRIPTION, PRICE
CUSTOMER	CUSTOMER-ID, CUST-NAME

Having as an example an order entry form, those are the possible ENTITIES and ATTRIBUTES identified when creating the E-R diagram.

DATABASE
E-R MODELLING

STEP 2



PRIMARY KEYS must be chosen for each ENTITY.

For many ENTITIES, the PRIMARY KEY is obvious and well-defined. The only thing to do is to select it.

When PRIMARY KEYS aren't obvious, they must be identified. If they don't exist, a new ATTRIBUTE is to be created. The choice has to respect at least four requirements.

Desirable Primary Key Characteristics
1. Uniquely identifies an entity instance
2. Non-null (always has a value)
3. Data-less
4. Never changes

for each instance of the ENTITY the value of the ATTRIBUTE must be unique

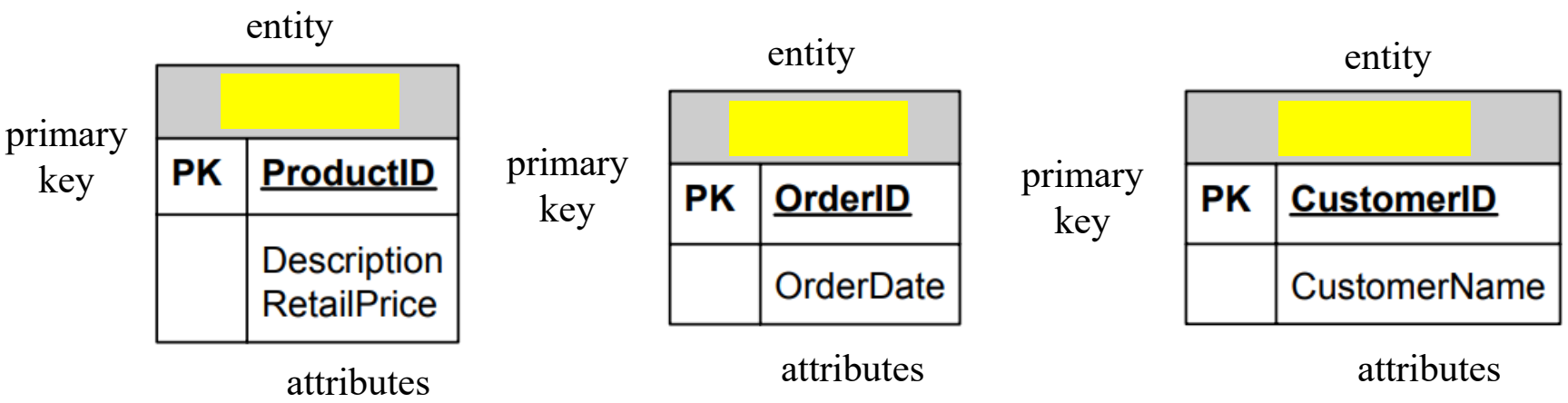
it must always have a unique, valid value for each instance of the ENTITY

it contains no useful information and its only function is to identify the ENTITY

DATABASE
E-R MODELLING

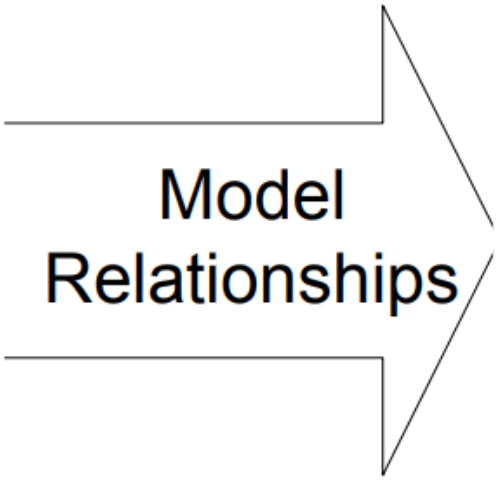
Document

-NO:	44-44-4444	-ID:	1002	
DATE:	10/31/98	**CUST-NAME:**	ABC Inc.	
ID	**DESCRIPTION**	**PRICE**	**QTY**	**EXT**
A123	STEREO SYSTEM	375.00	2	750.00
C235	8" SPEAKER	150.00	8	1,200.00
X002	SPEAKER WIRE	10.00	5	50.00
		TOTAL		2,000.00



DATABASE
E-R MODELLING

STEP 3



Some RELATIONSHIPS are very easy to determine, others are more complex.

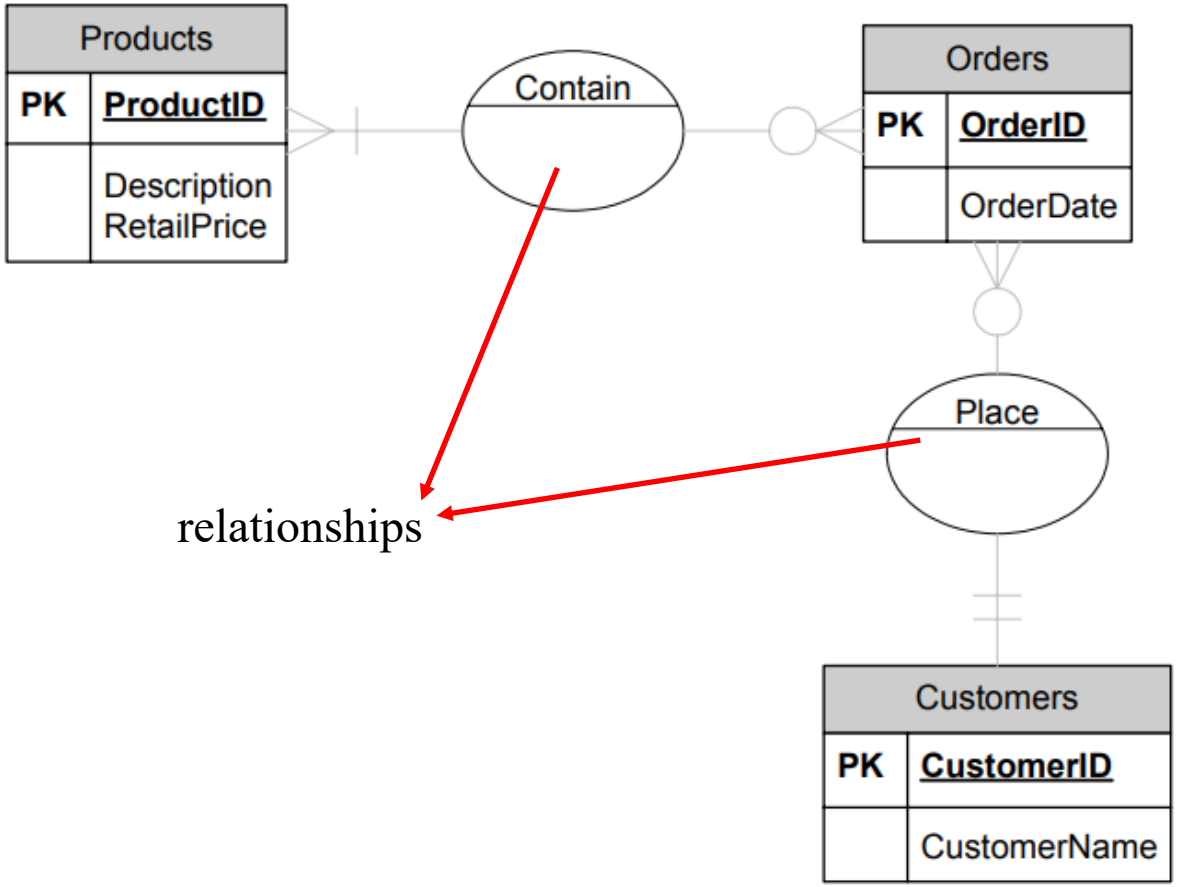
Anyway, there are some general guidelines that can help recognize RELATIONSHIPS.

The most important is to look at the document that you are working on and select the ENTITIES that appear in the same section: there is a strong possibility that they are related.

The fundamental thing here is to understand how the ENTITIES are related.

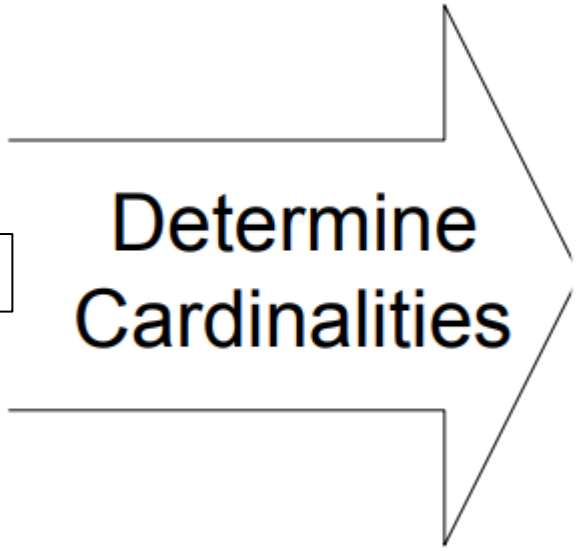
When the RELATIONSHIPS have been defined, it is important to assign meaningful names to them. If you can't come up with a name, use the names of the ENTITIES at the two ends of the relationship.

DATABASE
E-R MODELLING



DATABASE
E-R MODELLING

STEP 4



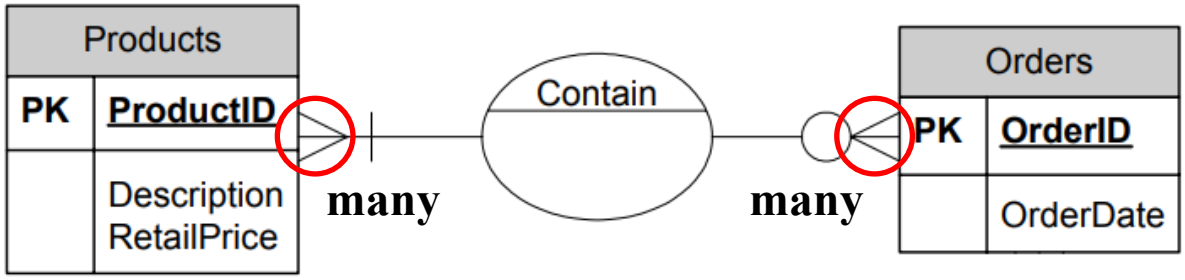
Cardinalities describe the minimum or maximum number of relationships that single instances of one ENTITY can have.

Maximum cardinality
The maximum number of relationships that instances can have.

Minimum cardinality
The minimum number of relationships that instances must have.

In the creation of the E-R diagram, maximum cardinalities are generally defined before minimum cardinalities.

DATABASE
E-R MODELLING



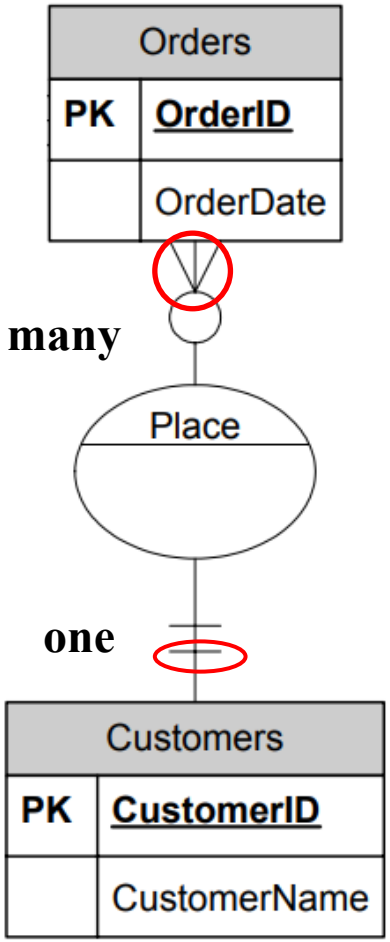
The maximum cardinality between PRODUCTS and ORDERS is shown in the picture above.

One instance of ORDERS can be related to many instances of PRODUCTS, because one order can contain many products. In this case, the cardinality from ORDER to PRODUCT is **many**.

One instance of PRODUCTS can be related to many instances of ORDERS, because one product can be ordered more than once. In this case, the cardinality from ORDER to PRODUCT is **many**.

Between ORDERS and PRODUCTS then there is a **many-to-many** cardinality.

DATABASE
E-R MODELLING



The maximum cardinality between ORDERS and CUSTOMERS is shown in the picture on the left.

A single ORDER can be related to only one CUSTOMER. This means that a single order can only be made by one customer and it can't be made by more customers. In this case, the cardinality between ORDERS and CUSTOMERS is **one**.

On the other side, a single CUSTOMER can place more than one ORDER. In this case, the cardinality between CUSTOMER and ORDER is **many**.

Between ORDERS and CUSTOMERS then there is a **one-to-many** cardinality.

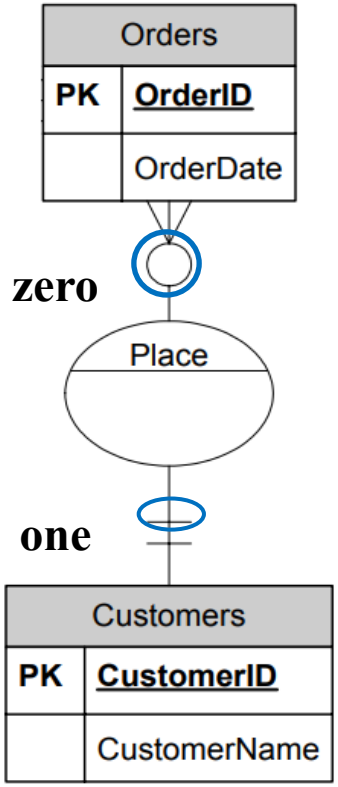
DATABASE
E-R MODELLING

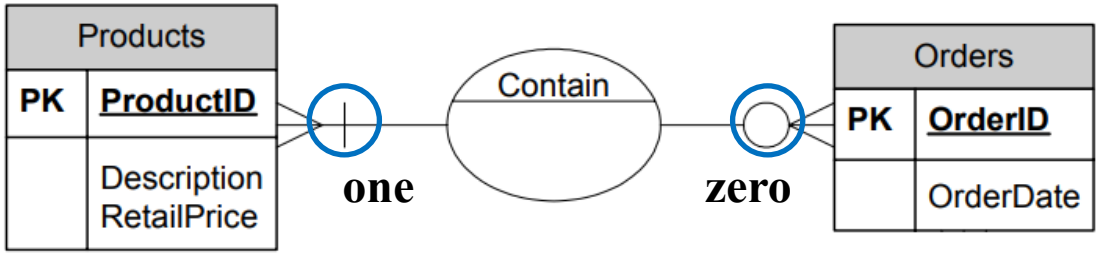
Defining maximum cardinality is easier than defining minimum cardinality.
In this latter case, cardinalities are less obvious if the information and knowledge on the document are not that clear.
A temporary solution is to make **assumptions** and to guess what the minimum cardinalities can be.
Assumptions can be a good starting point, but then the validity of the assumptions must be checked to continue creating the database.

In the case analysed so far, must the CUSTOMER be related to at least one ORDER?
The answer to this question is not clear, because maybe the organisation/shop allows the customers to set up accounts before placing their first order. In this case, the minimum cardinality between CUSTOMER and ORDER is **zero**, because the customer mustn't place at least an order to exist as an entity in the database.

The same happens for the minimum cardinality between PRODUCT and ORDER. It is possible to assume that there are products that haven't been ordered yet. If this kind of situation is allowed, then the minimum cardinality between PRODUCT and ORDER is **zero**.

DATABASE
E-R MODELLING

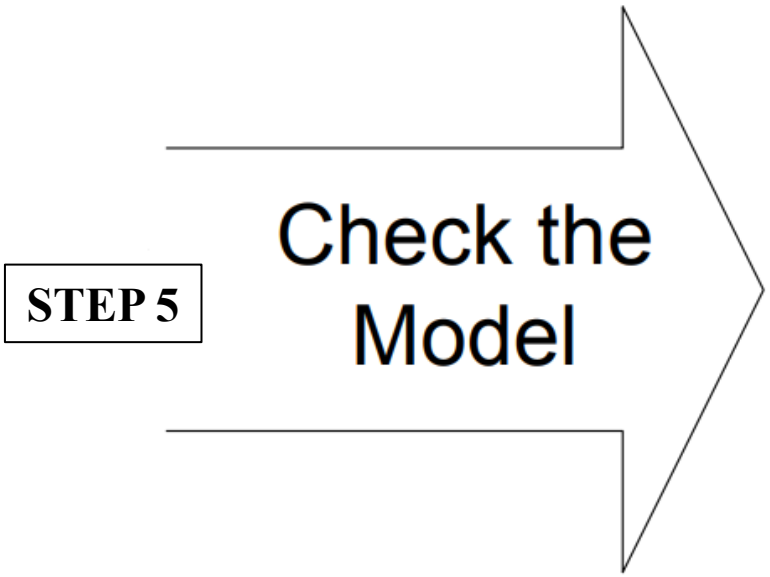




There may be PRODUCTS which havent't been ordered anything.
ORDERS, though, must contain at least one PRODUCT.

There may be CUSTOMERS who have an account but havent't ordered anything yet.
ORDERS, however, must be done by at least one CUSTOMER.

DATABASE
E-R MODELLING



The last step is often overlooked, but it is as important as any previous steps.

In this phase, the basic idea is to go back to the original document and make sure that the structure represented in the E-R diagram can satisfy the requirements.

This means that the representation of the E-R diagram must contain all the information of the original document.

DATABASE
E-R MODELLING

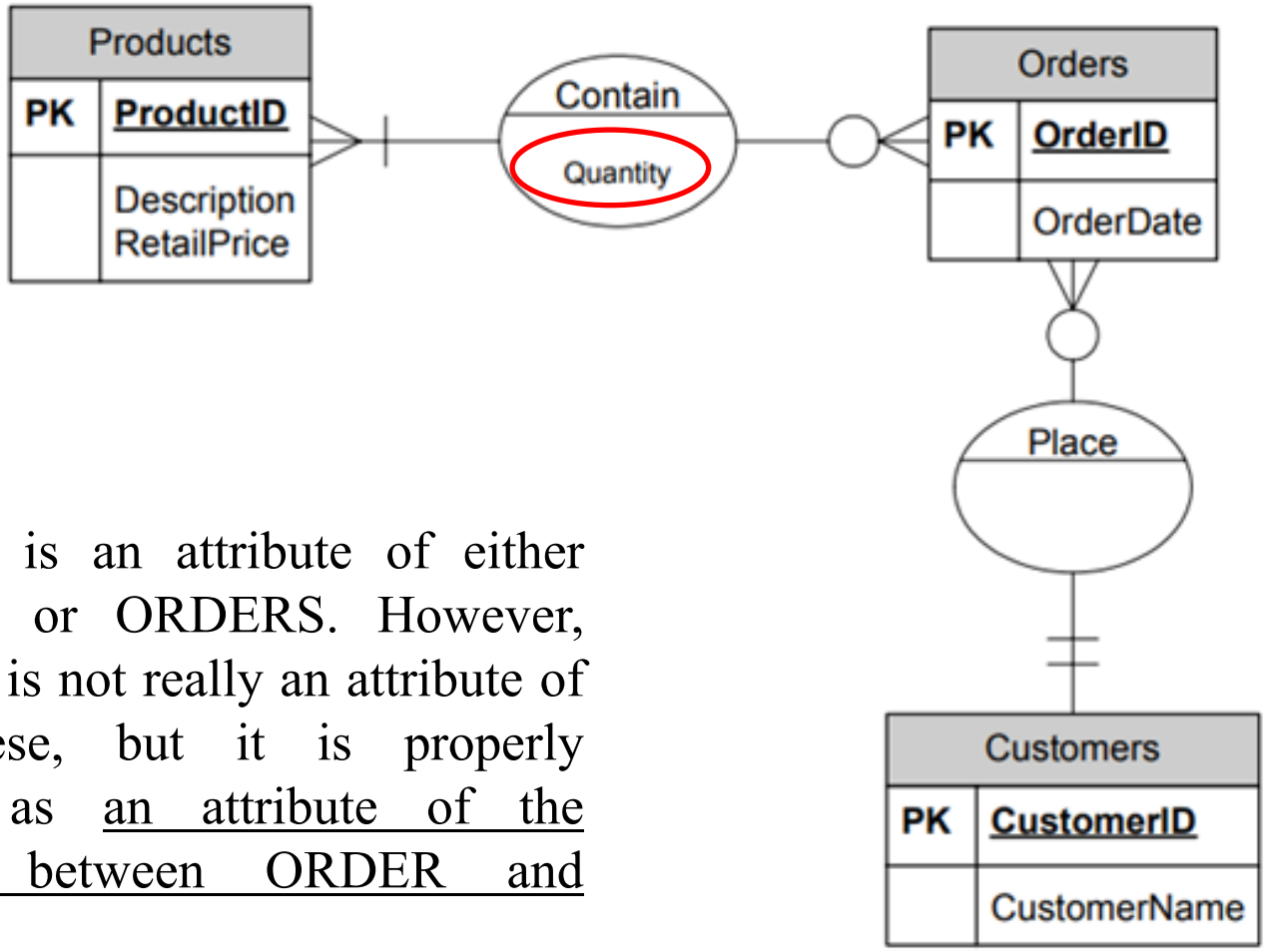
ORDER-NO:	44-44-4444	**CUSTOMER-ID:**	1002	
DATE:	10/31/98	**CUST-NAME:**	ABC Inc.	
PROD-ID	**DESCRIPTION**	**PRICE**	**QTY**	**EXT**
A123	STEREO SYSTEM	375.00	2	750.00
C235	8" SPEAKER	150.00	8	1,200.00
X002	SPEAKER WIRE	10.00	5	50.00
		TOTAL		2,000.00

Looking back at the original document, it is clear that three items are not represented in the E-R diagram: TOTAL, QTY and EXT.

TOTAL and EXT (EXTENDED: quantity multiplied by price per unit) aren't necessary to store because they are computed. It is not important to store data that can be computed.

QUANTITY, instead, is to be put into the E-R diagram.

DATABASE
E-R MODELLING



QUANTITY is an attribute of either PRODUCTS or ORDERS. However, QUANTITY is not really an attribute of any of these, but it is properly represented as an attribute of the relationship between ORDER and PRODUCT.

DATABASE
E-R MODELLING

In some cases, a **many-to-many relationship (M:N)** can be resolved, which means replaced by another type of structure.
A many-to-many relationship can be replaced by a new ENTITY, called **associative entity**. This is an entity that associates two instances of the two entities of the many-to-many relationship.

many-to-many relationship → associative entity

When the associative entity has been created, other important steps follow:

1. the definition of the ATTRIBUTES of the associative entity
2. the definition of the PRIMARY KEY
3. the definition of the CARDINALITIES

DATABASE
E-R MODELLING

ATTRIBUTES

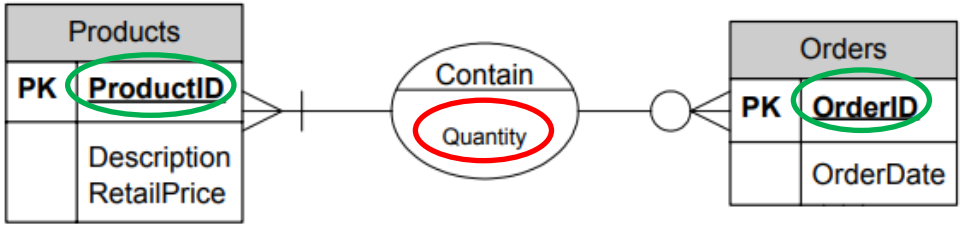
The attributes of the M:N relationship become the attributes of the associative entity.

M:N → PRODUCT-ORDER
ATTRIBUTES→ *QUANTITY*

PRIMARY KEY

The standard practice is to select the primary key taking it from the two original entities. We can use the combination of these two primary keys to form the new primary key.

M:N → PRODUCT-ORDER
ATTRIBUTES→ QUANTITY
PRIMARY KEY → *PRODUCTID + ORDERID*



DATABASE
E-R MODELLING

CARDINALITIES

Maximum cardinalities are always the same when converting M:N relationships.

The maximum cardinalities entering both sides of the associative entity are **many**; the maximum cardinalities entering the original entities are **one**. These cardinalities show the same relations shown in the original PRODUCT-ORDER relationship.

M:N → PRODUCT-ORDER
ATTRIBUTES→ QUANTITY
PRIMARY KEY → PRODUCTID + ORDERID
MAX. CARDINALITIES → *1:M (one-to-many)* in both ways

DATABASE
E-R MODELLING

CARDINALITIES

Minimum cardinalities are more complex.
The minimum cardinalities for the original entities are always **one**.
The minimum cardinalities going into the associative entity depend on the minimum cardinalities going into the original entities.
The minimum cardinality on the PRODUCT side of the associative entity is **zero**; the minimum cardinality of the ORDER side of the associative entity is **one**.

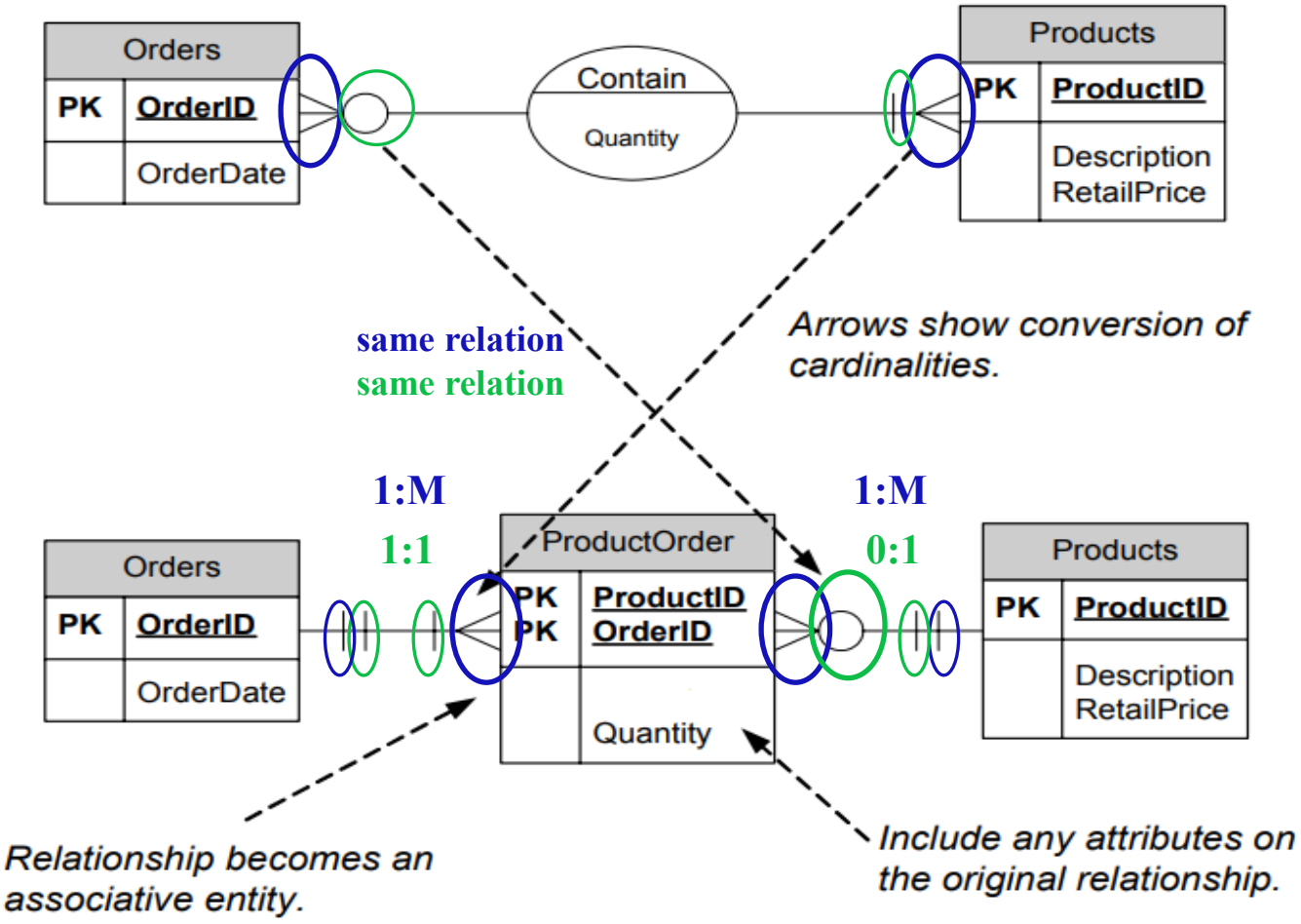
M:N → PRODUCT-ORDER
ATTRIBUTES→ QUANTITY
PRIMARY KEY → PRODUCTID + ORDERID
MAX. CARDINALITIES → 1:M (one-to-many) in both ways
MIN. CARDINALITIES → *1:1 (one to one)* and *0:1 (one to zero)*

DATABASE
E-R MODELLING

Explanation, with examples, of the minimum cardinalities:

In the ERD, the minimum cardinality going into ORDER is a zero, while the minimum cardinality going into PRODUCT is a one. This means that an instance of ORDER must be related to at least one PRODUCT, but an instance of PRODUCT does not have to be related to an ORDER. It is helpful to think of the associative entity as being an intersection of PRODUCT and ORDER. Since it is possible for an instance of PRODUCT to not be related to any instances of ORDER, it is also possible for a PRODUCT to not be related to any intersections of PRODUCT and ORDER. In other words, the minimum cardinality on the PRODUCT side of the associative entity is zero. Following the same logic, since each instance of ORDER must be related to at least one instance of PRODUCT, the minimum cardinality of the ORDER side of the associative entity is one. Figure 16 shows an annotated example of converting a many-to-many relationship to an associative entity.

DATABASE
E-R MODELLING



DATABASE
E-R MODELLING

The concept of **subtypes** and **supertypes** can be introduced. They are taken into account in two situations:

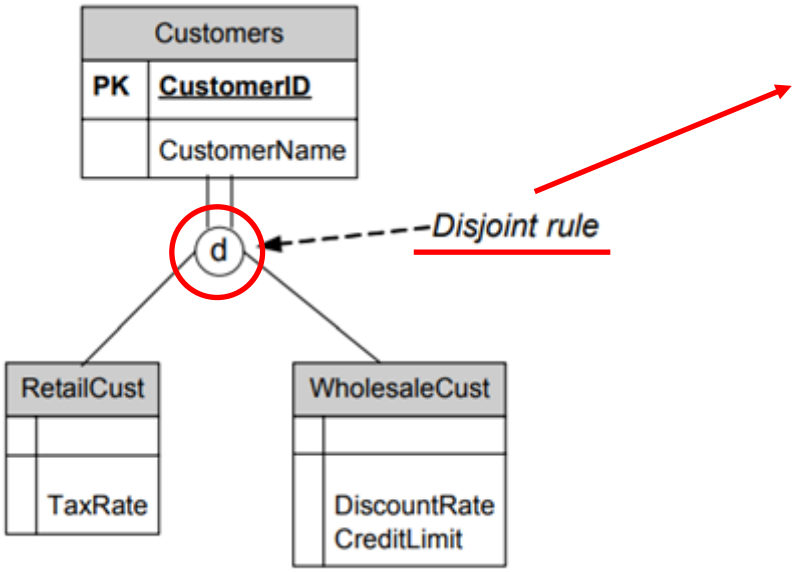
1. when some attributes refer only to some instances of an entity and not to all the instances of that entity
2. when some instances of an entity participate in a relationship and some other instances of that entity don't.

In these cases, it is useful to use a subtype/supertype structure, also called **generalisation/specialisation hierarchy**.

A key concept to associate to subtypes and supertypes is **inheritance**: each subtype inherits all the attributes of the supertype.

DATABASE
E-R MODELLING

For each CUSTOMER, we need to store an ID, and the customer's name. However, for retail customers we also need to track the sales tax rate. For wholesale customers there is no need to store the sales tax rate, but two additional attributes, discount and credit limit, need to be stored. Notice how each type of reference needs to have some common and some special information. This necessitates the use of a supertype/subtype structure.



The supertype can ONLY be one of the subtypes (either RETAILCUST or WHOLESALECUST).

In the E-R Diagram, it is represented by a "d" into a circle.

The opposite is the overlap rule, when the supertype can be both (or more) subtypes (not this case).

In the E-R Diagram, it is represented by a "o" into a circle.