Databases

(Different types of databases)
(Model of entity relations)

Dr Łukasz Piątek

Department of Expert Systems and Artificial Intelligence

University of Information technology and Management in Rzeszów

Lecture's schedule

- **PART I. Database's types:**
  - ***Flat*** **(one-table type data sources),**
  - ***Hierarchical*****,**
  - ***Network*****-type,**
  - ***Relational*****,**
  - ***Object*****-type, and**
  - ***Object-relational*****,**

- **PART II. E/R model (entities' relations).**

PART I

Database's types

Simple/basic databases

- **Data organised in the form of:**
  - **a single (and simple) table, or**
  - **a few unrelated tables (e.g., in the form of data sheet),**
- **New row(s) of data collected incrementally, in tables with a defined structure (i.e., collecting data by filling in the subsequent rows of table),**
- **Advantage:**
  - **a simple and intuitive way of collecting data,**
- **Disadvantages:**
  - **low functionality for the large data sets (*e.g., possible problems in a case of using significant variations of data types*),**
  - **no possibility of easy identification of the selected rows/data,**
  - **no relations between data (*redundancy*).**

Hierarchical databases (*data model*)

- **Data grouped in the form of subsequent tree's levels:**
  - ***parent-child*** **model type,**
  - ***root*** **– node defines the „beginning/start" of the tree structre,**
  - **Succesive descendent's levels (*till leafs*),**
- **Data set linked/connected to other data *(within the same tree only),***
- **Hierarchy of data created inductively (so-called from *general* to *detail*(s)),**
- **The hierarchical model allows to create/execute simple queries. Moreover, such queries could be further detailed.**

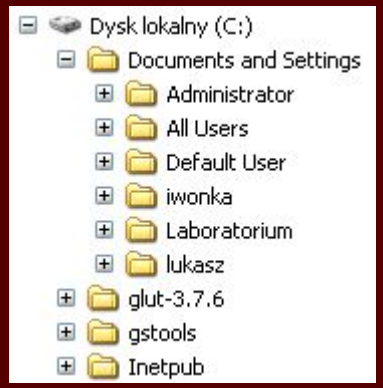
Hierarchical databases (*cont.*)

- **The hierarchical type databases impose the basic *conditions of data integrity*:**
  - **each row has got *exactly one* parent (except for the *root*),**
  - **a row having more than one parent should be copied for each parent separately,**
  - **deletion of a selected record/row means the deletion of all ‚derived' records (i.e., descendant),**

- **Disadvantage – it is not possible to create relations between records of two (or more) different trees,**

- **An example:**

  ***Windows* file system:**



Hierarchical databases (*cont.2*)

- ***XML*:**

  (*eXtensible Markup Language*) – the most popular format of saving data of a hierarchical model,

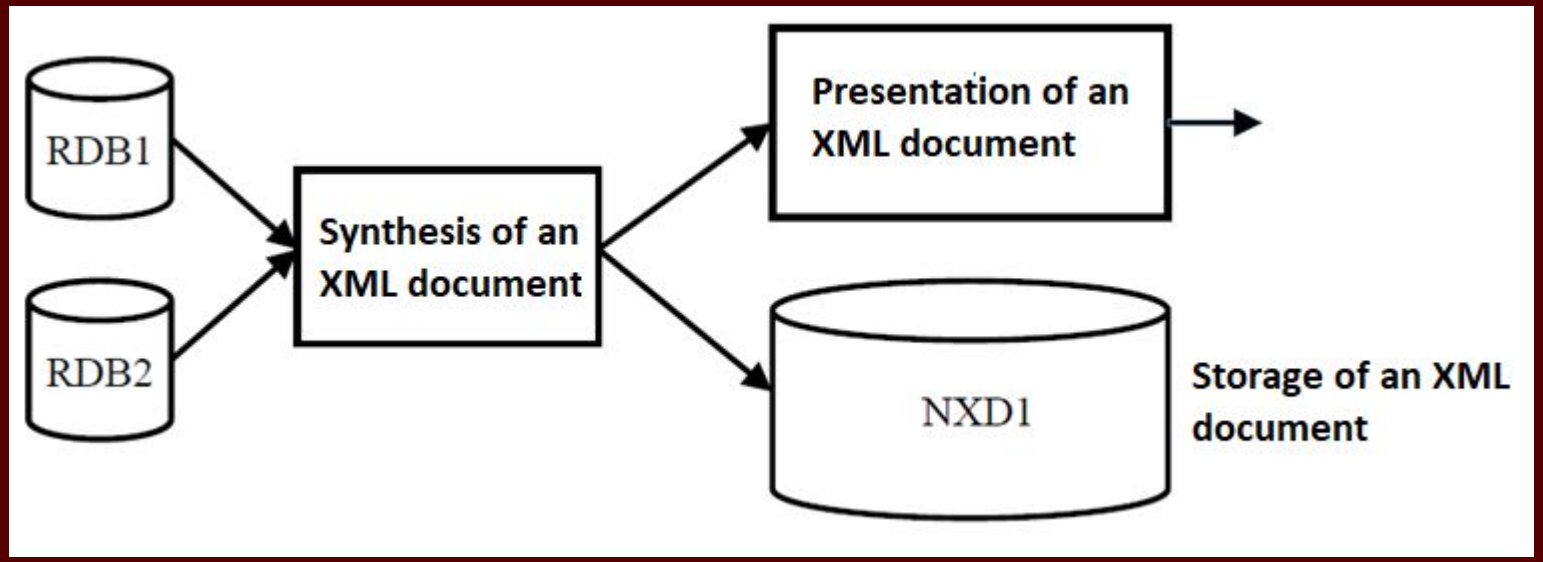
- **Used for:**

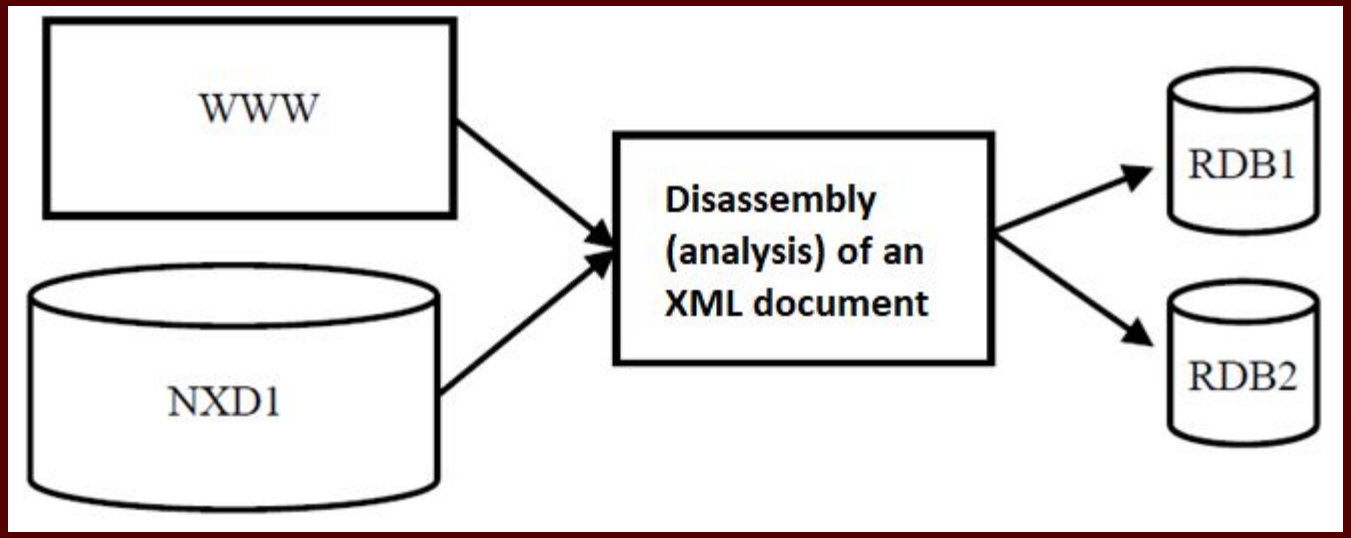
  intergration of data from different databases (but containing similar types of data),

- ***XML* and databases (*essential puroposes*):**
  - **storage of data in the form *XML* docs,**
  - **storage of *XML* data,**
  - **searching for *XML* documents, and**
  - **searching for data based on informations existed in *XML* documents.**

Hierarchical databases (*cont.3*)

- ***XML*** **and databases (*relationships scenarios*):**

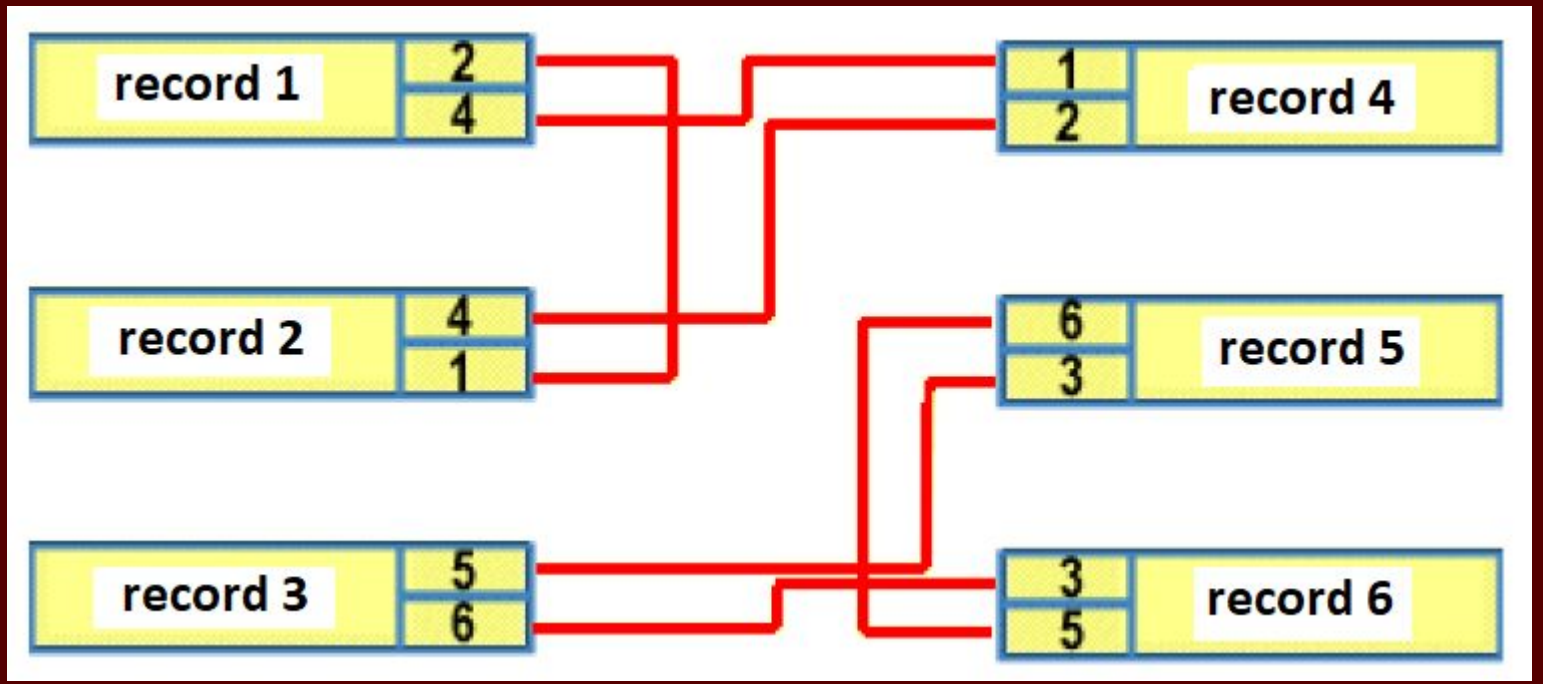




Network-type databases (*data model*)

- The network-type data model *„has nothing to do”* with computer networks,

- The *network*-type *model* enables to link any set of data to any other data (by using specific indicators/pointers). An example:



Network-type databases (*cont.*)

- **Difference with regards to the *hierarchical* model:**
  - **Each row/record can have *multiple* parents and/or *none*,**
- **The network database contains *two sets* of data:**
  - **a set of *record formats*, and**
  - **a set of *links/relationships*,**
- **Each relationship type specifies the records which are linked (i.e., *type of both – i.e., parent* and *child* – records),**
- **Each link (relationship) requires *indicator*/*pointer*.**

The network databases (*characteristics*)

- **Quick searching of data (*for small databases*),**
- **Complex and time-consuming process of data search (*large databases*),**
- **Extensive network of relationships – *indicators/pointers ocupy/require more space in memory than data*.**

Relational databases

- ***The most commonly*** **used (*implemented*) database model,**
- **Principle of action:**
  - **saving the data in the form of 2-dimmensional *table*(s) (so-called *relation*(s)),**
- **Record indentifiction:**
  - **based on the value of so-called *key* *(defining the key for the whole relation).***

Object-type databases

- **This data model using *object-oriented programming* paradigms,**
- **Used very rare (*rather within academic research*),**
- **However, the *object*-type *model* can be useful in the process of desigining the *relational* type database (!).**

Object-relational databases

- **Combining the advantages of *relational* and *object*-oriented databases:**
  - **manipulation of data (saved in the form of *objects*), and**
  - ***relational*-type manipulation mechanism.**

PART II

Entity-relationships (E/R) model

Entity/relationalship (E/R) model

- One of the most common models, enables an *abstract description* of the database structure/scheme,
- *Graphical representation* of the data structure in the form of entities-relationships diagrams.
- Basic elements:
  - set(s) of *entities*,
  - *attributes*, and
  - *relationships*.

Entity

- ***Entity*** **is a basic and undefined concept,**
- **Property:**
  - **An antity is distinguishable from other entities,**
- **Examples:**
  - ***Car, Person, Computer (items/things),***
  - ***Driving a car, buying a computer (events/activities),***
  - ***Feelings, imaginations (concepts/ideas).***

Set of entities

- A collection of all entities that are similar (in terms of characteristic), including for instance:
  - a collection/set of computer (e.g. laptops) entities,
  - A collection/set of students of a selected field of study,

- Similarity between an *entity* and an „*object*” in a sense of object-oriented programming:
  - a *set* of entities is similar to a *class* of objects,
  - on the other hand, the set of entities (*a*) has only static type, and (*b*) relates only to the structure of data.

Attributes

- *Attributes* are:
  features/props which describing the entities,

- Entities in a set are distinguishable based on values of their attributes,

- Attribute's properties:
  - the number of attributes is fixed by a given set,
  - the values of attributes are atomic (indivisable),
  - the names of attributes are (should be) unique/unambigous.

Key of entities' set

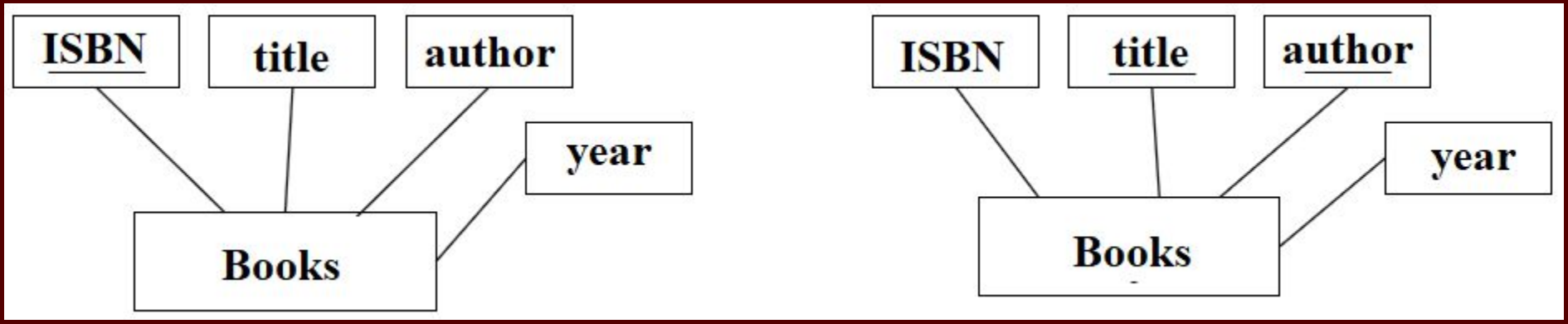
- **Attribute or group of attributes whose values unambigous identify an entity is a set of entities,**

- **Key's properties:**
  - ***Unambiguity*** **– in the set of entities there're no 2 entities whose key values are the same (there are no 2 „identical" entities),**
  - ***Minimality*** **– no any attribute can be removed from the key to ensure that the key is unique/unambiguous.**

Primary and alternative keys

- It happens that among attributes of a set of entities it is possible to select several different sets of attributes which meet the conditions of the key.

- Then one of such attributes (keys) should be selected as a *primary* key, and the others are so-called as *alternative* keys.



Relationship(s)

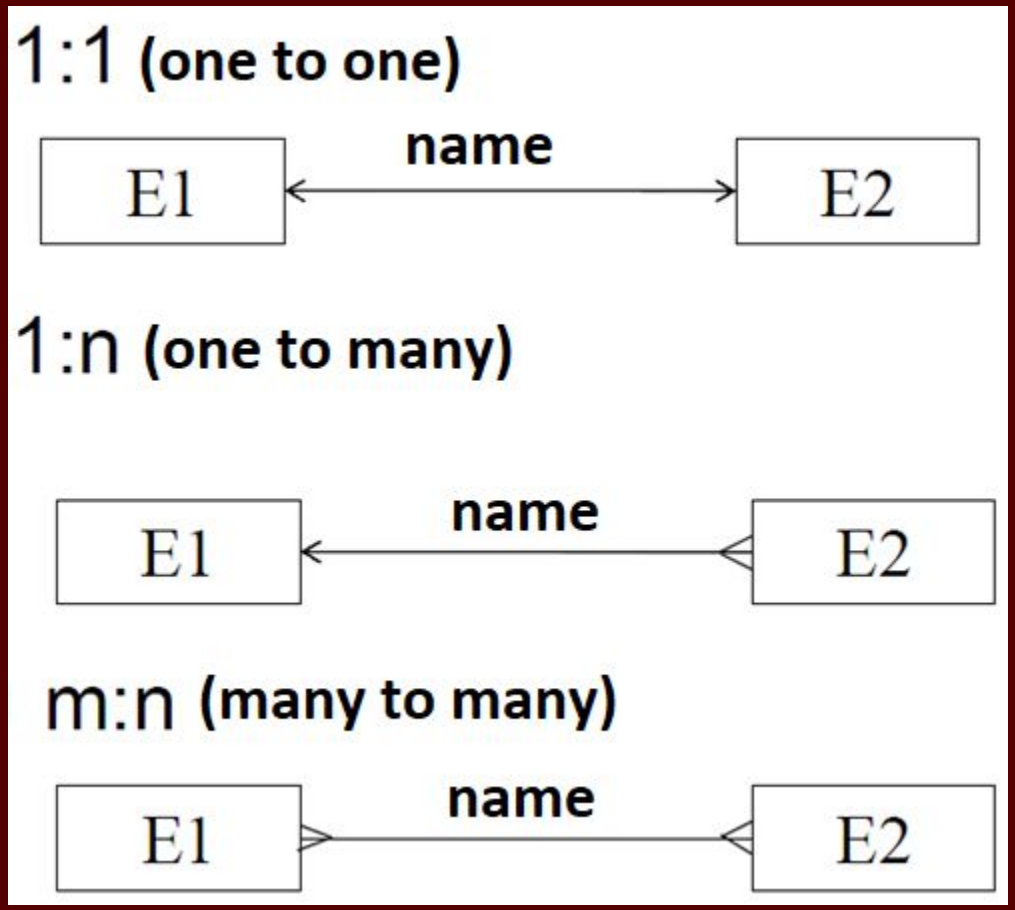
- **Relationship *REL* between entities *E1, E2,…En* is a set of n-elements tuples *(e1,e2,…,en)* such that *e1 ∈ E1, e2∈E2,…,en ∈ En*,**
- **Notation: *NAME(E1, E2,…En)*,**
- **Relationship can exist as well between selected entities in a given set.**

Relationships (*examples*)

MOTHERHOOD(PERSON, CHILD) – contains all pairs (m,c) where m is mother of a child c

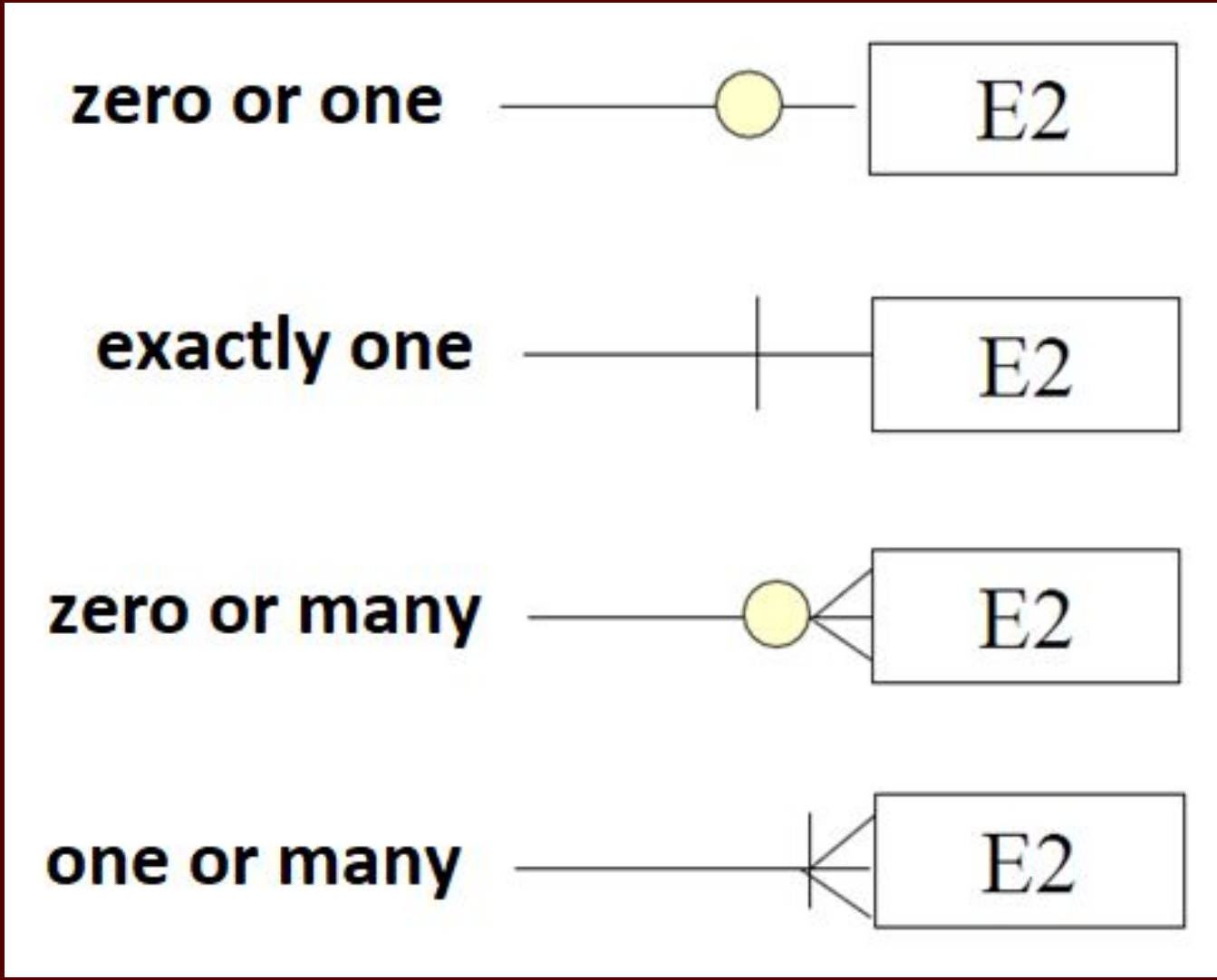
TEACHING(TECHER, STUDENT, SUBJECT) – contains all tuples (t,s,sub) where teacher t is teaching a student s from the sub subject

Multiplication of relationships



- The multiplication of relationship can be also equal to 0 (on one of the selected table/relation).

Multiplication of relationships (*another notation*)

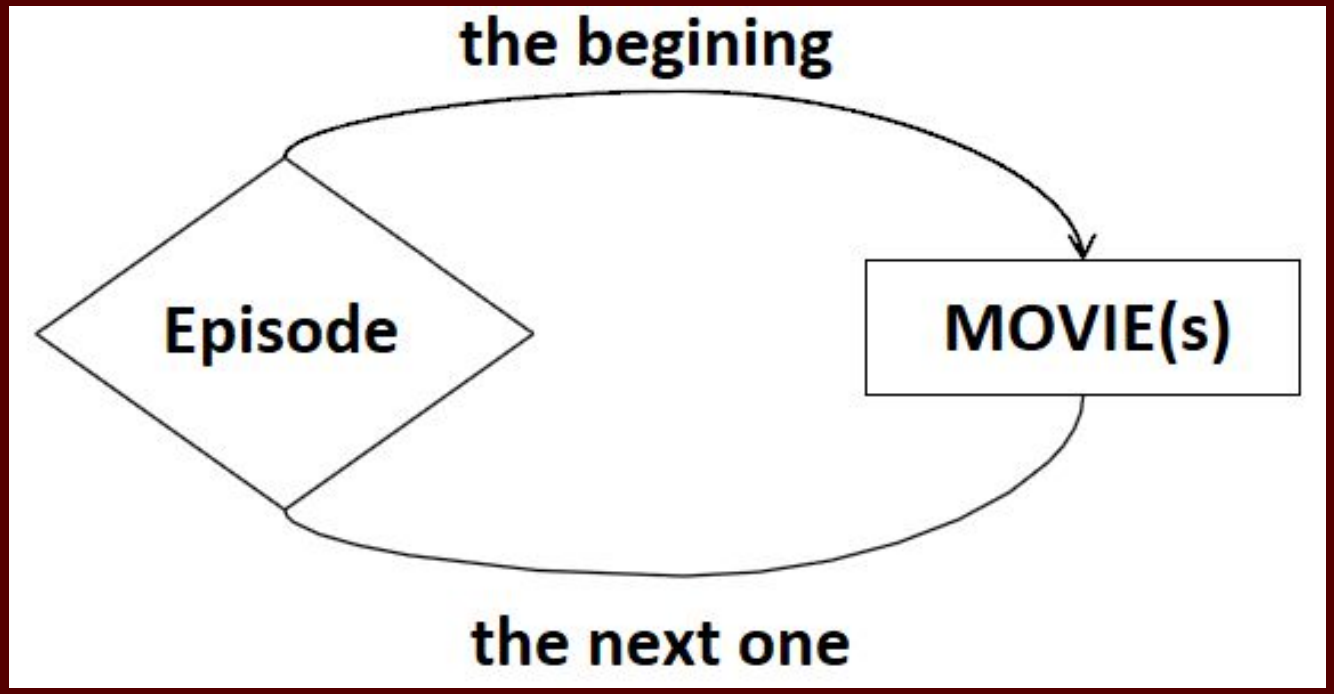


Roles in relationship

- A given set of entities may occur in a given relationship more that once. In such case we should draw as many edges between the relationship and the set of entities as many times as selected set of entities appears in this relationship (so-called *roles*).

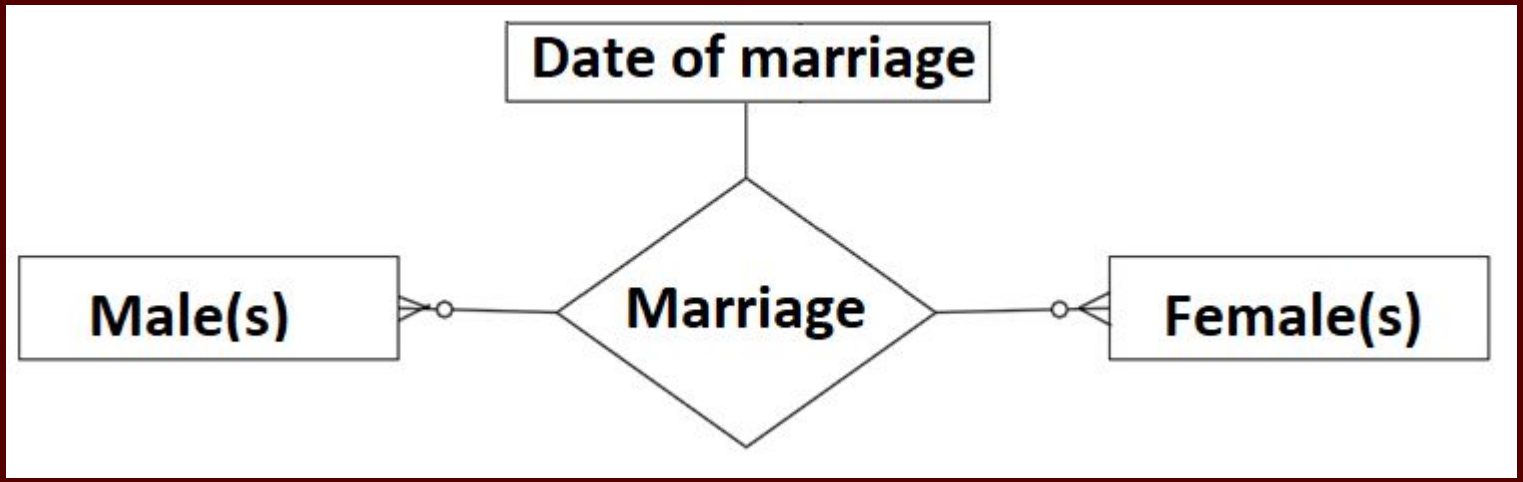
- The edges between the relationship and a set of entities are labelled with names so-called *roles*.

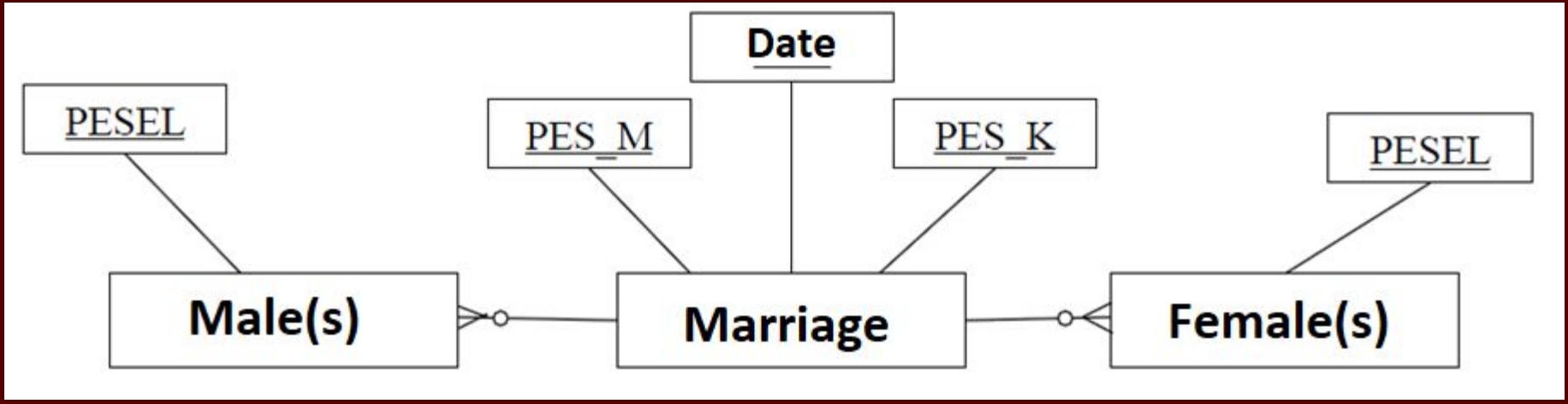
Roles in relationship (*cont*)



- There is a relationship between *2* films, where one of them is a continuation of the other.
- It is assumed that a film can contains many parts, but only one of them can be the first one (e.g, the 1st episode).
- Relation type – *1:N*.

Attributes in relationship





Relationship of higher level(s)

- Relationships do not have to be binary type (i.e., combining of 2 sets of entities),
- Relationships of higher levels can be reduced to a several binary type relationships.

Sub-classes in E/R model

- A set of entities can contain certain/specific entities with special properties that do not match all entities from this set. In a such situation an additional sets of entities (i.e., *subclasses*) can be created.

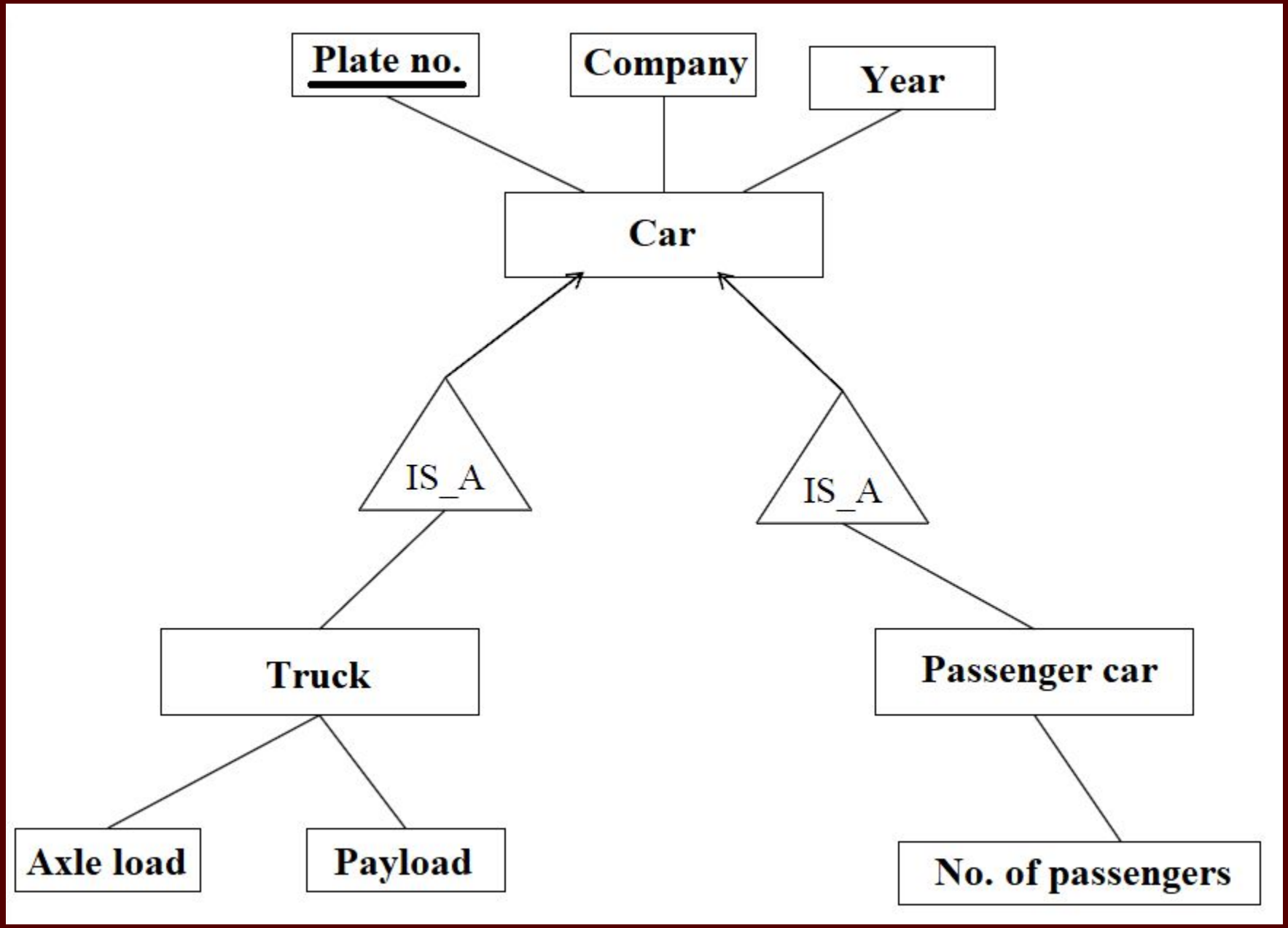
- Such *subclasses* contain additional (specialized) attributes and may create/establish another new relationship(s),

- A set of entities is linked with its *subclass* by relation so-called *IS_A*.

Relationship IS_A

- ***IS_A(E1,E2)*** **– means that each entity from the set *E1* is also an entity from the set *E2* (*E1* is a subset of set *E2*),**
- **Relationship *IS_A* is a relationship builded-in into *E/R* model (it is a relationship type as *0..1:1*),**
- **Each entity belonging to the set *E1* has all attributes from the set *E2* and additionally its own attributes.**

Relationship IS_A (*an example*)



Principles for the creation of E/R model

- **Proper identification of *enties* and *set(s) of entities*,**
- **Definition of appropriate *attributes* and *keys of entities*,**
- **Identification of all *IS_A*-type relationships. Using of *entities' subsets* and definition of relevant *attributes*,**
- **Identification of all relationships existing between sets,**
- **Reduction of all *multi-argument* relations to *binarny*-type relations.**

University of Information Technology and Management
Sucharskiego 2 Str., 35-225 Rzeszów, Poland

Thank's for your attention…