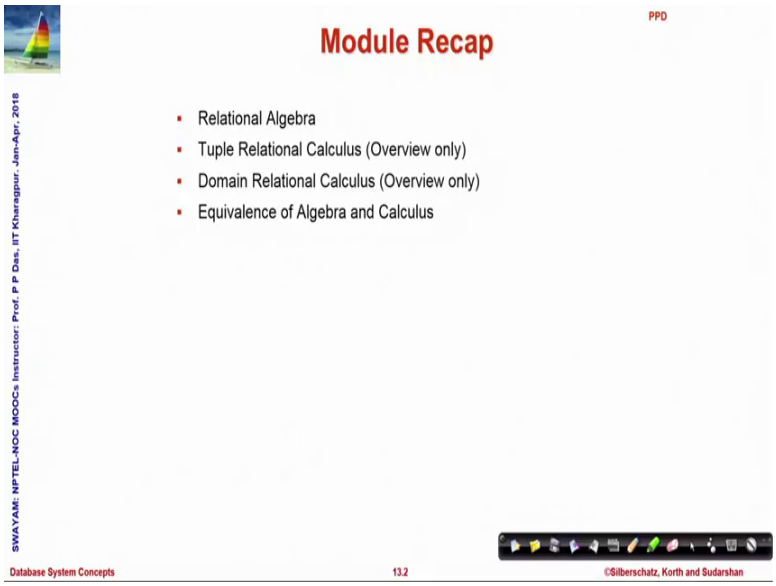
Database Management System
Prof. Partha Pratim Das
Department of Computer Science & Engineering
Indian Institute of Technology, Kharagpur

Lecture – 13
Entity-Relationship Model/1

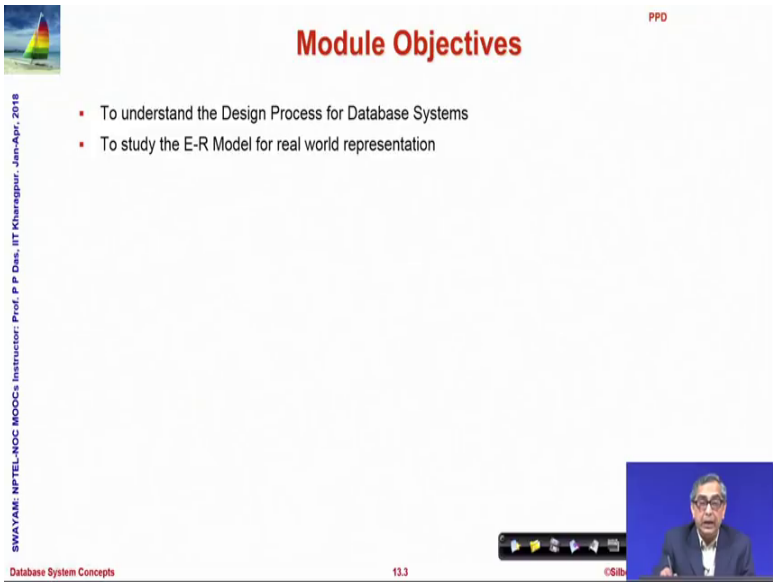
Welcome to module 13 of Database Management Systems. In this module and the next 2 we will discuss about Entity Relationship Model.

(Refer Slide Time: 00:34)



So, far we have had a good look into the SQL language, the query language and it is formal basis in terms of relational algebra and calculi.

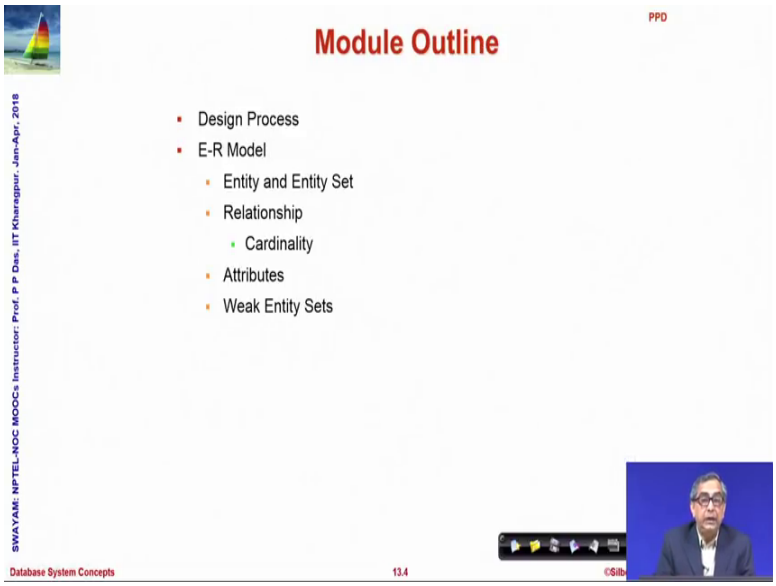
(Refer Slide Time: 00:44)



In this module we will un try to understand the design process for database systems, because so far whatever we have done we have assumed that the schema is known to us, that some instance is given to us and then we have tried to extract different query information from the relation; but now we will look into how do we model the real world and actually get into the design process.

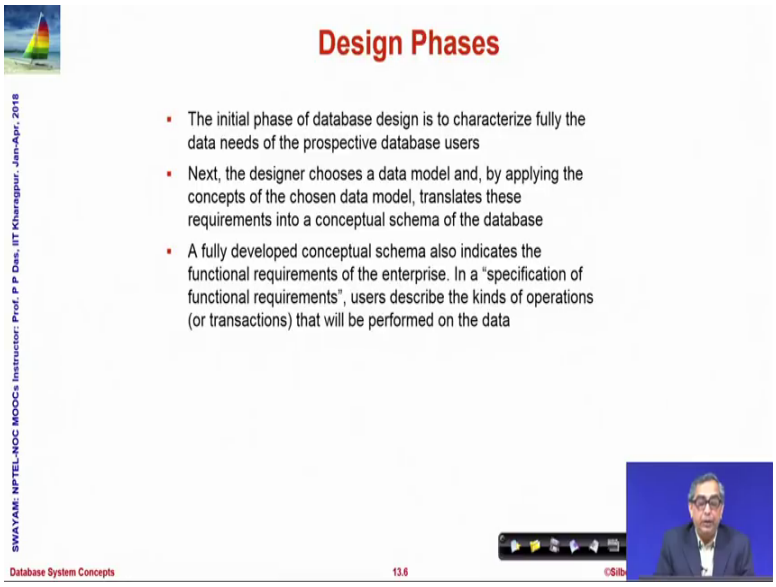
So, after an overview of the design process we would study entity relationship model, which is used to represent the real world whatever exists in the real world that will have to be represented for our use and final representation in terms of different relations.

(Refer Slide Time: 01:32)



The design process at an abstract level.

(Refer Slide Time: 01:43)



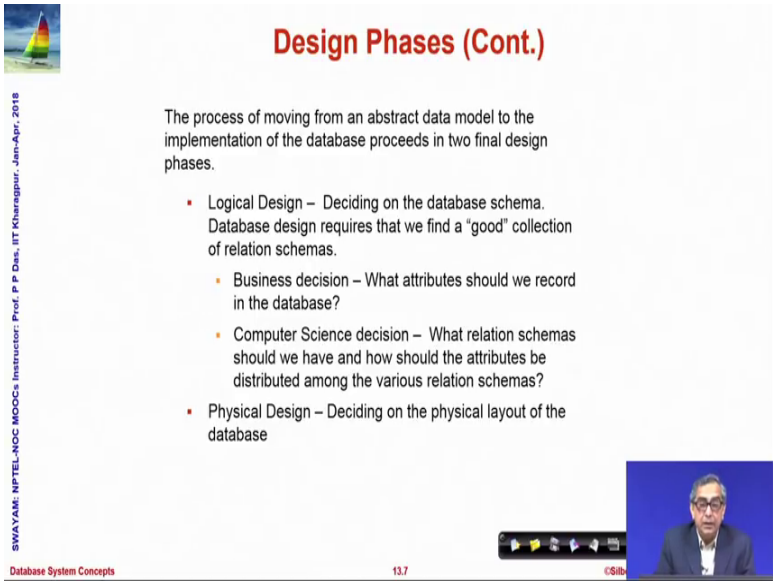
The initial phase of database design certainly has to characterize what data is required to be maintained for an enterprise. So, whether I am doing, if I am doing an university database naturally we will need to identify that what are the data needs the students need to be described, the instructors need to be described, the courses sections time slots grades examinations etc; but if I am trying to deal with an world which is say railway

reservation, then I will need to deal with stations trains date (Refer Time: 02:26) the different classes of coach that the train has and so on.

So, the initial phase is to characterize the data requirement next the designer has to choose a data model because, unless we can we cannot deal with a natural language or English kind of description and work towards getting a particular schema.

So, we will need to use a data model and apply the concepts of the data model that we choose and translate the requirements into what is known as a conceptual schema of the database which is not a not a very concrete 1. But, a conceptual one this is what grossly what I want to do and a fully developed conceptual schema will indicate my functional requirements, in terms of what usually is called a specification of functional requirements system requirements. If it will specify what kind of users will be involved what kinds of operations transactions will be performed and so on.

(Refer Slide Time: 03:43)

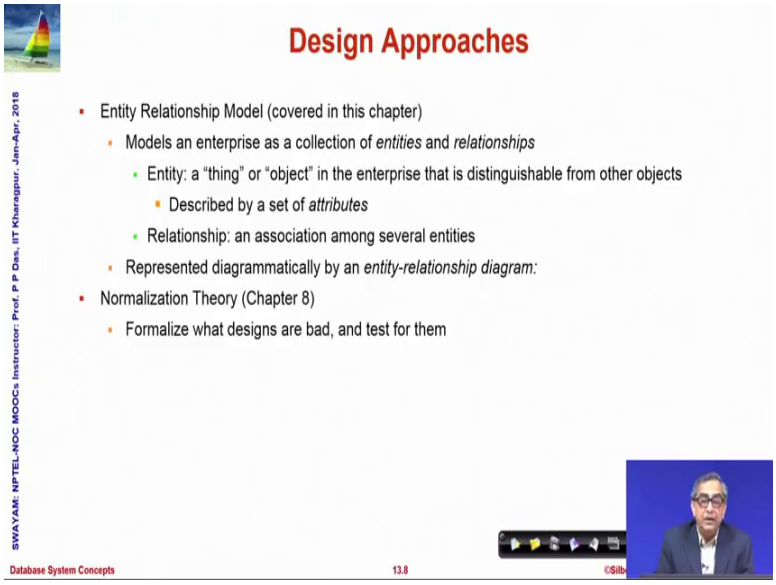


Now, once we have that kind of a conceptual model that abstract that is a conceptual more abstract data model, we will go to the next phase of the design which is finding out what is the more concrete design through a process of logical design. In the process of logical design we will first decide on the database schema, we need to decide on what is a good schema.

So, there are principles to say that what is good and what is not so good, we need to make business decisions to find out which attributes we record in the database; we need to make computer science decision as to how the relational schemas will be interrelated between themselves, how the attributes will be distributed and at a last phase we need to also decide on the physical design which will tell us what is the physical layout of the data.

So, conceptual design refined into logical design finalized with physical design is our gross process of design.

(Refer Slide Time: 05:06)



Now, in this for the conceptual design we primarily follow a model called entity relationship model; that tries to identify the collection of entities and relationships. An entity is nothing, but it is an object is a thing that is distinguishable from other objects. So, if I say that student is an entity then the student is distinguishable from another entity course both of them are distinguishable from a third entity instructor and so on.

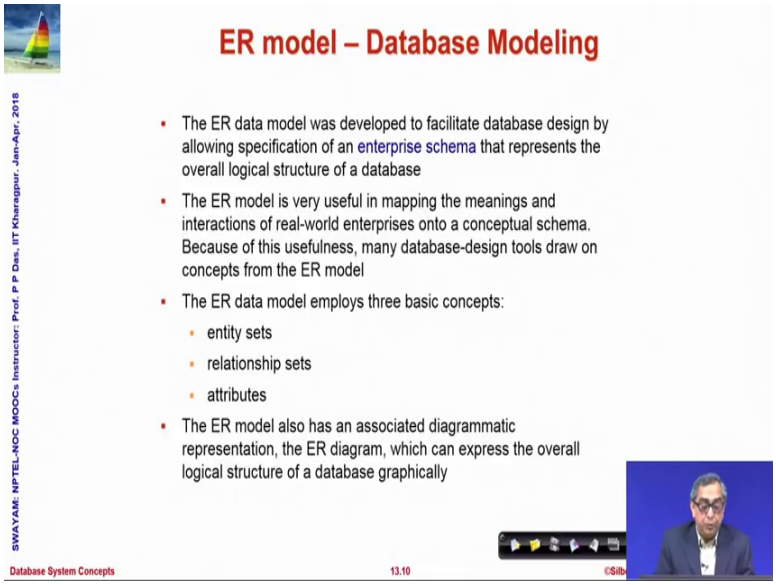
So, every entity for the purpose of distinction is described by a set of attributes or properties and these entities will have relations between them. For example, you can say that a course will be attended by students; students will be advised by instructors. So, this attended by advised by these are relationships or association between several entities and the model which represents initially diagrammatically and then in textual form this kind

of relationship is known as the entity relationship model or the entity relationship diagram.

We will then use it to get a relational set of relational schema which subsequently we normalize; the normalization is nothing but refinement of the design which improves a design to make it better in terms of correctness, in terms of ease of manipulation performance and so on. So, it basically removes bad designs from the database and converts them into good designs; we will talk about this normalization theory later in the course. Right now we are interested only in the entity relationship model, which will be used for conceptual design and then will give us the basis for the logical design in terms of the schemas.

So, let us take a deeper look into the entity relationship model and entity relationship model as I said is developed to facilitate the database design.

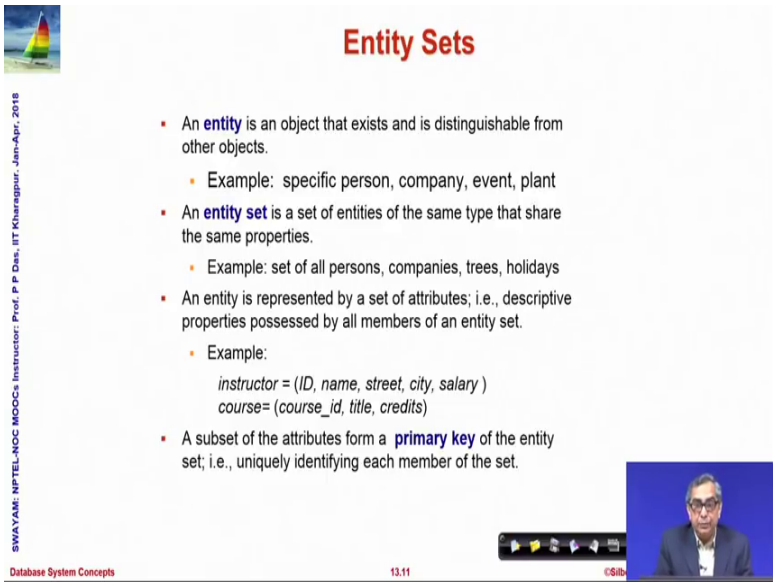
(Refer Slide Time: 07:45)



Get the overall logical structure it is useful in mapping the meaning and interactions of the real world in terms of certain diagrammatic schemas and it employs 3 basic concepts entities or entity sets we talked about entities, all entities that share the same set of properties like if student is an entity, then the collection of student is an entity set a instructor is an entity collection of instructors is an entity set.

So, all entities in an entity set will share the same set of attributes, will have relationship sets which define relationship between multiple entity sets and certainly in the process will use make use of attributes. These are the 3 key components of an ER model it also has a ER diagram as we will show soon.

(Refer Slide Time: 08:51)

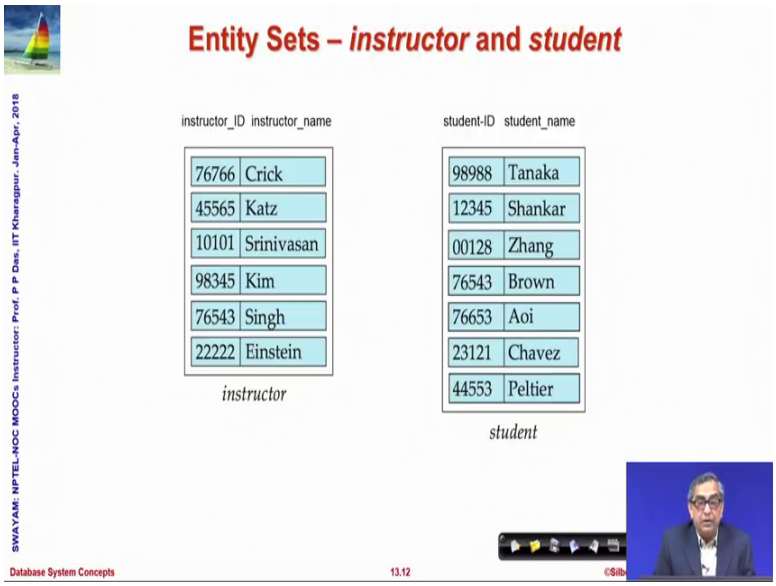


So, as already defined entity is an object that exist and is distinguishable from other objects, entity set is a set of entities of the same type that share the same properties and an entities is represented by the set of attributes or properties that describe it.

So, when we say instructive for example, if we say here these are my attributes you have already learned this in terms of studying SQL. So, it has there is 5 attributes and these 5 attributes together or the values of these 5 attributes for a particular instructor defines my entity set instructor, collection of these attributes define my entity set courses. So, these are my different entity sets that exist that can be defined.

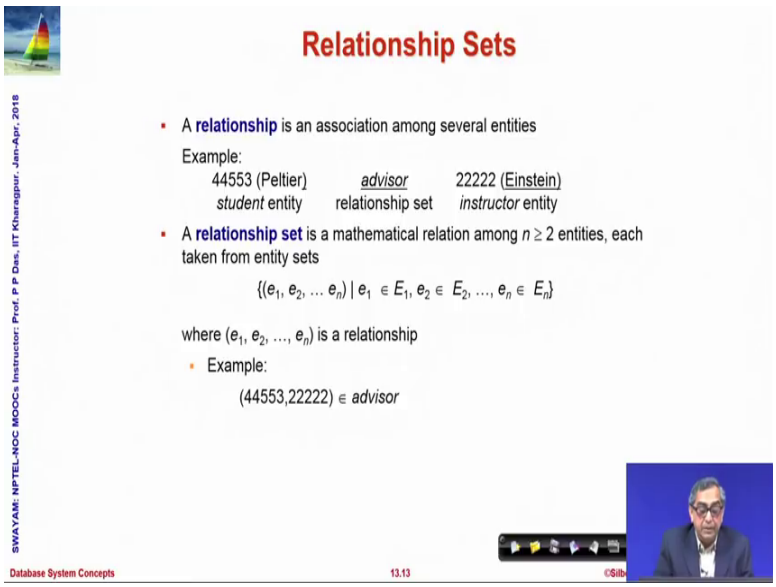
So, a subset of attributes in the entity set forms a key called the primary key, which can uniquely identify every entity in that entity set we have already been familiar with this concept of primary key the same concept continues.

(Refer Slide Time: 10:00)



So, these are examples of entity sets instructor with 2 attributes and student with 2 attributes as well.

(Refer Slide Time: 10:14)

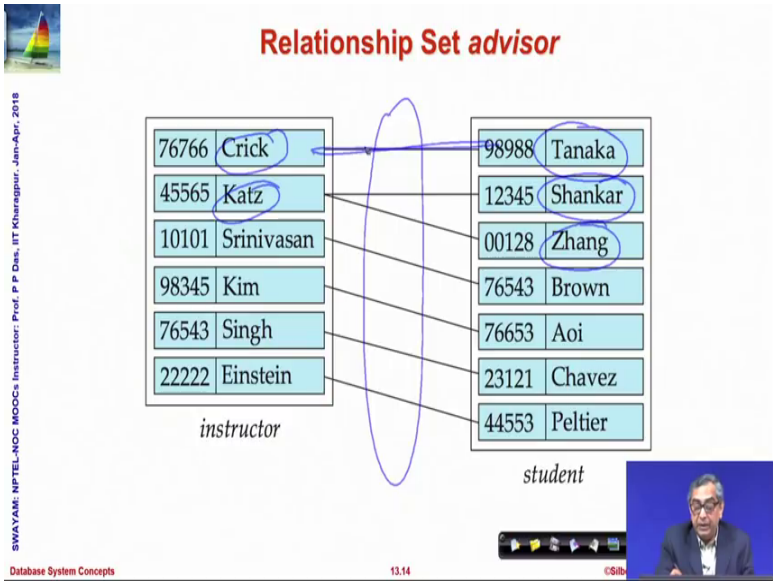


A relationship is an association among two or more entities, so here we have an entity here shown as a student this is a student entity, identified by the student id which is a primary key in the student entity set. We have an instance of an instructor entity identified by the id of the instructor Einstein, which identifies any instructor uniquely and then adviser is a relationship set which relates these 2.

So, what I we want to mean is if I say adviser relates 445532 2222, what I want to mean is peltier the student peltier is advised by the instructor Einstein. So, whenever we relate two or more entity sets like this we get relationships. So, a relationship is a mathematical relation Emma more than two or more entities, each taken from the entity set.

So, you can see that it can have components e 1 e 2 en, n entity sets and each entity e 1 should belong to entity set capital E1, e2 should belong to entity set capital E2 and so on and is called a relationship we have already seen the advisor relationship as above.

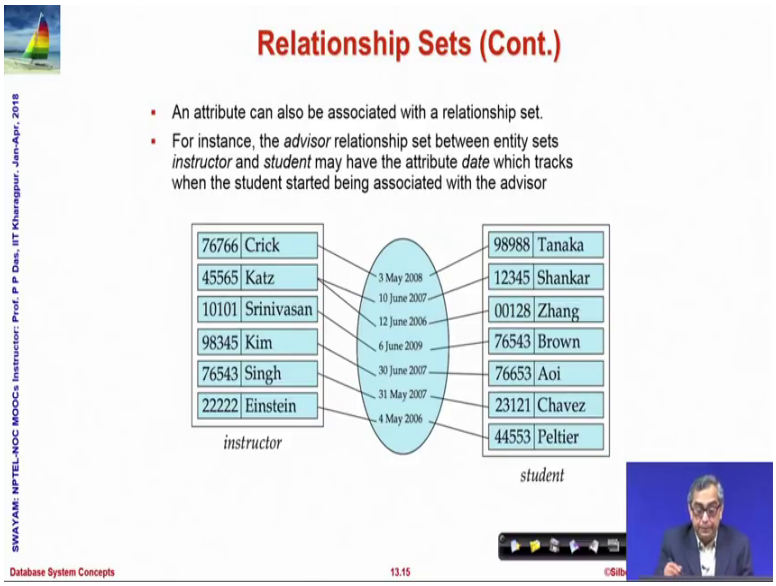
(Refer Slide Time: 11:58)



So, here what we show is a relationship advisor by these arrows these lines. So, what we are showing is this connection between these 2, show that this student is advised by this instructor where as you can see. So, crick advises Tanaka where as Shankar and Zhang both are advised by Katz.

So, this group of associations between instructor and student is the gives me the relationship adviser as to who advises whom.

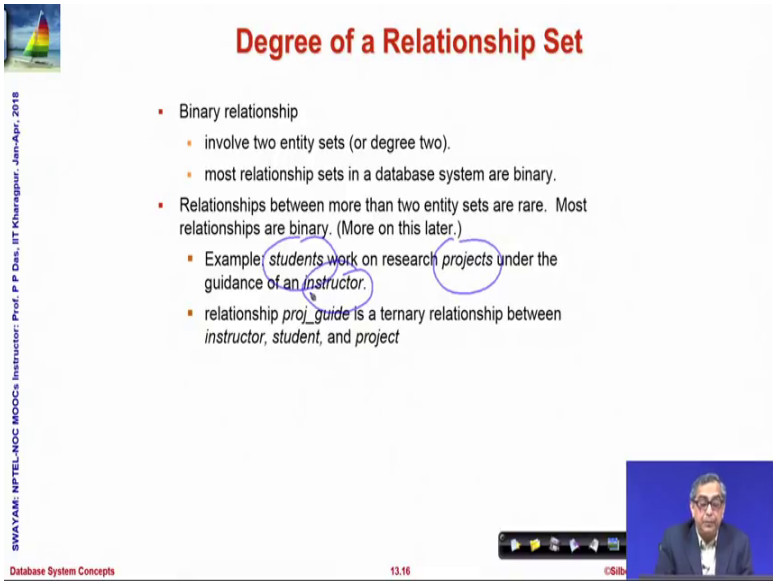
(Refer Slide Time: 12:39)



A relationship also like the entity sets the relationship also can have some additional attribute. For example, when I say that crick advises Tanaka I may associate an attribute date type attribute set third May 2018, to mean that when did this process of crick advising Tanaka started, we can it can be some other attribute also.

So, all that I am trying to highlight is attributes can be assigned to relationships as well.

(Refer Slide Time: 13:15)

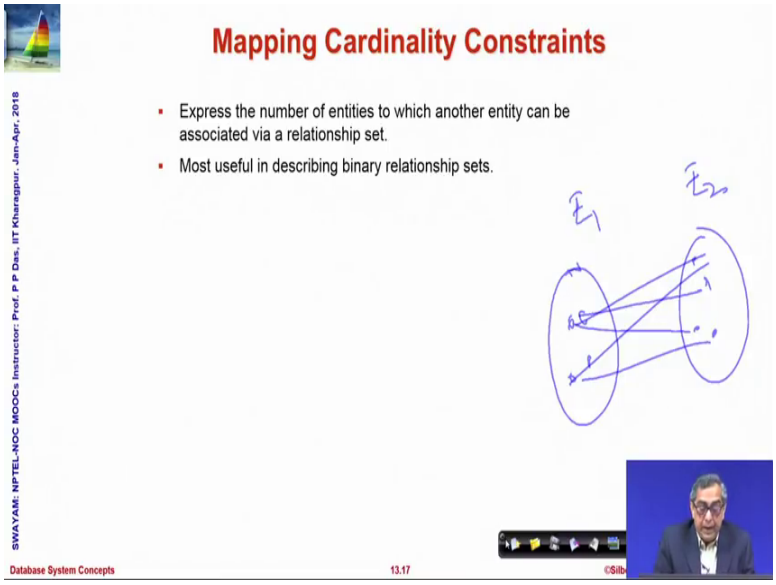


Now, how will a relationship span out, we have said that a relationship must involve 2 entity sets. So, primarily relationships are binary it involves 2 and most relationships in

most databases are binary in nature, but it could be that there are we will see later that there are possibilities of having relationships which are more than binary ternary and higher.

So, here are example students works on research projects under the guidance of an instructor. So, here we have as you can see student's research projects and instructors, so there are 3 entity sets. So, if I want to maintain a relationship of say project guidance between them then that turns out to be a ternary relationship we will talk about this more later.

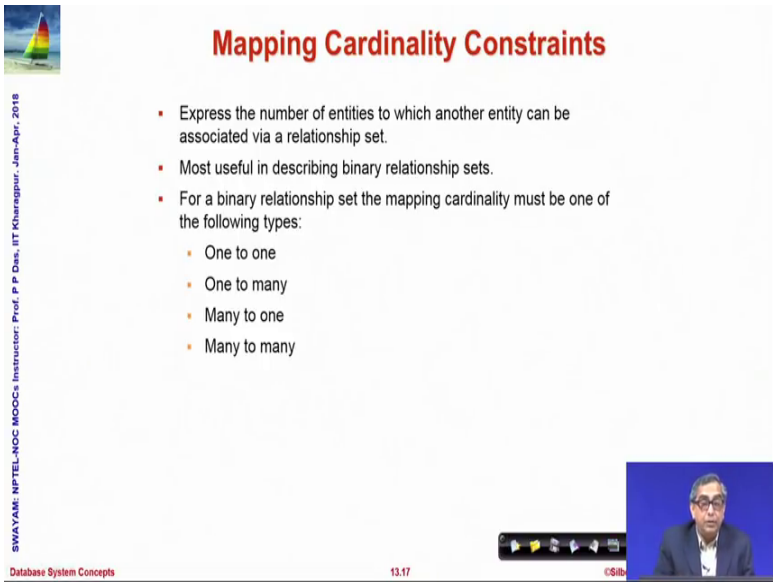
(Refer Slide Time: 14:16)



There are constraints in terms of the cardinality of the relationship, the cardinality basically talks of that when we have when I have a relation entity set E1 and identity set E2.

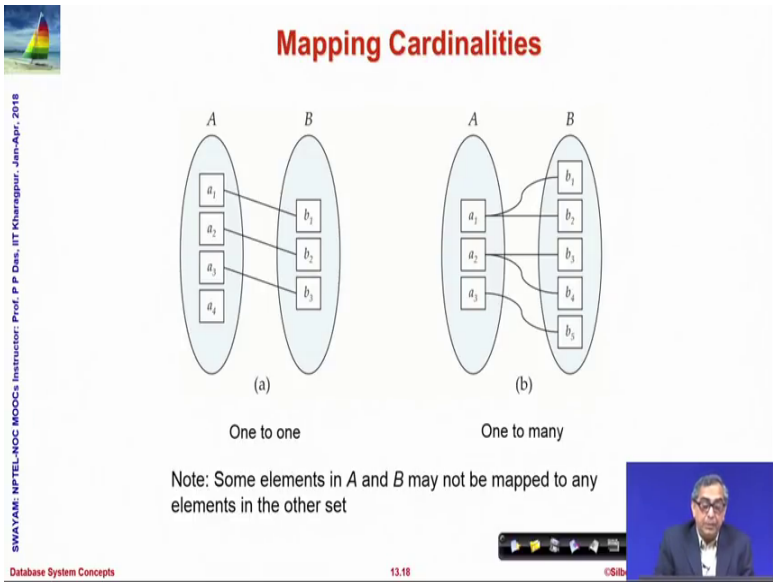
So, there are different entities in them and I have different associations between them, then the question is how many of the entity of one entity set is related to how many of the entities of the other entity set and certain types of cardinality measures are very important.

(Refer Slide Time: 14:58)



To track and we say it is whether it is 1 to 1 to many many to one or many to many.

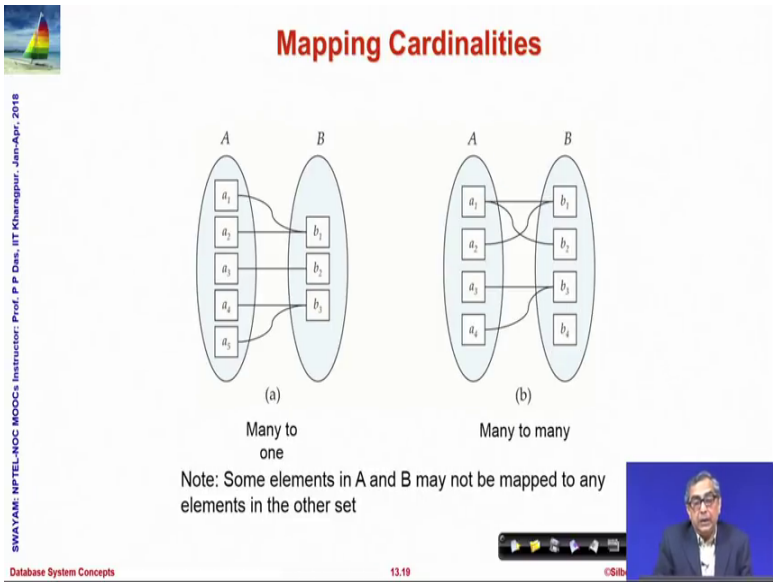
(Refer Slide Time: 15:08)



So, here the examples or the schematics, so in the first one in the diagram A you see that every entity from the entity set A relates to exactly one entity in the entity set B or you can say at most one entity in then they decide B similarly every entity in entity set B relates to exactly one entity in entity set A or at most 1 entity and entity set a if this holds then we say this relationship is one to one whereas, in diagram B you see that a 1 relates to b 1 as well as b 2 a 2 relates to b 3 as well as b4.

So, 1 entity in A relates to more than 1 entity may relate to more than 1 entity in B, but if you look from B side every entity in B is related to at most 1 entity in A then we say from A to B it is 1 to many.

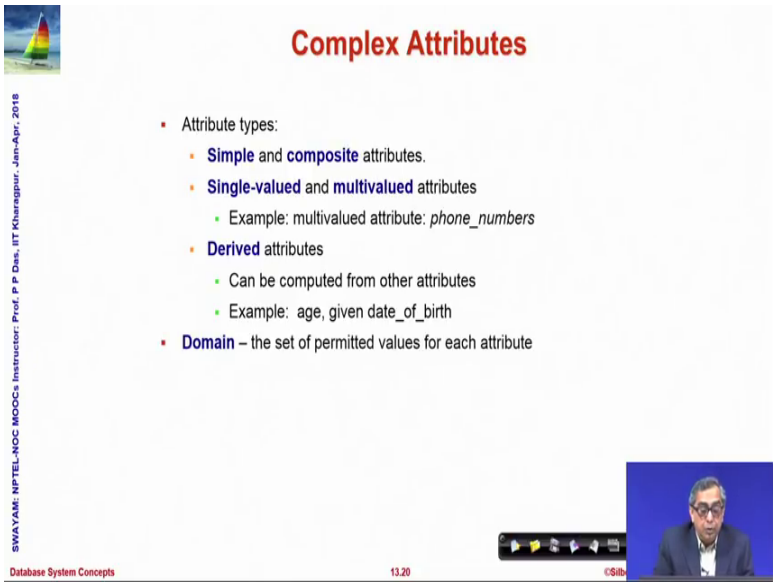
(Refer Slide Time: 16:04)



Now naturally since I can put the relations in any order as we have one to many if you look in the other direction it becomes many to 1. So, many to one is from A to B many to one is where more than one entity in set A may relate to one entity inside B, but all entities in set B relates to at most one entity inside A and when there is no restriction at all that is any number of entities in set A may relate to any number of entities inside B and any number of entities in set B may relate to any number of entities in set A we say it is a many to many relation.

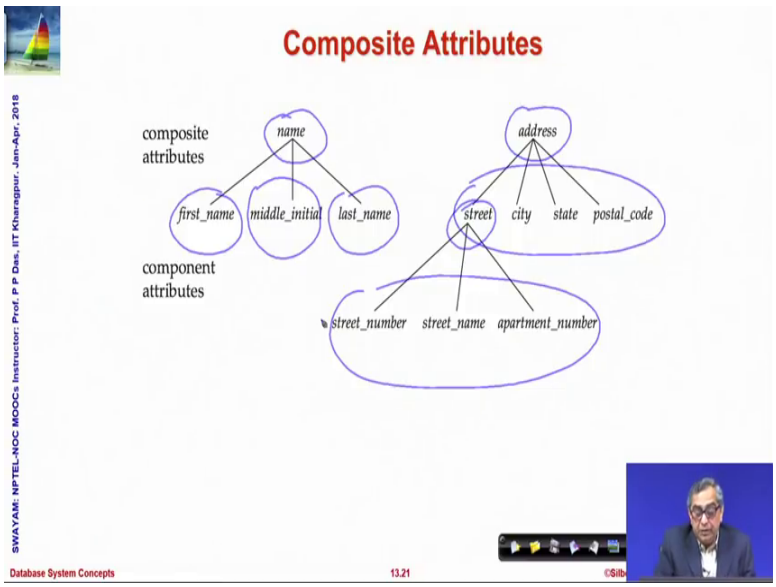
So, we have one too many one to one, we have one too many and many to one and we have many to many and it often helps in the design to be able to characterize which type of relationship we do have.

(Refer Slide Time: 16:51)



Coming to the attributes we can note that attributes are of different types, one is they could be simple or composite a simple attribute is just one single domain value like a salary number like an id like a name string and so on; whereas, a composite attribute may comprise of multiple parts.

(Refer Slide Time: 17:16)

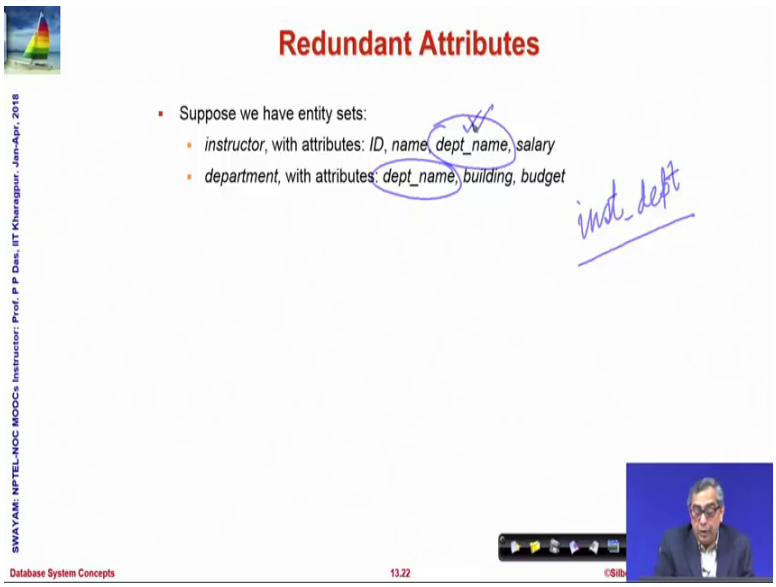


So consider this, this is a composite attribute, so name is an attribute if I think of then it has different parts, it is a first name middle name last name if I think of address it has. So, many different parts then street itself has so many different parts. So, whenever an

attribute is comprised some more of the components, when it is not a simple value then it is called a composite attribute we will see how to handle them; then some attributes may be single valued, for example a person has a has 1 name let us say, but has one address, but may have 2 or more phone numbers, the attributes which can take more than 1 value is known to be multi valued attribute.

So, we also need to specify whether certain specifying in the design whether certain attributes are single valued or multiple valued multi valued; of course, single value attributes are easy to deal with if it is multi valued we need to do some design changes. Certain attributes can be derived for example age, now I cannot keep the age of some a person in the database because, with every day the age changes. So, what we typically keep is a date of birth and the age is computed on the day when the particular query is made to find out what the age is so it is called a derived attribute and each 1 of them will have corresponding set of domains.

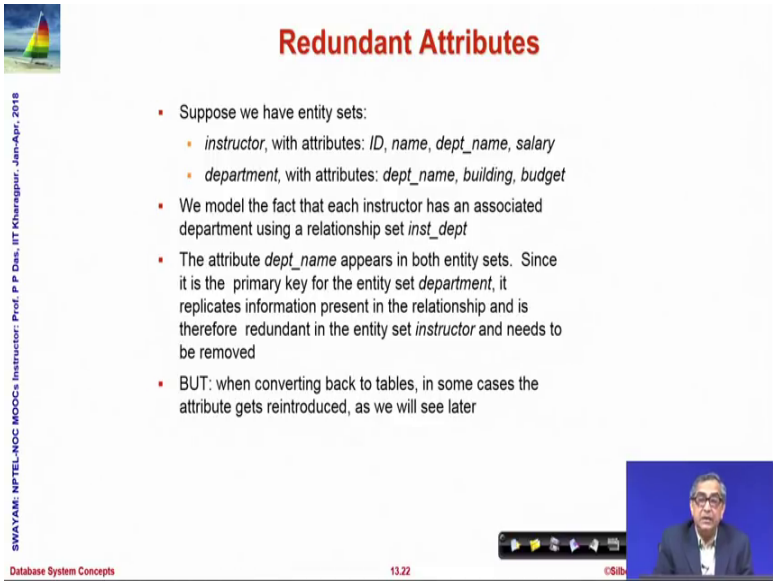
(Refer Slide Time: 18:53)



Some attributes in the design may turn out to be redundant also consider this you have already seen this this is an instructor, which has a department name along with the different attributes and certainly I have a department table, so which department relation which gives the details of the department. Now since every instructor belongs to a department, so naturally we might want to have a relation in department which could give the instructor and his or her department name. So, if we maintain that then this

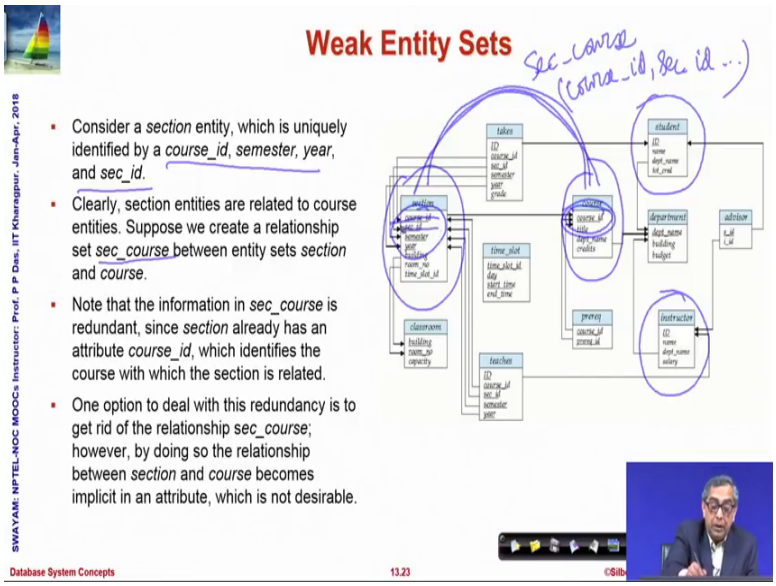
becomes a redundant attribute this is not required, because if that information is already there in this relation.

(Refer Slide Time: 19:49)



So, in several cases there is a question of whether we maintain some information in terms of a relation or we can make that directly include that directly in the entity set and get rid of that relation. So, if I have the inch depth relation and then the attribute department name appears on both these sets, inst dept as well as on the instructor and there is duplication replication of the data which we want to avoid. But we will see the different cases when which style of design, whether we would be better to maintain the department name as a part of the instructor relation or it would be better not to have it there and have a separate relation which maps instructor id against the department name.

(Refer Slide Time: 20:48)



Finally comes a concept of weak entity sets you need to understand this a little bit, consider the university database example. So, we have courses we have students we have instructors and we have section, a section is if a course is large then it needs to be taught in multiple sections. So, for the same course at the same semester in the same year I may have different sections, in which the students are divided and naturally there could be multiple instructors each teaching 1 section of that course and students will be distributed on the sections not on the course.

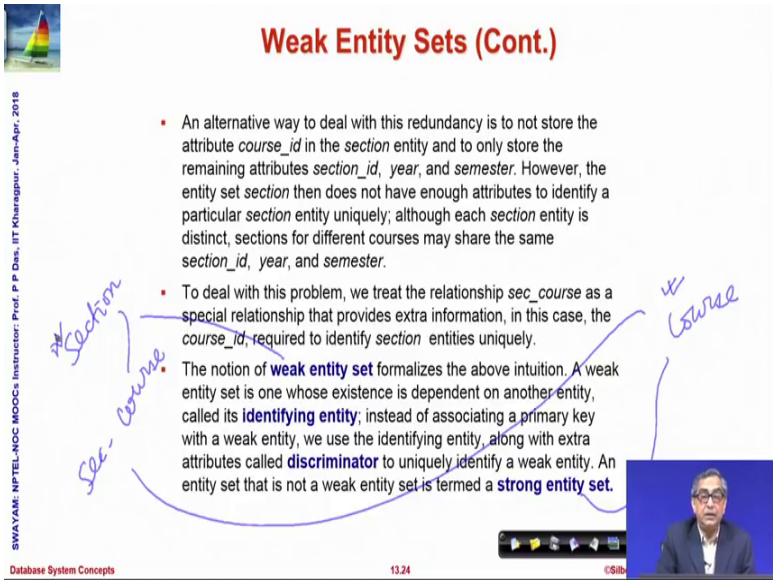
Now, consider this section entity, if you look into this then this is how what we we maintained we did a course id semester year and section id, but if you look into specifically and if you want to know know you know that there is a section and there is a course. So, you may want to relate these 2 section with the course and set up an entity between them. So, what will it relate it will relate the course id of the course with all of these, but the course id is already there as a part of the section right. So, you would say that well it is not required to have the course id, since it already has that and it it identifies it.

So, we can can we remove this course id from here, well if we remove the course id now we have a different problem. If you remove the course id then you have section id semester and year. But this does not uniquely represent the tuples of this relation

because, there could be 2 section as in the same semester in the same year for 2 different courses how do you distinguish them.

So, you get into a situation where the course the section gets identified uniquely provided, either you know the relationship between the section and the course in terms of the sec course relationship or you include the primary key of course, into the relation section which we did in the design and this is not a coincidence this is something which happens regularly and is is the characteristics that specify the existence of weak entity sets.

(Refer Slide Time: 24:11)

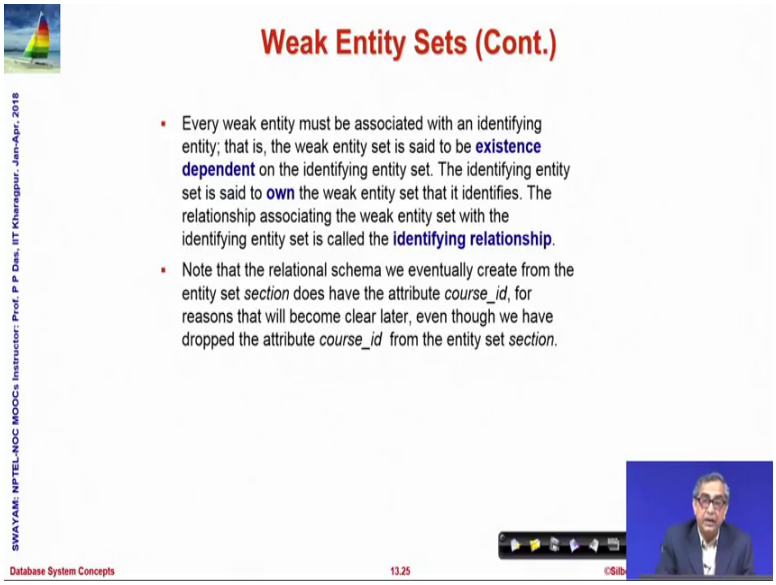


So, the weak entity set is one whose existence depends on another entity set. So, if I just say section having section id year and semester then it is not uniquely specified, until I have a relation section course which relates the section to the particular course id. When such relationships are used to identify entities of a particular entity set, then the unique side the course side which is unique is known as the identifying entity and the other attributes in this case section id year semester are known as the discriminators.

So, we have a relationship between a weak entity set which is section, we have a strong entity set which is course why is it strong because course is identified by course id itself. Section is not unless you have a set course kind of relationship set between the course and the section which specifies that well, for this course this is the section this is the year this is the semester. So, this is the identifying entity through which the entities of this set

gets specified and whenever that situation happens then we say that we have a weak entity set.

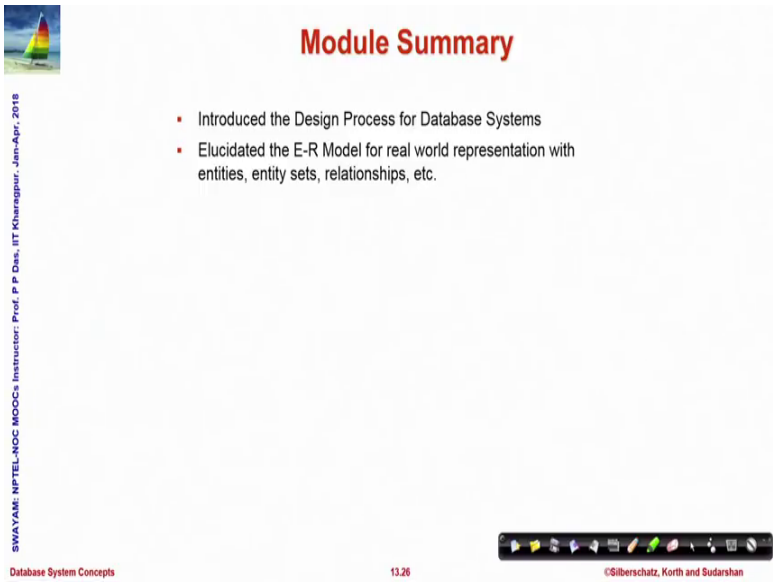
(Refer Slide Time: 26:21)



So, weak entity sets naturally cannot happen by themselves their existence dependent on identifying entity set and the identifying entity set owns the weak entity set. So, the courses in that way own the section and the identifying relationship between them is necessary to uniquely identify every entity of this weak entity set or section in our case.

So, this notion is very important for the design as we will see that the relational schema that we eventually created, in this case from the entity set section we did include course id as a part of the primary key not using the sec course kind of relationship and we will show how this design style for dealing with entity weak entity sets influences the different database designs. So, weak entity sets are critical notions that you need to be aware of need to be confident of.

(Refer Slide Time: 27:34)



So, in summary we have introduced the design process for database systems I will just quickly recap, the first stage is identifying the data items which is leading to the conceptual design, which will primarily do in terms of the entity relationship model identifying the entities the entity sets, the attributes that define the entity set, describe the entity set, the subset of attributes forming primary key that uniquely specifies every entity set, every entity in the entity set and the relationships typically binary may be non binary also, relationships that hold between the different entity sets.

So, this is the conceptual design that will lead to more detail logical design of how the relationship should be organized, what is the cardinality of that, what kind of attributes do I have, whether it is sample whether it is composite whether certain attributes are derived or not. So, all those are different aspects will have to be detailed out and we need to identify what are the weak entity sets and what are the strong entity sets, what are the identifying entities and with that we could complete the logical design and then we will need to make it in terms of it express it in terms of a relational schema.

So, in this module we have just taken a look in the first part the entity relationship model and the very basic of how the conceptual design. We will go forward, in the model we have seen all the different primitives required to represent the reality represent what holds in the real world.