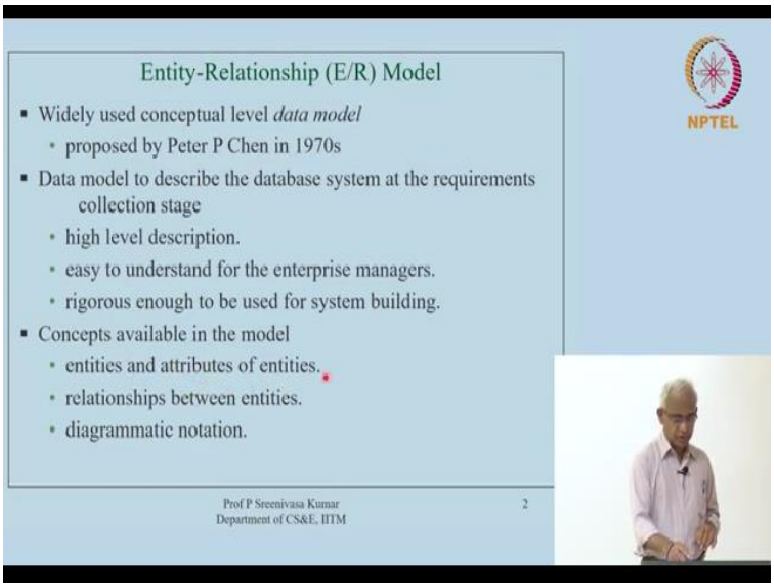
Database Systems
Prof. Sreenivasa Kumar
Computer Science and Engineering Department
Indian Institute of Technology-Madras

Lecture-04
Introduction To ER Model

In this lecture we are starting of with a new module the entity relationship model, ER model. Let us briefly recall what are data models. A data model is essentially a set of conceptual tools that we will use in order to describe the database at a certain level of abstraction. And we need data models because database systems are the databases that were built are pretty complex, and we would like to describe them at various levels of details.

Now, to describe the database at a conceptual level, where we can use that for discussion with the stakeholders and get the requirements clearly, we there are a couple of models. The most popular model is this ER model entity relationship model.

(Refer Slide Time: 01:28)



So let us begin with this entity relationship model. It was proposed by Peter Chen in the 1970s and it is kind of widely used conceptual level data model. In this module, will restrict ourselves to a basic ER model and we say there is another there is an extended ER model called EER

model extended entity relationship model. That would be actually required. If you want to go into more details about the lead modeling.

Typically required for advanced databases like databases dealing with engineering, designs, artifacts, multimedia databases, and such databases might require a more detailed modeling than offered by the basic ER model. The basic ER model is sufficient for most purposes, and especially for capturing the typical databases that we will build while dealing with business applicant.

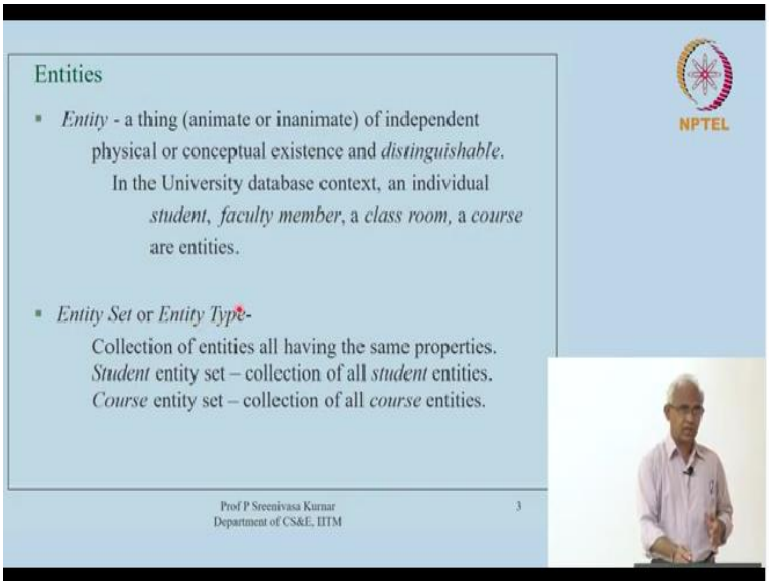
Typical and enterprises like banks and online stores and things like that. So, we also want to move on to the central data model which is the relational data model quickly. So, that you know we will keep this ER model will restrict our discussion to the basic ER model and if time permits we will look at the extended ER model okay. Now, so typically this is the model the data model that we will use to describe the data model database system at the requirements collection stage.

So, it cannot gives a very high level description. And it is easy to understand for the enterprise managers, business enterprises for which were for whom were constructing this database. This is important because we would not expect all the man, all these people who are involved in the management of the enterprise in order to be conversant with computer science and you know, know all the details about how database systems have to be built and things like that.

It is not required for them. But the database has to be you know built based on some clear understanding of the requirements. And so we will need a rigorous enough model to describe the system that is being built okay and ER model offers that kind of facility. So, the concepts available in this model or the concept of entities and attributes of these entities.

And the concept of a relationship which is a association between entities and we also discuss a detailed diagrammatic way of writing down this model. So that we can share it with the stakeholders and then we can use that as a means of describing what we have understood about the database and then get a clear go ahead from the stakeholders okay.

(Refer Slide Time: 06:02)



Okay so let us begin with entities. What exactly are these entities, an entity is a thing of independent, physical or conceptual existence in our domain of interest and could be animate or inanimate, but it is a, what is important is its independent existence and it is also indistinguishable okay, first let us look at some examples of this entities, in a university database context, let us say each individual student, each individual faculty member, a classroom, a course.

All of these can be thought of as entities. Now you can see that the student, faculty member, a classroom, these are real physical things, whether the course is a conceptual thing and they are all distinguishable bah they have certain distinguishing characteristics and we can tell one entity apart from the other entities. For example, if you take a bunch of blue marbles, this distinguishable does not it is very difficult to distinguish between one from one marble and the other.

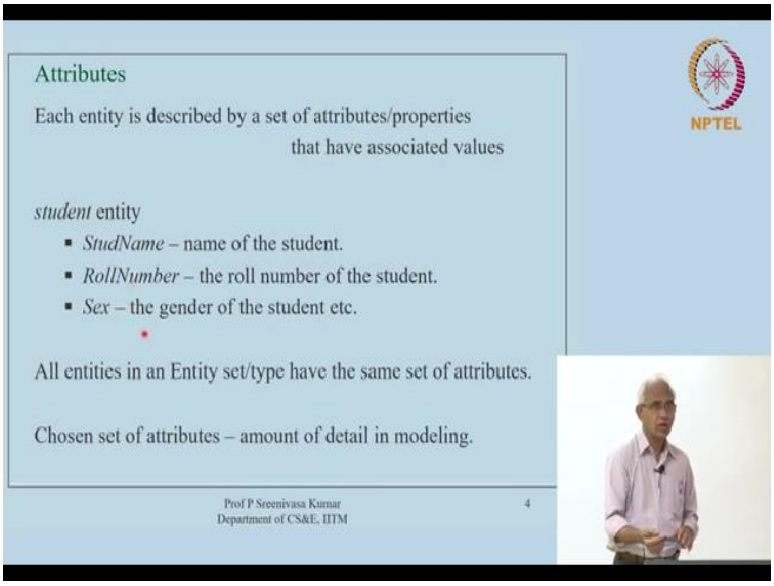
So, such kind of things we cannot treat them as entities a bunch of eggs right, all of them look like, there is here the these entities are clearly distinguishable from each other and they are of independent existence and this is the and these are. So this concept of entity is basically made use of in the data model to kind of capture as to what are the entities of interest in the domain to be kept track of okay.

So, for example in the university database context it is these entities like students, faculty members, classrooms courses, all of these things have to be kept track of. Now, the concept of entity set or entity type is also associated with this. Basically the collection of entities all having the same characteristics or properties okay are together called as the entity set or entity type.

This is very familiar idea to us, because we have already been, you know, dealing with data types. Data types are nothing but collections of pieces of data all of the same time at integer data type, real floating value data type. So now we will be introduced entity type. So, all entities of the same kind are grouped together and we call them as entity type. So, the student entity set is the collection of all student entities, individual student entities.

The course entity set is the collection of all course entities like that okay. So, this is the one of the basic concepts provided by the model the concept of individual entities and the entity type.

(Refer Slide Time: 09:46)



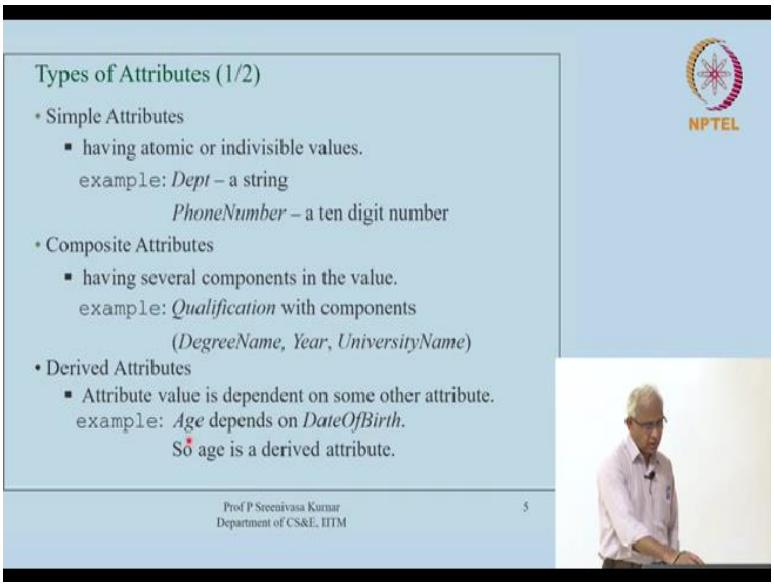
Now, the concept of attributes is used in order to capture the associated information for an entity. In fact, not only for entities even for these what are called relationships there are some associated information and we will use the concept of an attribute to capture this associated information. So, basically attributes are used to kind of describe the entity in more detail. So, each entity is described by a set of attributes or properties.

And in this context we will also assume that the entities of the say of a particular entity type all of them have the same set of attributes, okay. Okay, so we will come to that. So let us take student entity. So, these are some of the various attributes that we would like to keep track associated information with the student entity, like student name. Name of the student, role number is an attribute that takes assumes values of role numbers for the student.

Sex is the gender of the student like that. So, these are the various, we will go a little deeper into these attributes in a short while, there are various kinds of attributes we will see them all. So, all entities in an entity type or set you know that entity characterize that entity okay. Now so of course these the reason we are interested in a entity name, like the student name is that this will you know, capture a value.

So, okay, now, what are the chosen set of attributes, should I include student blood group also in the same set of attributes. Now, that is a matter of design choice, and a matter of how detailed you want to go into modeling and what is the purpose for which the data is being captured and then used. So it all depends on that. So the amount of detail in the modeling is a design choice okay. So some appropriate set of attributes will be chosen to so the purpose of why we are building the database.

(Refer Slide Time: 15:43)



Now, let us look at the various it turns out that there are various types of these attributes. And that will give us an idea of the modeling capabilities that we have in this model. Simple attributes, simple attributes are those attributes that have atomic or indivisible values that means for our purpose we will treat those values as individual values like a phone number is a 10 digit number okay.

Department is a string. So department attribute use for discovering some entity is a string. The phone number attribute is a 10 digit number and we will treat that as an atomic number. So some phone numbers do have area codes and then other things but then for all purposes we will treat it as atomic okay. So the important thing is that if we model it as a simple attribute, we are agreeing that we will treat it as a atomic indivisible value**.**

For example, a building name or a room, for example, my room is the BSB 309 okay, you may say that okay, the BSB actually stands for building senses block and your room number is 309 okay. Should we split it like that. It is again a design choice. We will if we model it as a simple attribute, then we choose not to split it and use it as it is okay. So that is the spirit behind simple attributes.

Whereas, we have composite attributes. These composite attributes have several components in the value several components in that particular value, like for example, if you take qualification as an attribute of an individual person, then it is natural that it has components like degree name, what is the name of the degree and, year, the year of granting of that particular degree.

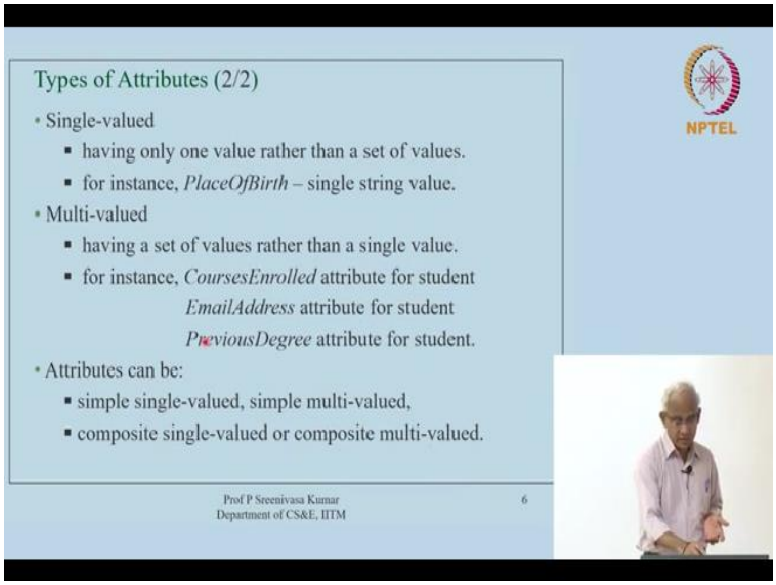
And the university name, what is the university that granted that particular degree to that particular person. So, these are the various components that we can imagine to be present in the qualification attribute. So, these kind of attributes are called composite attributes. So, we have simple attributes as well as composite attributes, and depending on when we discuss with the end users as to how they would like to see their information to be captured.

We will use them appropriately We will ask them is this is a simple attribute enough for this particular capturing of the attribute or is the composite attribute required what is the kind of

information that you want to capture and then make a choice. Now, there are also what are called derived attributes. Attributes whose value is dependent on some other attributes. The classic example here is the age.

The age depends on date of birth. So, date of birth is an attribute and age is another attribute and it is called a derived attribute because you do not have to specify the value for this age. If you have already specified what is date of birth, it can always be calculated. So, we have the notion of derived attributes as well. Like the interest amount on a deposit is a derived attribute because it depends on the rate of interest.

(Refer Slide Time: 16:59)



Now moving on the another kind of classification for attributes is this attribute a single value is it just one value or is going to be a collection of values for example, okay. So, single value means the attribute has exactly one value rather than a set of values. For instance the place of birth is a single valued thing, you cannot have 2 places of birth right. So, by its very nature, the place of birth attribute exactly has 1 value.

So, it is a string value and whereas there are some attributes which are naturally multivalued, that means they have a set of values associated with them you know but not a single value. For example, the email address attribute of a person is typically having multiple values because you

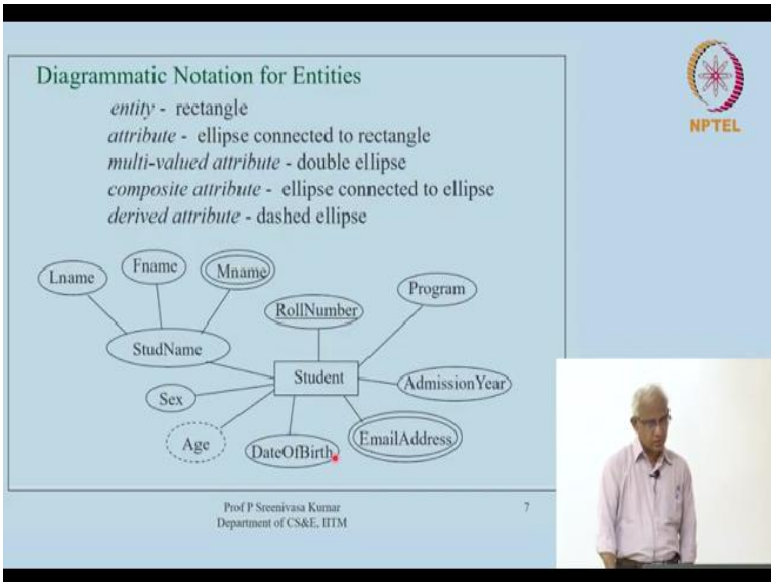
have a institutional email, you have personal email address and maybe some other email address and things like that. You have multiple email addresses.

The previous degree attribute for a student typically has more values, a student might have multiple degrees earned before he is enrolled into one particular degree. The courses enrolled attribute for the student is typically multivalued because each student typically enrolled for multiple courses in a particular semester. So like that these are certain attributes or multivalued and whereas certain attributes a single variable.

So now we have seen 2 kinds of ways of looking at attributes, one of them is are they simple or composite okay. Simple means it has only one component, it does not have multiple components. Composite means it has components inside where a single valued means it always has a one value associated with it whereas, a multivalued means it has a set of values associated with it. Now, all these 4 thing when the 2 of the simple composite and single multivalued can actually combine and then we can have 4 possible combinations of these things.

So, attributes can be simple, single valued or simple multivalued. Composite single valued or composite multivalued. For example, the previous degree attribute of a student is actually multivalued. And it is also composite because each degree has components in it, what is the name of the degree, what is the year in which it was granted who granted that degree it is naturally multivalued. So it is a composite multivalued attribute okay, so are there any questions now.

(Refer Slide Time: 20:41)



So here is the diagrammatic notation for entities, that is typically made use of in this context. So let me tell you describe to you this. So we write the name of the entity in like this student, we typically do not write students, okay, we do not write the plural of the web there necessarily. Now so usually right just the singular name, student.

So, that is the name of the entity and entity is always described is in a rectangular box. And then around that, we have lines connecting to the entity and then we write attributes in ellipses like this. So the program is an attribute. So it is supposed to have a value depending on what is the program the student is in enrolling, like the a B tech student and M. tech student, Ms student, a PhD student and all that.

And then we have roll number as an additional attribute. And then here is a student name attribute, this one is a composite attribute. So, if it is a composite attribute, then it will be an ellipse and it will have other ellipses connected to it. So this particular attribute has the last name, first name, last name, and multiple middle names, so multiple middle names. So if this is a multivalued then we put double ellipse to indicate there is multivalued.

So, this is a very this kind of modeling is appropriate for our names because most people have a first name and a last name and multiple middle names. Typically Indian names do have several middle names okay. Especially people from Andhra Pradesh have long names right. And so it is

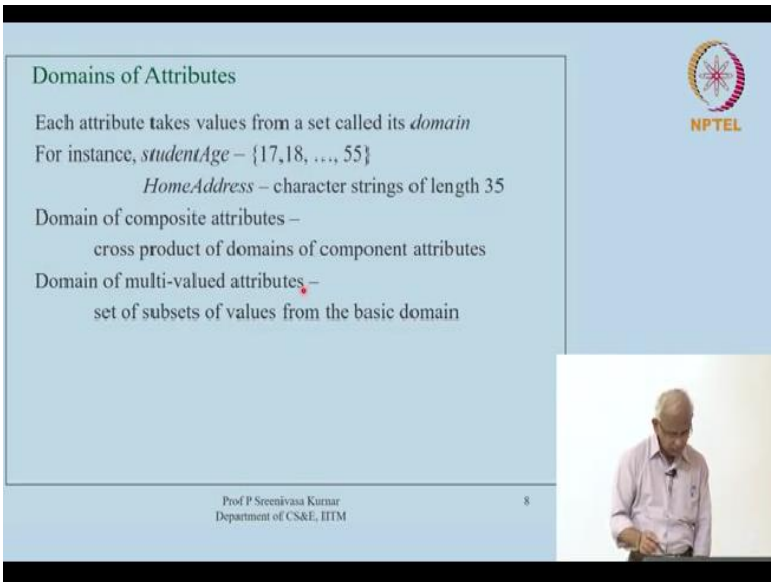
natural to capture them using multiple middle names okay, then we have this simple email address, but it is multivalued.

So it is simple multivalued here is a composite attribute, one of its components is multivalued okay, and here is date of birth which is a simple attribute and sometimes we underline a particular attribute, I will tell you why we do that in the next slide, sex is another attribute, age is a derived attribute, it is the data from date of birth. So, like this, we have this diagrammatic notation to describe the entity and it is.

So, typically the situation that we are modeling for which and this situation for which we are building the information system has several entities of this kind and we under engage with the stakeholders in a discussion and then illicit I mean collect all this information from them and then develop these diagrams and then have a discussion with them and to kind of finalize our understanding of the, the domain for which we are building the information system.

Now, remember that this is all data model requirements, this is not functional requirements, functional requirements are separate, right functional requirements are not captured in this particular model.

(Refer Slide Time: 24:52)



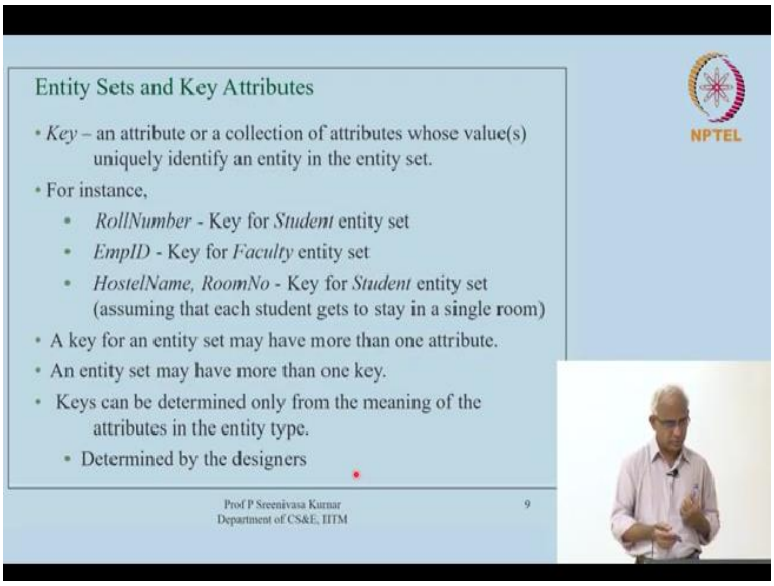
Now, what exactly are these mathematically. So, mathematical each of these attributes has a domain associated with it, domain also are sometimes called value sets. So, each attribute takes values from a set you know which is called it is domain. For example, the student age, we can say that it has values. This is the set of values associated with, we do not expect students to have age more than 55.

And I think the entry age is 17 years right. And then we say that home address is a character string of length that if I will fix it saying that that four characters probably enough to take care of the whole mattress. So like that, we have domains, domains are basically not nothing but sets of values. And these are associated with these attributes. Now for composite attribute we can think of that as having a value, which is an element of this cross product of the domains of the component attributes right.

So, for a composite attribute, the value is a tuple, actually right. It is a composite attribute yes component. So naturally we can think of the value as a tuple. And tuples are so these tuples come from the cross product of the domains of the corresponding component attributes. There are components inside the composite attributes. So for those component attributes, we have domain sets. So we can take those domain sets from the cross product.

And I am considered a number of the cross product as the value of this composite attribute. In a similar way, for multivalued attributes, we basically take the sets of subsets, the power set all the values from the basic domain as the domain, because it can have multiple values, like for example, courses registered is a multivalued attribute. So we will take the basic domain which is the course thing and then consider subsets of courses as the values for this multivalued attributes okay.

(Refer Slide Time: 27:37)



So we have these domains. Now moving on there is an interesting aspect about certain kinds of attributes and they are called key attributes. So for an entity set we designate an attribute or a collection of attributes okay is it either a single attribute or a collection of attributes together that have this interesting property that their values would be able to uniquely identify an entity in the entity set, such attribute as correlate key attribute okay.

So let me repeat a key attribute or key is basically is either a single attribute or a collection of attributes whose values can be used to uniquely identify one entity in the entire entity set. So, entity set has lots of entities. So, how do you identify one of them, how do you pick one of them if necessary. What we are saying here is that there are certain attributes whose values can help us to uniquely pinpoint one attribute to one entity in the central entity set.

So will be given examples the roll number. The roll number is a key for this student entity set. The moment you give me a roll number I will be able to identify what student you are talking about. Who are you talking about, the employee ID is the key for the faculty entity set okay, for the same student entity set, let us assume that each student gets to stay in a single room assigned a single room and these rooms are, you know, they are in hostels.

And so the name of the hostel and the room number together would be a key under this assumption that the each student gets to stay in a single room okay. So we are kind of you know,

the situation that we are assuming here is that all students stay in hostels, first thing all students stay in hostel and there are multiple hostels each hostel is identified by a name and then there is a room number.

And this hostel name together with the room number will serve as the key for the store and we can uniquely identify this student because of this assumption right. Now so as I was mentioning here key for an entity set may have more than one attribute. For example here hostel name, room number are 2 attributes and also an entity set may have more than 1. For example the student entity set here has roll number as 1 key and then the hostel name roll number to whether as a key.

For example for a faculty member who is you know an employee, employee ID is a key and there are certain global you know, attributes like you know, the, for example, the PAN number, the permanent account number of taxpayers that is given. We all have a PAN, so you can even clarify the ID faculty by his PAN numbers also. The institute actually also gives a brief abbreviated names for faculty.

So that they can be identified uniquely using that abbreviated name okay, so you can see that basically the point I am trying to convey here is that entity set may have more than 1 key and a key can have more than 1 attribute also. Now, who determines this case. That is an interesting question, keys can only be determined from the meaning of this attributes of those entity types okay and under underlying assumptions in the domain.

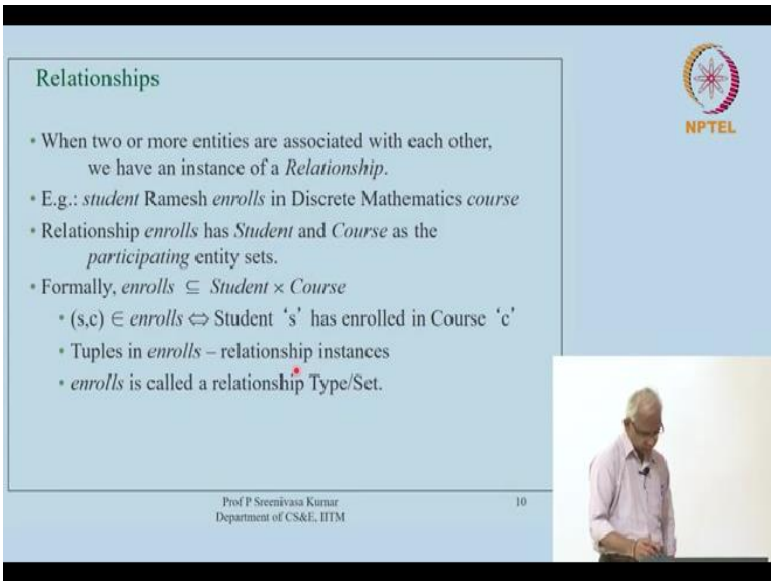
So, they have to be kind of determined by the designers at this stage when we are designing the model we have to identify, so you cannot take your set of you know entities and then just look at them you know and the values and things like that and they simply decided that something is a is a key okay. It has to come from the underlying meaning of the attribute. That is the important thing.

For example, if you write down a bunch of names and other details of student entities, some 100 student entities on a sheet of paper and then I can go through and then I figured out that okay, the

first name seems to be a key, because it so happened that you know to Sreenivasa is in that particular set, but that does not really make first name as a key because we know that there are people who have there are, you know, certain first names that are very common.

And so you first name cannot be a key for a bunch of people, so, it kind of comes from the natural meaning of this attribute. So, it has to be designed by the attribute by we cannot examine a collection of entities and then come to some conclusion based on that particular data set okay. So, let us summarize. So we have this concept of entities, this is the entity relationship. So the first thing is entity and we have this notion of entities and entity types and then attributes for them and various kinds of attributes. So these are all collection of conceptual tools that are available for us to start modeling the domain.

(Refer Slide Time: 35:11)



Moving on, let us look at relationships now. Okay, now relationship is another major concept that is available in the ER model. Basically what it allows us to do is to capture associations between entities. So entities do not simply exist, you know, in a domain. They also interact them and there is some associations between them. So we want to capture them, when 2 or more entities are associated with each other we have an instance of a relationship okay.

Let me give you an example. For example, the student Ramesh enrolls in discrete mathematics course, typical scenario. So, the entity types here are in italics. So student is an entity type,

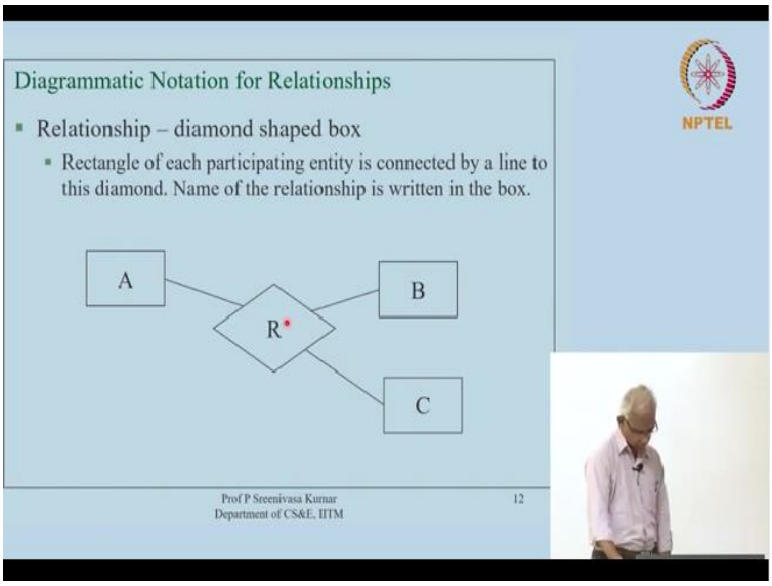
Ramesh is, you know okay, let us assume that we are talking about one particular individual and discrete mathematics is a course and enrolls is this association between these 2 entities. So, this is a name of a, this is a potential name of a relationship that enrollment or enrolls.

So, this particular thing, in fact, is one instance of that relationship. There are several instances of that kind of relationship that there are several people who are enrolling into different kinds of courses. So, each of them is it is an instance of this enrolls relationship So, the relationship enrolls basically has the student and course as the participating entity sets, we call it as a participating entity sets.

Now, formally speaking enrolls can be thought of as a subset of the cross product of the student and course entity types, the student entity course entity. So each of them is a entity type. So, it has a collection of entities in this student, a collection of course, so who is doing what course. So, if some s, c belongs to enrolls what it means is that student S has enrolled course c okay. So basically tuples in this enrolls are these relationship instances.

And enrolls itself is called a relationship type or set. Just like we had entity type, we now also have a relationship type okay, which captures basically a bunch of associations of the same kind or put together in a relationship type. So, enrolls is a relationship type and there are several instances of this relationship, each of them having 2 components, which is this student, course in this case.

(Refer Slide Time: 38:50)



Now we use a term called degree for a relationship, basically it was to capture as to how many number of entities are participating in that relationship. So, typically it is binary, but in some situations we may have to use ternary and sometimes n-ary relationships okay. So, degree 3 means 3 entities are participating in that relationship. So, the tuples will be having 3 components degree.

Degree n is n-ary so, in general, you can see that so the basically the idea of the term degree is used to indicate the number of participating entities in the relationship. So, here is the diagrammatic notation for relationships. Relationships are shown using a diamond and all the participating entities which are rectangles, right are connected to the relationship symbol and we write the name of the relationship here and the name of the.

So, remember that this has other details to it like for example, all the attributes are around this right. So, there is a lot we have not shown those things, but those things are there. Now, so, similarly, this also has attributes around it and now the new thing we are bringing in relationship and so the relationship we draw in a diamond shaped box and then draw lines to indicate that all these 3 entities are participating in that relationship okay.

Now, probably I will stop here for today because it is it 45 will continue in the next lecture with various kinds of relationships that are existing and how we can capture details of the domain using those relationship kinds okay.